Bayesian Network Reconstruction Using Systems Genetics Data: Comparison of MCMC Methods

Shinya Tasaki,*,[1] Ben Sauerwine,[†] Bruce Hoff,[‡] Hiroyoshi Toyoshiba,* Chris Gaiteri,[‡,§,,1,2] and Elias Chaibub Neto[‡,1,2]**

*Integrated Technology Research Laboratory, Pharmaceutical Research Division, Takeda Pharmaceutical Company, Fujisawa, Kanagawa, Japan 251-8555, [†]Google, Seattle, Washington 98103, [‡]Sage Bionetworks, Seattle, Washington 98109, [§]Modeling, Analysis, and Theory Group, Allen Institute for Brain Science, Seattle, Washington 98103, and **Rush Alzheimer's Disease Center, Rush University Medical Center, Chicago, Illinois 60612

ABSTRACT Reconstructing biological networks using high-throughput technologies has the potential to produce condition-specific interactomes. But are these reconstructed networks a reliable source of biological interactions? Do some network inference methods offer dramatically improved performance on certain types of networks? To facilitate the use of network inference methods in systems biology, we report a large-scale simulation study comparing the ability of Markov chain Monte Carlo (MCMC) samplers to reverse engineer Bayesian networks. The MCMC samplers we investigated included foundational and state-of-the-art Metropolis–Hastings and Gibbs sampling approaches, as well as novel samplers we have designed. To enable a comprehensive comparison, we simulated gene expression and genetics data from known network structures under a range of biologically plausible scenarios. We examine the overall quality of network inference via different methods, as well as how their performance is affected by network characteristics. Our simulations reveal that network size, edge density, and strength of gene-to-gene signaling are major parameters that differentiate the performance of various samplers. Specifically, more recent samplers including our novel methods outperform traditional samplers for highly interconnected large networks with strong gene-to-gene signaling. Our newly developed samplers show comparable or superior performance to the top existing methods. Moreover, this performance gain is strongest in networks with biologically oriented topology, which indicates that our novel samplers are suitable for inferring biological networks. The performance of MCMC samplers in this simulation framework can guide the choice of methods for network reconstruction using systems genetics data.

KEYWORDS Bayesian networks; MCMC methods; causal inference; eSNPs; network reconstruction

COMPLEX diseases such as Alzheimer's disease and type 2 diabetes are influenced by intricate gene-to-gene and gene-by-environment interactions (Peila *et al.* 2002; Huang *et al.* 2005; Liu *et al.* 2007; Rhinn *et al.* 2013). The goal of gene network reverse engineering is to learn the gene-to-gene interaction architecture underlying such diseases. Some algorithms output networks that correspond to putative causal relationships among genes, by combining SNP and expression data. These inferred networks can be used to generate hypotheses that can be validated experimentally (Schadt *et al.* 2005; Chen *et al.* 2007; Aten *et al.* 2008; Ferrara *et al.* 2008; Chaibub Neto *et al.* 2008, 2013; Duarte and Zeng 2011). In particular, Bayesian approaches to inferring causal networks are becoming a common practice in the field of systems biology (Zhu *et al.* 2007, 2008; Chaibub Neto *et al.* 2010; Hageman *et al.* 2011; Moon *et al.* 2014) and have successfully generated novel insights into biological processes (Zhang *et al.* 2013). Nonetheless, inferring the structure of Bayesian networks remains a challenging statistical and computational problem. Specifically, the relative performance of different causal network reconstruction algorithms is unclear when they are applied to real biological data sets. Therefore, to enable more accurate network reconstructions, we conduct a systematic comparison of the performance of several Markov chain Monte Carlo

doi: 10.1534/genetics.114.172619
Manuscript received December 14, 2014; accepted for publication January 26, 2015; published Early Online January 28, 2015.
Available freely online through the author-supported open access option.
Supporting information is available online at http://www.genetics.org/lookup/suppl/doi:10.1534/genetics.114.172619/-/DC1.
[1]Corresponding authors: Takeda Pharmaceutical Company Limited 2-26-1, Muraoka-Higashi, Fujisawa, Kanagawa 251-8555, Japan. Email: stasaki@gmail.com; Sage Bionetworks, 1100 Fairview Ave. N, Seattle, WA 98109. Email: elias.chaibub.neto@sagebase.org; Rush University Medical Center, 600 S Paulina St., Chicago, IL 60612. Email: gaiteri@gmail.com
[2]These authors contributed equally to this work.

(MCMC) samplers, as well as novel sampling methods. Our conclusions facilitate the recovery of correct networks under a range of biologically realistic scenarios.

We report the results of a large-scale simulation study comparing the relative performance of state-of-the-art MCMC samplers (Grzegorczyk and Husmeier 2008; Goudie and Mukherjee 2011) with novel sampler variations developed by us and with the standard Metropolis–Hastings structure samplers (Madigan *et al.* 1995; Giudici and Castelo 2003). Efforts to compare MCMC samplers have focused on data generated from a handful of benchmark networks such as the ALARM network (Beinlinch *et al.* 1989). Such limited test sets can lead to conclusions about the performance of various methods that are not necessarily robust. Therefore, to reach high-confidence conclusions, we evaluated the merits of various network reconstruction algorithms over a wide range of biologically relevant networks. For instance, envision that researchers develop sampler *A* that, in theory, is expected to outperform an alternative sampler, *B*, on sparse networks. To empirically check this hypothesis, the researchers may restrict their attention to a single benchmark network. Even if multiple networks are generated to test this hypothesis, they likely come from a specific parameter set, such as 200 data samples generated from a sparse network consisting of 40 continuous variables, via a set of linear structural equations with moderate values for the regression coefficients and residual variances. Suppose further that, as expected, sampler *A* does outperform sampler *B* across most of the simulations. Although the researchers might be inclined to claim that sampler *A* is better than *B* in sparse networks, we argue that the researchers cannot really support this statement because the effect of sparsity on the sampler's performance is confounded with the effects of the other simulation parameters, namely, the sample size, the number of nodes, and the amounts of signal and noise. That is, it is not possible to determine whether sampler *A* outperforms sampler *B*, because the benchmark network is sparse or because, in reality, sampler *A* is better than *B* for the particular choice of simulation parameters adopted in the simulation. All the researchers can claim is that sampler *A* performs better than *B* for the particular values of the simulation parameters adopted in the study, but there are no guarantees that sampler *A* would outperform sampler *B*, had the researchers chosen a different set of simulation parameter values.

Therefore, to perform a rigorous comparison of the performance of different MCMC samplers and to investigate the conditions under which one sampler performs better than another, we designed a multifactorial simulation study with crossed factors. In our simulation study, network parameters played the role of factors, and the difference in area under the response curve of different samplers played the role of the response variable (Figure 1). By comparing the performance of methods across many different situations, the differences in performance we observe are less likely to be due to a specific parameter setting. Moreover, we can track how different types of network topology influence the performance of all methods or a specific method. This enables biologists to understand how data sets with different origins and features are likely to affect the accuracy of estimated networks and which methods are optimized for the characteristics of a particular biological data set.

In our simulations, we investigated the effects of eight distinct simulation parameters, namely sample size, network topology, the number of network nodes, incorporation of genetic information, average edge density, average gene-to-gene signal, average SNP-to-gene signal, and intrinsic expression noise. (See *Methods* for further details.) These simulation variables were chosen to represent aspects of biological data sets that are reasonably expected to vary in a wide range of future applications. For instance, the strength of gene–gene correlations is controlled by a variety of biophysical processes such as transcription factor binding, chromosome configuration, and epigenetics (Gaiteri *et al.* 2014). We compared the performance of four published MCMC samplers and five novel sampler variations that we developed. The published samplers included (i) the foundational Metropolis–Hastings structure sampler ("STR sampler") (Madigan *et al.* 1995; Giudici and Castelo 2003), (ii) the "REV" Metropolis–Hastings sampler (Grzegorczyk and Husmeier 2008), (iii) the single-parent set block Gibbs sampler ("1PB"), and (iv) the two-parent sets block Gibbs sampler ("2PB") (Goudie and Mukherjee 2011). The new samplers included (v) the three-parent sets block Gibbs sampler ("3PB"), (vi) the four-parent sets block Gibbs sampler ("4PB"), (vii) the connected two-parent sets block sampler ("c2PB"), (viii) the connected three-parent sets block sampler ("c3PB"), and (ix) the connected four-parent sets block sampler ("c4PB"). Descriptions and comments on each of these samplers are provided in the *Appendix*.

Since our multifactorial design involves a large number of distinct factor-level combinations (1458 in total), and for each combination we run nine distinct MCMC samplers, we considered only a single replication per simulation parameter combination. Each sampler was run for a fixed time with either a single longer chain or multiple shorter chains. Collectively, the experiment included 52,488 instances of reverse-engineering gene regulatory networks. To the best of our knowledge, this is the largest simulation study comparing MCMC samplers for structure learning in Bayesian networks.

Methods

Bayesian networks

Background and technical material on Bayesian networks and the MCMC samplers investigated in this study are presented in the *Appendix*.

Setting of the MCMC runs

Each sampler was run for a fixed time with either a single longer chain or multiple shorter chains, starting from different random initial structures. For 30, 65, and 100 node networks, we executed MCMC samplers in 30, 300, and 2100 sec, respectively, on an Opteron 2.2-GHz core. For single-chain running, we collected at most 5000 networks uniformly from

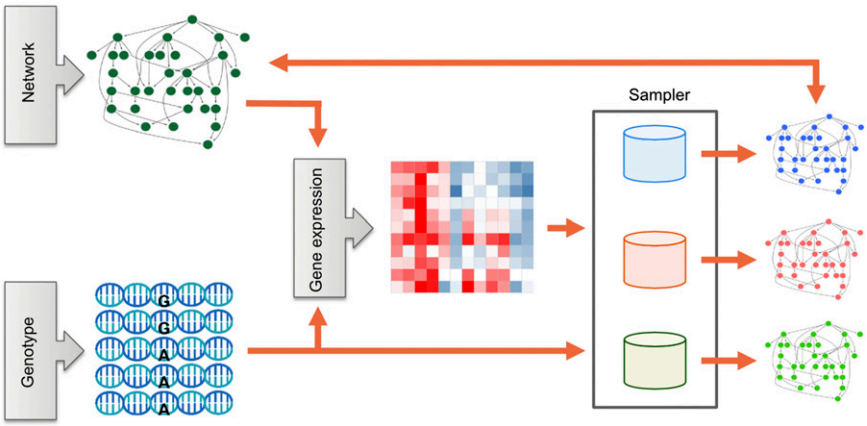

Figure 1 Workflow of simulation study. Systems genetics data composed of gene expression and genotyping data were generated based on a regulatory network consisting of genes and SNPs. Bayesian networks were estimated by applying various structure MCMC samplers to these systems genetics data, generated by a known network structure. The composite networks obtained by Bayesian model averaging were then compared with a true network structure. AUCPR and AUCROC are employed as a performance measure. Overall performance of different MCMC samplers and identification of the network features that affect the accuracy of network reconstruction were investigated through ANOVA of the AUCPR or AUCROC results.

a chain and 20% of initial networks were discarded as burn-in. When running multiple chains, the chain length was controlled so that on average 50 chains completed in the time allotted. The last network of a chain was collected from each chain.

Program implementation

All the MCMC samplers were implemented in MATLAB 2012a (MathWorks). To conduct fair comparisons, the routines shared across the MCMC samplers were implemented in the same way by using the same subfunctions. Moreover, the routines specific for each MCMC sampler were highly optimized. All the programs were compiled by MATLAB Compiler to enable batch execution. The code to run all methods is available for download (DOI: 10.7303/syn2910187).

Simulation parameters

The selection and range of simulation parameters were designed to reflect characteristics of typical systems genetics experiments. To investigate the relative performance of the different structure samplers, we designed a multifactorial simulation experiment with crossed factors, focusing on the effects of seven distinct simulation parameters, namely the following:

1. Network topology t, with levels "random" and "EIPO" (exponential in-degree and power law on out-degree, as proposed by Guelzim *et al.* 2002).
2. Number of network nodes, p, with levels "30," "65," and "100."
3. Edge density, d, with levels "low," "medium," and "high." The edge density of a network is defined as the number of edges divided by the number of possible edges, namely, $p \times (p-1)$. For each simulation, d was set to 0.02, 0.04, and 0.06 as low, medium, and high edge density, respectively. We did not limit the maximum number of parent nodes in simulated networks, even though sampling methods often limit the number of possible parents, for reasons of computational efficiency.
4. Average gene-to-gene signal, η, defined as the average absolute value of the nonzero coefficients (of the regression of a gene on another). For each simulation, η was sampled uniformly from the ranges [0, 0.33], [0.33, 0.66], [0.66, 1] corresponding to low, medium, and high, respectively.
5. Intrinsic expression noise, σ^2, with levels low, medium, and high, and defined as the variance of the error term, $\epsilon \sim N(0, \sigma^2)$, used in the simulation of the expression values (via linear structural equations). For each simulation, σ^2 was sampled uniformly from the ranges [0, 1], [1, 2], [2, 3] corresponding to low, medium, and high, respectively.
6. Average SNP-to-gene signal, γ, defined as the average absolute value of the nonzero coefficients (of the regression of an expression phenotype on a SNP). For each simulation, γ was sampled uniformly from the ranges [0, 1], [1, 2], [2, 3] corresponding to low, medium, and high, respectively.
7. Sample size, n, with levels 100, 200, 300. This choice reflects typical sample sizes observed in the literature for causal gene networks (Zhang *et al.* 2013).

In total, our experiment was conducted based on $2 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 1458$ distinct networks.

Simulation of network structures

We generated network structure based on EIPO topology observed in transcriptional regulatory networks (Guelzim *et al.* 2002). Random topology networks were also generated as reference. We did not limit the maximum number of parent nodes for the simulated networks. Network structures were generated using the SysGenSIM software (Pinna *et al.* 2011). This software allows the user to control the number of nodes and the network average degree ($=2(p-1)d$), as well as choose among several topology types, including the random network topology and the EIPO topology. Although the software can also simulate genotype and expression data from experimental crosses (where the latter are generated according to nonlinear ordinary differential equations), we do not employ those features for data generation. Instead, we generate genotype data from outbred populations and simulate expression data from a multivariate normal distribution consistent with the structural equation model representing the network structure.

Simulation of SNP data

To incorporate realistic genetics data in our simulations, we generate SNP data matrices by randomly selecting chunks of real SNP data from the HapMap3 database. To do this, we first randomly choose p genes from the refGene in the UCSC Genome Browser and then select all *cis*-SNPs associated with the p genes from genotype data on a Caucasian population. We define a *cis*-SNP as any SNP physically located between (−) 110 kb upstream of the transcription start site and (+)40 kb downstream of the transcription end site, because this region is thought to contain 99% of *cis*-eSNPs (e: expression single nucleotide polymorphism) (Veyrieras *et al.* 2008).

Simulation of network data

Given the SNP data and a (possibly cyclic) directed network structure, G, with continuous (gene expression) and discrete (SNP) nodes, we generated multivariate normal gene expression data with a covariance structure implied by the structural equation model (SEM) describing G. Because the SNP nodes are necessarily exogenous variables in G, the associated SEM can be represented, in matrix notation, as

$$X = BX + CQ + \epsilon \tag{1}$$

(Liu *et al.* 2008), where (i) ϵ is a $p \times n$ matrix of independent and identically distributed $N(0, \sigma^2)$ residual error terms, (ii) X is a $p \times n$ matrix of expression levels, and (iii) Q is a $k \times n$ matrix of SNP genotype codes. B is a $p \times p$ matrix of gene-to-gene causal effects, where the element β_{ij} represents the partial regression coefficient of the regression of gene i on gene j. The structure of matrix B corresponds to the directed graph representing the interactions between the genes with edges corresponding of nonzero elements of B. Because we adopt independent residual error terms, matrix B can be rearranged into a lower triangular matrix when G is acyclic, but will be necessarily nontriangular for cyclic graphs. C is a $p \times k$ matrix of SNP-to-gene causal effects. Each entry (i, j) of C represents the partial regression coefficient of the regression of gene i on SNP j. The SNP-to-gene edges in G are represented by the nonzero elements of C. Given the matrices B, C, Q, and ϵ, we generate the expression data matrix as $X = (I-B)^{-1}(CQ + \epsilon)$. Note that because each column of ϵ follows a $N_p(0, \sigma^2 I_p)$ distribution, we have that the conditional distribution of each column X_j of X given B, C, and Q is $N_p\left((I-B)^{-1}CQ_j, \sigma^2(I-B)^{-1}((I-B)^{-1})^t\right)$. Furthermore, no diagonal dominance enforcement of $(I-B)$ was necessary to ensure matrix invertibility since the simulated networks were small; the diagonal and most of the off-diagonal entries of B were zero (due to the absence of self loops and relatively low connectivity of the simulated networks), and the nonzero entries of B consisted of low values.

The values of the nonzero partial regression coefficients in B and C are sampled randomly from a uniform distribution and then rescaled so that the average of their absolute values is η and γ, respectively. The range of η, γ, and ρ is determined so that the variance of the expression phenotypes explained by eSNPs in simulated data covers that observed in real data, which ranges from 0.1 to 0.3 (Grundberg *et al.* 2012; McKenzie *et al.* 2014; Yang *et al.* 2014). In particular, the median variance explained by SNPs of our simulated data was 0.11 with a median absolute deviation of 0.15. An additional constraint to C was that 20% of the expression phenotypes have SNPs, which was based on the empirical proportion of genes with eSNPs, ranging from 5% to 35% (Brown *et al.* 2013). In addition, each such expression phenotype can have at most two SNPs, since three independent eSNPs for a single gene were very rare in studies with moderate sample size (Stranger *et al.* 2012). Consequently, 80% of the rows of C are completely 0, and the remaining 20% of the rows can have at most two nonzero entries.

eSNP mapping

We perform eSNP mapping tailored to the network structure by evaluating conditional independence relations between expression traits and SNPs, given their expression trait parents (Chaibub Neto *et al.* 2010). The likelihood ratio between the full model and the null model that is generated by removing the SNPs term from the equation can be used as a formal test of conditional independence. The empirical null distribution of the likelihood ratio is estimated by 1000 permutations of individual labels as follows. At each time, the individual labels of either expression trait or SNPs were randomly permuted so that relations between expression trait and SNPs were broken while keeping the correlation structure among expression traits intact (and while preserving the correlation structure of the SNPs as well). Then the likelihood ratios of all the pairwise relationships between expression traits and their *cis*-SNPs were calculated, followed by the collection of the maximum value across all computed likelihood ratios. The permutation null distribution generated by this procedure is then used to test the null hypothesis of detecting an association given that none of the SNPs are associated with the expression trait. The genome-wide error rate of detecting *cis*-eSNP associations was controlled at 0.05.

Performance measures

We evaluated network reconstruction performance, using the area under the curve of the precision-recall plot (AUCPR), where

$$\text{precision} = \frac{\text{number of true positives}}{\text{total number of edges detected}},$$

$$\text{recall} = \frac{\text{number of true positives}}{\text{total number of true edges}},$$

and a true positive was defined as an inferred directed edge that existed in the true network used to generate the expression data. In some cases, the area under the curve of the receiver operating characteristic (AUCROC) was used

for performance evaluation. AUCROC is a composite measure based on true positive rate (recall) and false positive rate, defined as

$$\text{false positive rate} = \frac{\text{number of false positives}}{\text{total number of true absent edges}},$$

where a false positive was defined as an inferred directed edge that did not exist in the true network used to generate the expression data.

Results

Network characteristics affect the performance of Bayesian network reconstruction

To elucidate effects of various biological characteristics on network estimation accuracy, we performed a systematic simulation experiment with an ensemble of systems genetics data where each parameter (network characteristic) was sampled over a biologically plausible range. To evaluate the accuracy of an estimated network, we applied Bayesian model averaging to the probabilistic network estimates produced by any given method, to obtain a composite adjacency matrix. Then, AUCPR as adopted as a measure of correctness of an output adjacency matrix, given a true network. Using this response variable, we used ANOVA to identify network features that have significant impact on the performance of network reconstruction.

All the network characteristics tested, as well as various 2×2 interactions among these, were found to be significantly associated with AUCPR (Figure 2), but the strength of the effects and form of the interactions varied widely. Our experiment indicates that network topology affects the accuracy of network reconstruction. Networks with EIPO topology were reconstructed more accurately than those with random topology (Figure 2A). The experiment also identified harmful and beneficial effects of network characteristics on estimation accuracy. The increase in the number of genes, edge density, and noise level decreased AUCPR monotonically. Conversely, the increase of sample size and strength of SNP-to-gene signal positively contributed to the accuracy of reverse engineering (Figure 2A). Unlike the other network characteristics, gene-to-gene signal strength showed a nonmonotonic effect on AUCPR (Figure 2A). Initially, the increase in gene-to-gene signal strength had a beneficial effect on performance because stronger signals aided in distinguishing true regulations from noise. However, once the gene-to-gene signal strength exceeded a certain point, it showed the opposite effect (Figure 2A). Strength of gene-to-gene signal also showed significant interactions with other characteristics, including sample size, the number of genes, edge density, and network topology (Figure 2B). However, these interactions did not alter the nonmonotonic trend of the effect of gene-to-gene signal on AUCPR (Supporting Information, Figure S1).

We found that secondary effects of gene-to-gene signaling accounted for the nonmonotonic effect of gene-to-gene signal on AUCPR. We observed that high average gene-to-gene signal strength monotonically increased the correlations between directly connected genes and also those between indirectly connected genes (Figure 3A). These indirect effects of gene-to-gene signal strength were quantified by $M = \log10(\text{directCor}/\text{indirectCor})$ based on the average correlations of directly connected genes and those of indirectly connected genes. This M measure showed a nonmonotonic trend along with the amount of gene-to-gene signal strength (Figure 3A) and was well correlated with AUCPR (Figure 3B). This suggests that the poor network reconstruction performance against networks with high gene-to-gene signal is due to the low M of the data set through the increase of indirect signals.

Comparison of structural samplers

Next we investigated general performances of structural samplers. For rigorous quantification, both AUCPR and AUCROC were employed as performance measures. Based on these measures, c2PB, c3PB, and REV outperformed other samplers, whereas STR, a traditional sampler, showed the worst performance (Figure 4A). The main methodological difference between top-performing samplers c2PB, c3PB, and REV and STR is the magnitude of modifications to a network offered at each step. Specifically, REV and c2PB update parents of two nodes and c3PB updates parents of three nodes simultaneously at each step. By contrast, STR modifies only a single edge of a network at each step. Superior performance of c2PB, c3PB, and REV was quite robust over many types of networks tested (Figure S2), suggesting samplers that execute drastic network modifications at a single step can potentially generate more accurate network structures within a set time compared to methods that employ smaller updates.

The performance of the samplers interacted with various network and simulation parameters (Figure 4B). We found that sample size and MCMC chain type had practically important contributions to the performance of the samplers. The performance gain offered by larger sample size was not observed uniformly: a limited set of samplers, namely c2PB, c3PB, and REV, benefits most from larger sample size (Figure 4C). Note that these samplers are all able to execute drastic network modifications in a single step and also showed superior performance even for small sample sizes. Samplers such as STR and 1PB showed only slight or no performance gains with the increase of sample size (Figure 4C). This observation suggests method selection is crucial in benefiting from larger sample sizes. MCMC chain type also showed a significant interaction with structural samplers. We employed two types of MCMC chains: a single longer chain and multiple shorter chains. Our results indicated that STR, 2PB, 3PB, 4PB, c3PB, and c4PB performed better with a longer single chain, whereas 1PB, c2PB, and REV performed better with shorter multiple chains (Figure 4C). The differences between these two types of samplers could be generated by differences in the time required to reach a stationary distribution. Specifically, in the case of samplers with slow convergence such as STR, 4PB, and c4PB, each short chain was terminated before reaching the stationary distribution. Therefore, the average Bayesian

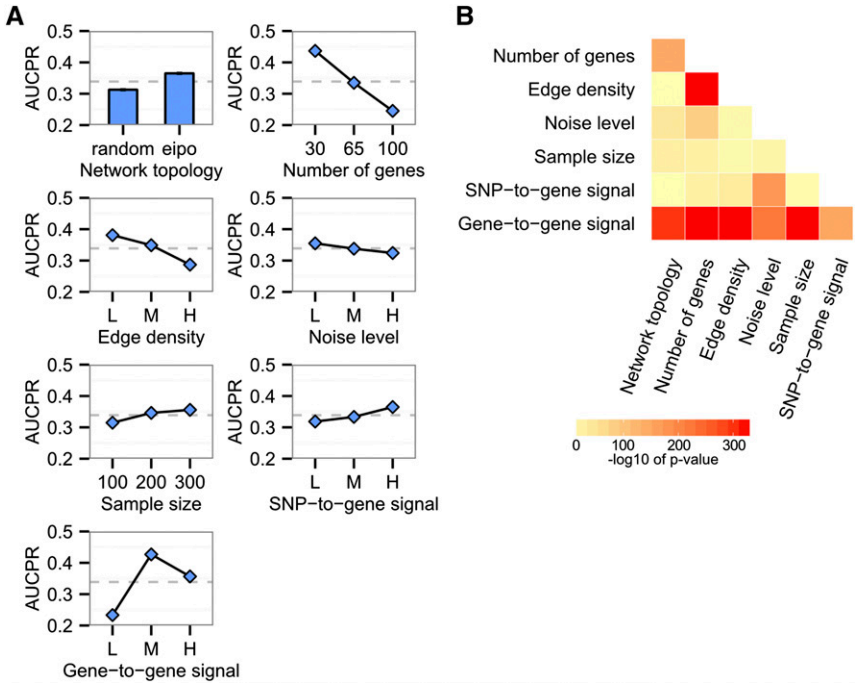

Figure 2 Effect of network features on the performance of Bayesian network reconstruction. (A) Marginal performance plots for each one of the seven simulation parameters. The dashed horizontal line represents the mean of the AUCPR distribution. (B) The heatmap represents significance for the interactions between any two pairs of the seven simulation parameters.

information criterion (BIC) of multiple chains is higher than that of a single chain (Figure S3), which resulted in lower AUCPRs. Conversely, the samplers including 1PB, c2PB, and REV that reach a stationary distribution faster would have potential benefits from exploring optimal networks over the large network space by employing shorter multiple chains from different initial networks. We note, however, that the minimum BIC of each sampler did not depend on the convergence speed (Figure S3). This suggests that some samplers with fast convergence speed such as 1PB can reach suboptimal networks quickly, but are not able to exit from the suboptimal configuration state.

To classify MCMC samplers based on their performance, we used hierarchical clustering analysis of AUCPR and AUCROC results (Figure 5). In this analysis, we treated a single longer chain and multiple shorter chains as separate methods. The clustering analysis based on the AUCPR result suggested MCMC samplers can be clustered into four types, which were driven by differences in performance on four primary types of data sets (Figure 5). Examination of network characteristics associated with the four groups of data sets revealed the number of genes, edge density, and gene-to-gene signal strength are major characteristics that differentiate the performance of various samplers (Figure S4A). In AUCPR-based clustering, 1PB, 2PB, and 3PB running with multiple shorter chains were in the same group and showed strong performance, especially for networks with high numbers of genes, high edge density, and high gene-to-gene signal strength. The top-performing samplers, c2PB, c3PB, and REV, were also clustered together and worked well across many simulations, except for networks with low gene-to-gene signals. STR and 4PB fell into the same cluster, showing poor performances over most network results. In the AUCROC-based clustering, we also observed four clusters of MCMC samplers and four groups of datasets (Figure 5). The three network characteristics, numbers of genes, edge density, and gene-to-gene signal strength were again associated with the four groups of data sets (Figure S4B). This further supports the large impact of these three network parameters on the accuracy of the reverse engineering of gene-regulatory networks. Similar to the AUCPR result above, 1PB, 2PB, and 3PB and c2PB, c3PB, and REV clustered into the same group (Figure 5). However, the strongest factor contributing to the cluster separation was the chain type employed. Running with multiple chains resulted in clearly higher AUCROC than with single-chain execution.

Detailed comparison of top-performing samplers

Next we investigated the differences among the top-performing samplers, REV, c2PB, and c3PB. To evaluate relative performance of any pair of structure samplers, A and B, we used single-replicate multiway ANOVA, with the response given by $W_{AB} = \text{AUCPR}_A - \text{AUCPR}_B$. We focus here on the AUCPR results from a single-chain condition to reveal differences in characteristics across samplers. Two parameters, gene-to-gene signal strength and network topology, had significant effects ($P < 1e\text{-}16$) on the performance of c3PB relative to REV (Figure 6). As the strength of gene-to-gene signaling increased, the c3PB performance improved considerably over that of REV. Also c3PB performed significantly better than REV for the networks with EIPO topology compared to networks with random topology. Compared to REV, c2PB also performed slightly better with increased gene-to-gene signaling (Figure 6). None of the other parameters were significantly associated with the difference between c2PB and REV, indicating these two samplers are almost comparable in terms of the type of networks where they work well. Finally, two parameters, gene-to-gene

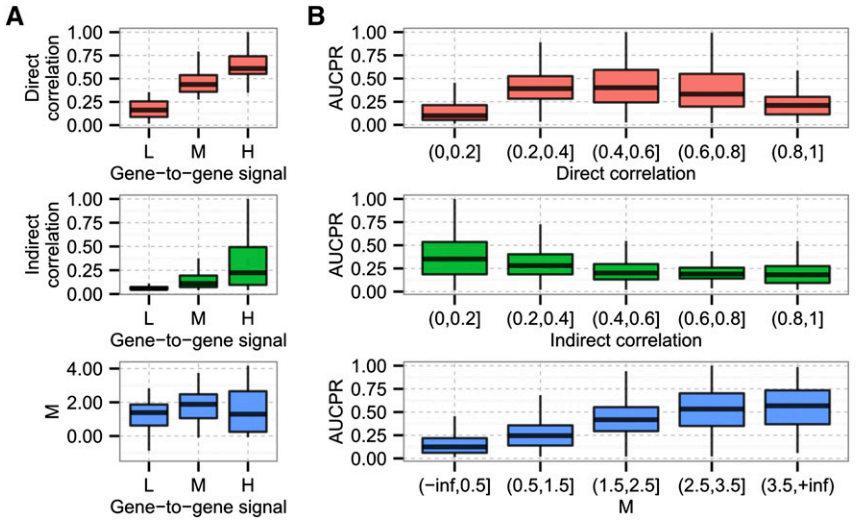

Figure 3 Tracking both direct and indirect gene-to-gene correlations explains how moderate correlations lead to the most accurate network inference. (A) Increasing the gene-to-gene signal strength increases the average correlation between genes, as expected. The strength of indirect correlations between genes also increases as gene-to-gene signal strength increases. Direct and indirect correlations are the average correlation of directly connected genes and indirectly connected genes, respectively. The *M* measure represents the log10 ratio between direct correlation and indirect correlation in each data set. (B) The bell-shaped response of AUCPR to increasing direct gene-to-gene correlations might be unexpected, but once the concurrent increase of indirect correlations is taken into account (through the *M* measure), the expected relationship between increasing signal and increased network accuracy is found (bottom right).

signal and network topology, are associated with the difference between c3PB and c2PB, just as they are between c3PB and REV (Figure 6). These results demonstrate that a newly designed sampler, c3PB, is superior to REV, especially for highly correlated biological networks.

Evaluation of the effect of genetic information

We designed our algorithm to incorporate SNPs as nodes in a Bayesian network, which means we estimated the structure of gene-to-gene networks and the eSNP mapping simultaneously. The incorporation of SNPs creates new sets of conditional independence relations, allowing us to distinguish between gene regulatory networks with equivalent likelihood. Furthermore, by including SNPs in the model we can potentially improve the model fit, leading to higher network recovery scores. We evaluated the benefit of incorporating SNP information for both the accuracy of the estimated network and the model fit to gene expression data, as follows. First, gene expression data and SNP data were simulated from networks composed of genes and SNPs. Then, gene expression data alone or both gene expression and SNPs were applied to the Bayesian network estimation programs.

Our results show that AUCPR and average BIC score significantly improve when both gene expression data and SNP data are used for reverse engineering (Figure 7A). We also investigated parameters that affected the performance gain when SNPs were incorporated (Figure 7B). As the strength of SNP-to-gene signal ramped up, the benefit of incorporating SNPs also increased the network reconstruction accuracy. Conversely, as the number of genes, edge density, and noise level increased, the beneficial effect of SNPs became more limited. The effects on inferred network accuracy that stem from incorporating SNPs are independent of MCMC sampler, sample size, and network topology. Our simulation study indicates that SNP information is often helpful to estimate Bayesian networks, but might be less effective on data sets generated by real networks that are large, are noisy, or have dense sets of interactions.

Discussion

Reverse engineering of gene networks from systems genetics data (Jansen 2001) is an active research area. Although we focused on MCMC samplers for gene network structure learning in the present article, many other statistical approaches/frameworks have been proposed in the literature. For instance, Chaibub Neto *et al.* (2008) applied the PC algorithm (Spirtes *et al.* 2000) to first infer the skeleton of the gene network and then used expression QTL (eQTL) to determine the directions of the edges in the phenotype network. Liu *et al.* (2008) proposed a two-step approach wherein an encompassing directed network (EDN) is first generated from assembled pairwise regulator–target relationships (which might be direct or indirect), followed by the application of structural equation models to search for a (possibly) cyclic network nested in the EDN, which best fits the data according to the BIC criteria. By restricting the network space to networks nested in the EDN, this approach is able to handle networks with hundreds of genes and eQTL. Logsdon and Mezey (2010) proposed a multiple-step approach where an association analysis is carried out to identify local eQTL, followed by the inference of an undirected network of gene expression traits and eQTL nodes via a covariance selection approach based on the adaptive lasso feature selection procedure and subsequent mapping of the undirected network into a possibly cyclic gene expression network. Zhang and Kim (2014) adopted sparse conditional Gaussian graphical models for modeling undirected gene networks under SNP perturbations, where the gene network structure learning task and the SNP feature selection task are performed jointly by solving a single optimization problem containing the lasso (for SNP feature selection) and the graphical lasso (for gene network structure learning) as special cases.

A common feature of these non-Bayesian approaches is that the output of the reverse-engineering algorithm is a single point estimate of the gene network, and no statistical measure of uncertainty about the inferred network is available. The MCMC samplers, on the other hand, output an entire posterior distribution of network structures, from which Bayesian

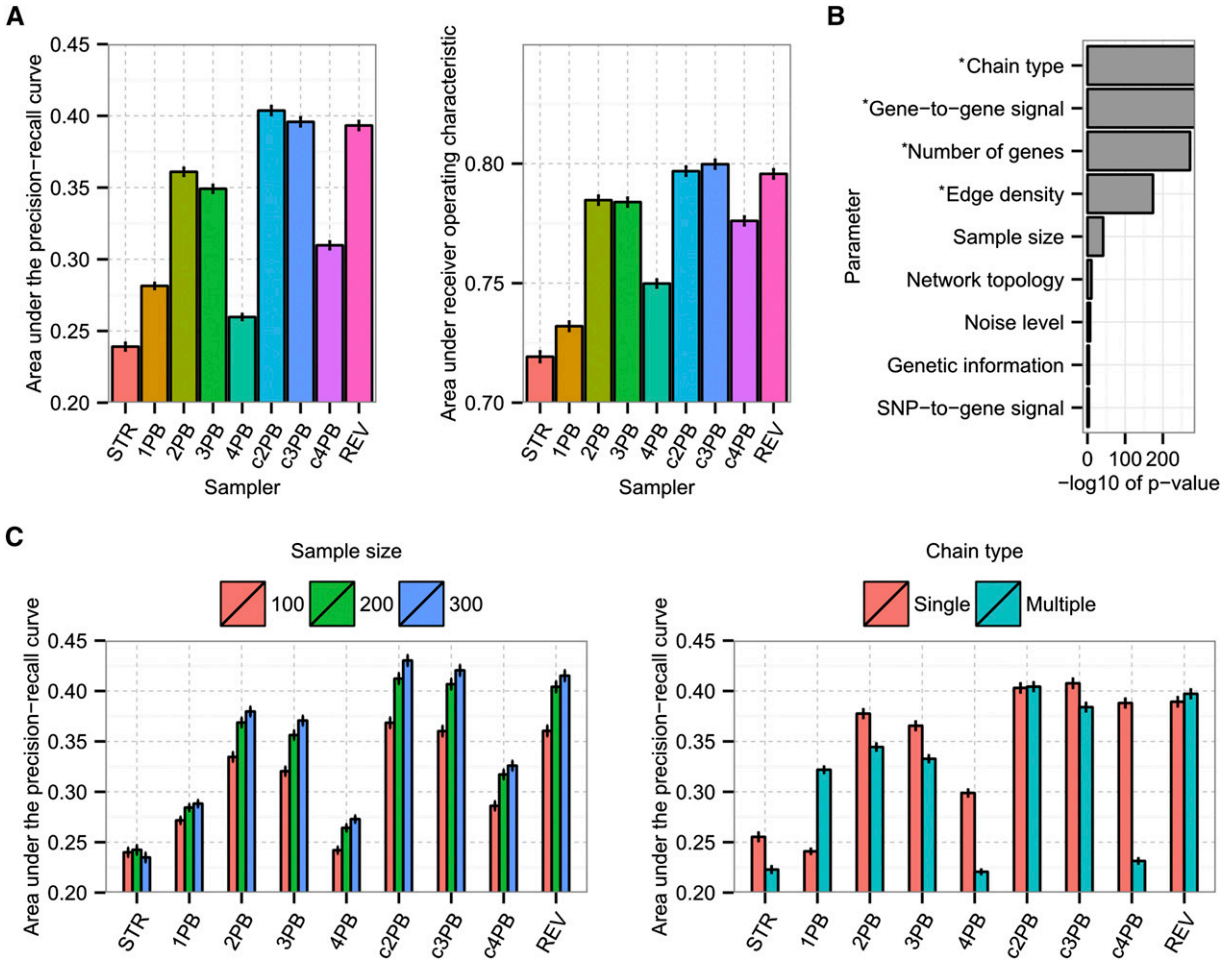

Figure 4 Performance comparison of network structure samplers. (A) Marginal performance plots for the nine MCMC samplers. (B) Significance of interaction between MCMC samplers and the nine simulation parameters. Asterisks indicate the variables that are significant at *P*-value <10*E*-16. (C) The interaction plots for the nine MCMC samplers and two simulation parameters: sample size and MCMC chain type.

model-averaged estimates of the probability for the presence and direction of any particular edge in the network are readily available. This provides information about which parts of the inferred network were reconstructed with stronger confidence. This feature is particularly important for reverse engineering of gene networks since expression data are notoriously noisy. The main drawback is the increased computational requirements, compared to non-Bayesian/point estimate approaches, which are generally faster and scalable to larger networks. Nonetheless, MCMC approaches have been successfully applied to genome-scale real data (Zhu *et al.* 2008; Zhang *et al.* 2013) by first clustering the gene expression data into much smaller groups via weighted gene coexpression analysis (Zhang and Horvath 2005), followed by Bayesian networks reconstruction for the separate and more manageable gene clusters.

MCMC samplers based on moves in the space of node orders (Friedman and Koller 2003; Eaton and Murphy 2007; Ellis and Wong 2008) have been shown to considerably improve the mixing of the Markov chain. Nonetheless, these samplers do not allow the explicit specification of prior distributions over network structures. Since our main applied interest is the reconstruction of Bayesian networks with noisy genomic data, where the incorporation of prior knowledge can improve reconstruction performance, we focus our attention on MCMC approaches based on moves in network structure space.

To investigate network inference in the context of data sets that are likely encountered by experimental biologists, we conducted the largest simulation study comparing MCMC samplers for structure learning in Bayesian networks to date. By ranging over combinations of biologically plausible parameter settings, we can be confident in our conclusions about the relative performance of different inference methods. The variability in generative network structures also allows us to understand how network characteristics affect the performance of various network inference methods. Specifically, our simulation study is designed in the spirit of a multifactorial experiment with crossed factors, where simulation parameters play the role of factors. This allows us to investigate the effect of each parameter on the accuracy of Bayesian networks, in addition to comparing performance of MCMC samplers.

The average edge density and the average gene-to-gene signal significantly affected the learning performance. As the

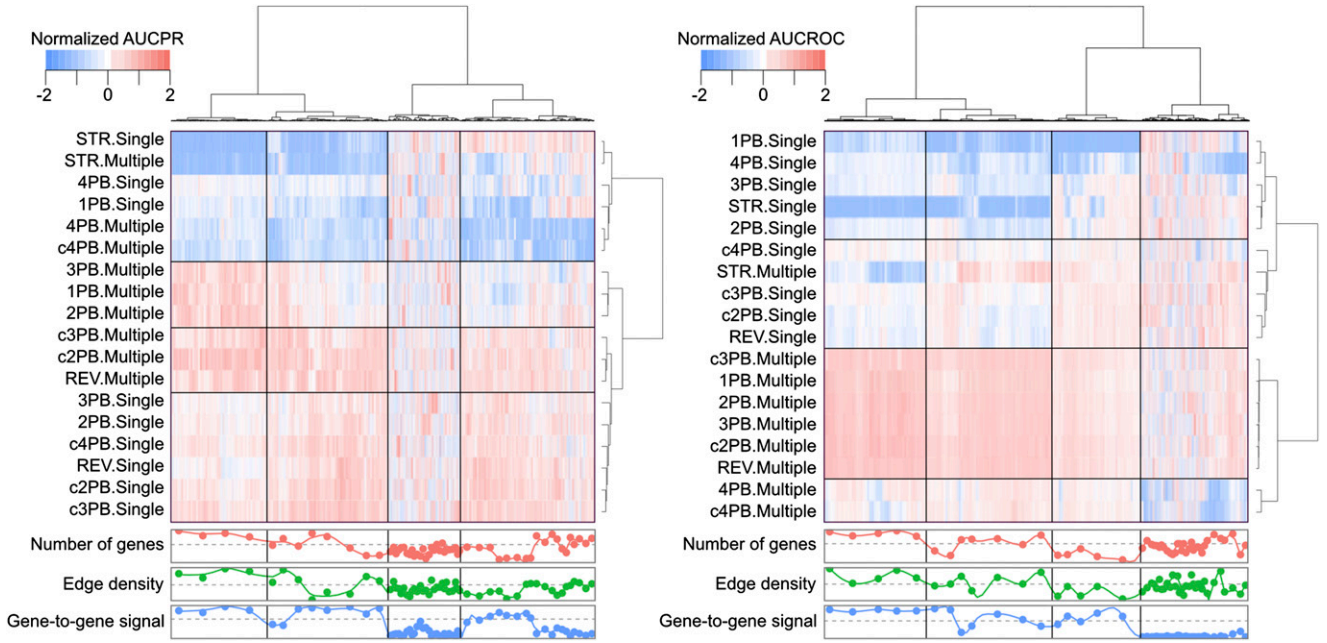

Figure 5 Cluster analysis for MCMC samplers. Top panels contain the heatmaps of the normalized AUCPR and AUCROC and associated dendrograms obtained through hierarchical clustering of the results. The rows indicate MCMC samplers running either single chain or multiple chains. The columns indicate simulated networks. Bottom panels represent the number of genes, edge density, and gene-to-gene signal averaged across networks clustered together in the dendrograms.

average edge density increased, the learning performance of every method decreased monotonically. One possible reason for this behavior is that for Bayesian network inference, we restricted the number of parents to $< = 3$, to reduce the size of the network space to evaluate. On the other hand, we did not pose any restriction on the number of parents in simulated networks where the maximum edge density was set as 0.06, which means each node in networks with size 30, 65, and 100, is expected to have 1.8, 3.9, and 6 parents on average, respectively, assuming random network topology. Because of this limitation, it is possible that we miss some true incoming regulatory relationships for genes that are regulated by many other genes. However, in real applications, since screening the genes with many children is likely to be a predominant question of interest, this limitation may not be critical. Another reason for the generally harmful effect of edge density on performance could be the correlation strength of the data set. In our simulation framework, the average gene-to-gene correlation is influenced by edge density and the strength of gene-to-gene signaling. Generally, it is difficult to estimate true regulations in a highly correlated data set, since the Markov equivalence classes of a directed network cannot be identified uniquely (Uhler *et al.* 2013). STR, the simplest sampler, showed the worst performance against networks with high edge density and high gene-to-gene signal, probably due to being trapped in suboptimal structures. This limitation has practical consequences for recent attempts to define causal networks within gene coexpression networks, since those coexpression networks are composed of correlated gene variables. However, our new sampler, c3PB, showed better performance compared with STR as well as REV, for the networks with high gene-to-gene signal, indicating c3PB works well for elucidating complex and correlated biological networks.

Our multifactorial simulation was designed to cover many biologically plausible conditions, but there still exist a few common assumptions underlying the simulation and the inference that might bias our comparison results. For instance, during the Bayesian network estimation process, we restricted the number of parents per node. Since the amount of computation required for each MCMC iteration differs greatly among the samplers, this restriction may affect the performance of each sampler differently. Specifically, since REV and higher-order PBs samplers need to calculate all the parent set combinations in every MCMC iteration, the number of parents affects the number of iterations completed in a fixed time window. On the other hand, the STR sampler does not need to keep track of the parent sets. However, the inherent poor mixing of the STR sampler becomes an even more important issue when no restriction on the number of parents is imposed, as the sampler needs to search over a larger network space and a larger number of iterations is required for the convergence of the chain. Another assumption in our study is that 20% of the genes have eSNPs, and no more than two SNPs per gene are allowed. While these conditions are based on characteristics of real data (Stranger *et al.* 2012), they still influence the extent of the SNP's contribution to the gene regulatory system, and different samplers might respond differently to the distinct amounts of genetic influence. One factor that partially mitigates this limitation is that in addition to SNP frequency, other

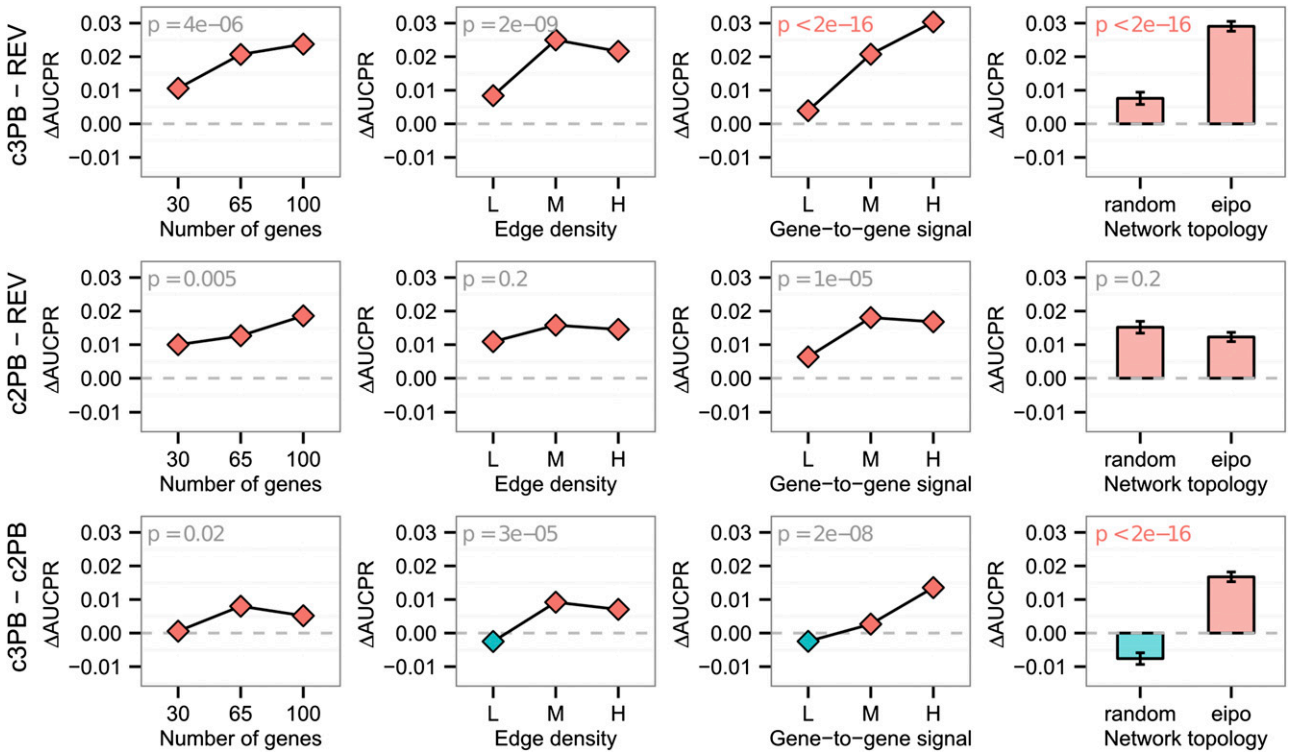

Figure 6 Pairwise comparison of top-performing samplers. Shown are marginal effect plots for the comparison between REV, c2PB, and c3PB with a single chain running. The dashed horizontal line represents a baseline representing equivalent performance.

factors such as the strength of the SNP-to-gene associations also contribute to the total percentage of variance explained by SNPs in real systems. While we fix the number of genes with eSNPs, we still examine a range of scenarios for percentage of variance explained by SNPs, because we utilize three levels of SNP-to-gene coupling.

Zhu *et al.* (2007) provided a detailed simulation investigation of the effect of the integration of genetic and expression data in Bayesian network reconstruction performance. They focused on simulations from a single (biologically motivated) network structure, in contrast to our present study investigating many characteristics across 1458 distinct and biologically plausible networks. In Zhu *et al.* (2007), the authors concluded that the integration of SNP and expression data can greatly improve network reconstruction. In our simulations, we also observed that the genetic information has beneficial effects, but found that the effect size for genetics is smaller than the effect of other parameters, such as the particular method employed. We further clarified the benefit of genetics by showing that the advantage of including genetics was maximized when the strength of SNP-to-gene signal was high and the intrinsic noise was low. Signal strength and noise level actually determine the extent of SNPs' contribution to gene expression levels. The middle range of signal strength and noise level corresponds to an average effect size of eSNPs commonly observed in real data. At this level of effect size, improper selection of an inference method and running setting of the MCMC chain potentially diminishes the benefit of utilizing SNP information. Alternatively, for systems where SNPs strongly influence expression, incorporating SNP data can produce a greater marginal increase in performance than the choice of MCMC method. For instance, incorporation of SNP data will likely be most beneficial for causal network inference of gene expression phenotypes clustered in strong hotspots.

In our generative model, single SNPs regulated the expression level of single genes. This setting might underestimate the contribution of SNP to gene expression, since pleiotropic effects of SNPs have been noted (Wagner and Zhang 2011) and a single SNP might be able to regulate multiple genes directly through remodeling of chromatin structure and sharing functional DNA motifs. From the viewpoint of eSNP identification, the mapping of SNPs in the context of gene-to-gene networks potentially increases power and reduces false positives for detecting eSNPs (Chaibub Neto *et al.* 2010). Latent variables such as influences from other genes or experimental conditions can mask true signals from SNPs. Stegle *et al.* (2010) proposed the removal of such effects from the expression data prior to eSNP analysis. SNP-incorporated Bayesian networks jointly infer the effects of SNPs and latent effects from other genes and thus are promising tools for investigating genetic architecture.

In this study, we extended the block Gibbs samplers proposed by Goudie and Mukherjee (2011) to, in theory, include an arbitrary number of parent sets. As a practical demonstration, we test one to four parent sets, to understand how higher-order blocking contributes to performance. While the performances of 2PB and 3PB are clearly superior to that of

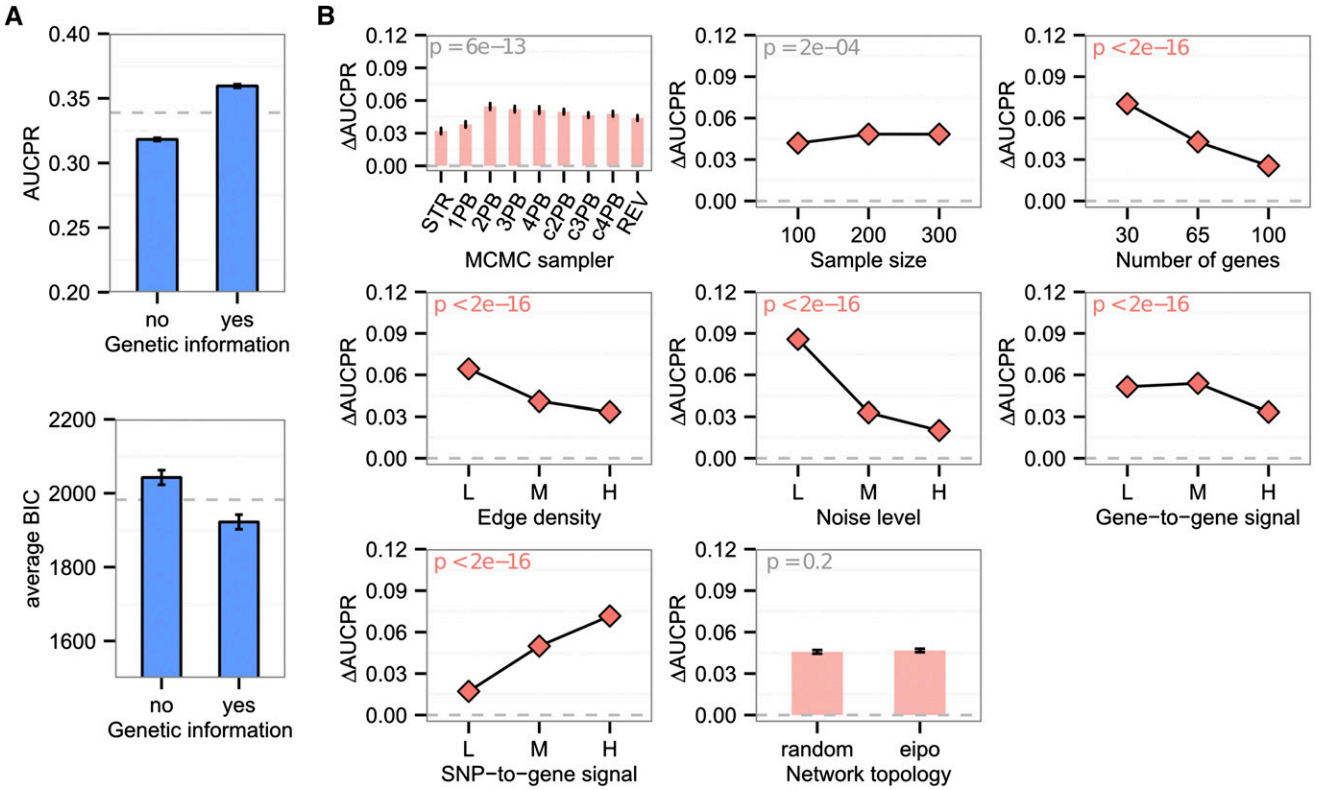

Figure 7 Effect of genetic information on network accuracy. (A) Marginal AUCPR and average BIC plots for the use of genetic information. The average BIC is calculated as mean of BIC of sampled networks in each MCMC run (lower BIC scores are better). (B) Marginal effect plots for the comparison between the Bayesian network reconstruction based on gene expression and SNP information *vs.* gene expression only. The dashed horizontal line represents a baseline representing equivalent performance. ΔAUCPR is defined as the difference of all AUCPRs from simulations with genetics minus all AUCPRs from simulations without genetics.

1PB, the performance of 4PB decreased substantially (Figure 4A). Although the use of higher-order parent blocks improves the mixing of the Markov chain, it also exponentially increases the amount of computation necessary to keep track of the allowable combinations of parent sets, so that the number of steps completed by the sampler in a fixed time window is decreased. Hence, in spite of the better mixing of the chain, we still observe a performance drop. These results indicate that both drastic modifications for escape from suboptimal networks and computational efficiency are important in effective sampler operation.

We designed "connected" block Gibbs methods, as a hybrid of the classic block Gibbs and REV methods, motivated by limitations of the older methods exposed by our simulation framework. At a given time step, both REV and 2PB update the parent sets of two nodes. However, the performance of 2PB was inferior to that of REV (Figure 4A). This result was unexpected because Gibbs samplers accept network modification at every step and thus are potentially more efficient than a Metropolis–Hastings sampler. A major difference between 2PB and REV that accounts for this is that 2PB updates parent sets of every pair of two nodes sequentially, whereas REV updates only parent sets of the two connected nodes. Therefore, we designed c2PB, which performs 2PB updates for only the two connected nodes at each step and also extended c2PB to ones with higher-order blocking, namely c3PB and c4PB. Figure 4A shows our connected PB samplers clearly improve the formal PB Gibbs samplers, resulting in comparable or better performance than that of REV. The performance improvements of cPB likely result from two sources. First, the speed of chain convergence is higher in cPBs compared with corresponding PBs (Figure S3). Second, cPBs can reach networks with lower BIC, meaning optimized network structures explain the data set better (Figure S3). These improvements are also evident when comparing c2PB with REV. cPBs can be seen as weighted versions of a PB sampler. Specifically, each gene is not updated with equal frequency, but rather the update probability for each gene is weighted based on the current network structure. Since cPBs weight all linearly connected gene pairs uniformly, a more complex weighting procedure may be useful to explore in the future.

The relatively close performance of top-performing methods c2PB, c3PB, and REV (Figure 4) compared to other methods raises the possibility of a performance ceiling in Bayesian network reconstruction for biological data sets. While performance among these top methods differs substantially for certain classes of networks, does their collective performance indicate anything about the future of network inference? The issue at the heart of performance limitations on network inference is finding reasonable accurate network structures amid

the enormous space of possible networks, which is superexponential with the number of nodes. For instance, networks from our "medium-size" simulations containing 65 nodes have a number of possible configurations that exceed the estimated number of atoms in the observable universe. Both algorithmic advances in network update steps or brute-force computational approaches could be used to improve network inference, with more efficient or more comprehensive approaches to exploring this huge search space. Our method comparison primarily focuses on algorithmic advances, although computational constraints also influenced the results, especially for inefficient methods. From a computational perspective, starting from many different random networks can help to avoid becoming stuck in local minima, as can performing more drastic update moves. Our results for single *vs.* multiple chains indicate that 1PB, c2PB, c3PB, and REV methods that converge rapidly will benefit the most from a massively parallel approach to finding optimal networks. But results from the classic STR method demonstrate that even an excess of computational resources is not sufficient to overcome a method that is easily trapped in local optima. At the same time, limitations on computational capabilities can hold back algorithms that should produce superior performance. For instance, we see increasing performance across 1PB, 2PB, and 3PB, yet performance of 4PB falls off, as computational overhead overcomes its theoretical advantages. Therefore, based on this simulation study, our general advice for future applications of the methods tested here is to devote substantial computational resources to the latest methods, such as c3PB, that are able to benefit from these resources. If a data set is known to have particular characteristics such as high levels of noise, etc., then the recommended method will shift to one that is known to function well under that regime, although we note that some methods such as c3PB function well under a majority of tested scenarios.

One practical goal of network inference methods in early-stage drug discovery is to identify a small number of genes that control a specific molecular network or a disease signature. Perturbation experiments in model disease systems indicate that causal networks can indeed predict some disease-relevant genes and downstream targets (Chen *et al.* 2008; Yang *et al.* 2009). While large-scale causal network inference has been most frequently applied to data from mouse crosses whose common genetic background creates strong eQTL, these methods have also been applied to paired genetics and gene expression data sets from humans (Zhang *et al.* 2013). However, these previous applications employed variants of the STR method, which is outperformed by all other methods tested here. While previous studies have made some correct predictions, there are also numerous false positive and false negative predictions, based on comparing the expected *vs.* actual (measured) downstream targets of perturbed genes in model systems (Zhang *et al.* 2013). The greater accuracy of top-performing methods is therefore key to deriving accurate predictions from human expression data sets, with or without genetics.

Because many gene expression studies do not include paired genetics data, correlation-based coexpression networks are frequently used to attempt to identify key disease genes (Gaiteri *et al.* 2014). Specifically, hub genes within disease-correlated networks are often invoked as key disease genes. However, coexpression hubs are not necessarily causal in generating a phenotype of gene signature, because a gene at the top of a regulatory cascade may have only a few direct interactions. Such genes may be ignored when focusing on coexpression hub genes. Similarly, hub genes may be subject to incoming regulatory relationships, which cannot be detected in an undirected coexpression framework. Our simulations show that for typical human data sets, even in the absence of genetic priors, we can identify conditional dependence networks of genes that accurately reflect the true regulatory structure. This sparse hierarchical output could be utilized to aid in selecting genes that control a particular disease-relevant molecular system. Causal networks could even be useful in defining relationships across different diseases, to find directed interactions that mediate comorbidity. However, such data sets likely involve >100 genes, which entails a large network search space. The potential of causal inference to identify human regulatory networks among a large number of genes, with or without genetic priors, reinforces the need to use top-performing methods that do not become trapped in suboptimal/inaccurate network states.

Literature Cited

Aten, J. E., T. F. Fuller, A. J. Lusis, and S. Horvath, 2008 Using genetic markers to orient the edges in quantitative trait networks: the NEO software. BMC Syst. Biol. 2: 34.

Beinlinch, I., H. Suermondt, R. Chavez, and G. Cooper, 1989 The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks, pp. 247–256 in *In Second European Conference on Artificial Intelligence in Medicine*, edited by J. Hunter, J. Cookson, and J. Wyatt. Springer-Verlag, Berlin.

Brown, C. D., L. M. Mangravite, and B. E. Engelhardt, 2013 Integrative modeling of eQTLs and cis-regulatory elements suggests mechanisms underlying cell type specificity of eQTLs. PLoS Genet. 9: e1003649.

Chaibub Neto, E., C. T. Ferrara, A. D. Attie, and B. S. Yandell, 2008 Inferring causal phenotype networks from segregating populations. Genetics 179: 1089–1100.

Chaibub Neto, E., M. P. Keller, A. D. Attie, and B. S. Yandell, 2010 Causal graphical models in systems genetics: a unified framework for joint inference of causal network and genetic architecture for correlated phenotypes. Ann. Appl. Stat. 4: 320–339.

Chaibub Neto, E., A. T. Broman, M. P. Keller, A. D. Attie, B. Zhang *et al.*, 2013 Modeling causality for pairs of phenotypes in system genetics. Genetics 193: 1003–1013.

Chen, L. S., F. Emmert-Streib, and J. D. Storey, 2007 Harnessing naturally randomized transcription to infer regulatory relationships among genes. Genome Biol. 8: R219.

Chen, Y., J. Zhu, P. Y. Lum, X. Yang, S. Pinto *et al.*, 2008 Variations in DNA elucidate molecular networks that cause disease. Nature 452: 429–435.

Duarte, C. W., and Z.-B. Zeng, 2011 High-confidence discovery of genetic network regulators in expression quantitative trait loci data. Genetics 187: 955–964.

Eaton, D., and K. Murphy, 2007 Bayesian structure learning using dynamic programming and MCMC. Proceedings of the 23rd

Annual Conference on Uncertainty in Artificial Intelligence (UAI-07), Corvallis, OR, pp. 101–108.
Ellis, B., and W. H. Wong, 2008 Learning causal Bayesian network structures from experimental data. J. Am. Stat. Assoc. 103: 778–789.
Ferrara, C. T., P. Wang, E. Chaibub Neto, R. D. Stevens, and J. R. Bain *et al.*, 2008 Genetic networks of liver metabolism revealed by integration of metabolic and transcriptional profiling. PLoS Genet. 4: e1000034.
Friedman, N., and D. Koller, 2003 Being Bayesian about network structure. Mach. Learn. 50: 95–126.
Gaiteri, C., Y. Ding, B. French, G. C. Tseng, and E. Sibille, 2014 Beyond modules and hubs: the potential of gene coexpression networks for investigating molecular mechanisms of complex brain disorders. Genes Brain Behav. 13: 13–24.
Giudici, P., and R. Castelo, 2003 Improving Markov chain Monte Carlo model search for data mining. Mach. Learn. 50: 127–158.
Goudie, R. J. B., and S. Mukherjee, 2011 An efficient Gibbs sampler for structural inference in Bayesian networks. Paper no. 11-21. Center for Research in Statistical Methodology. Coventry, United Kingdom. Available at: www.warwick.ac.uk/go/crism.
Grundberg, E., K. S. Small, A. K. Hedman, A. C. Nica, A. Buil *et al.*, 2012 Mapping cis- and trans-regulatory effects across multiple tissues in twins. Nat. Genet. 44: 1084–1089.
Grzegorczyk, M., and D. Husmeier, 2008 Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. Mach. Learn. 71: 265–305.
Guelzim, N., S. Bottani, P. Bourgine, and F. Képès, 2002 Topological and causal structure of the yeast transcriptional regulatory network. Nat. Genet. 31: 60–63.
Hageman, R. S., M. S. Leduc, R. Korstanje, and B. Paigen, and G. A. Churchill, 2011 A Bayesian framework for inference of the genotype-phenotype map for segregating populations. Genetics 187: 1163–1170.
Huang, T. L., P. P. Zandi, K. L. Tucker, A. L. Fitzpatrick, L. H. Kuller *et al.*, 2005 Benefits of fatty fish on dementia risk are stronger for those without APOE epsilon4. Neurology 65: 1409–1414.
Jansen, R., 2001 Genetical genomics: the added value from segregation. Trends Genet. 17: 388–391.
Kass, R. E., and AE. Raftery, 1995 Bayes factors. J. Am. Stat. Assoc. 90: 773–795.
King, V., and G. Sagert, 2002 A fully dynamic algorithm for maintaining the transitive closure. J. Comput. Syst. Sci. 65: 150–167.
Liu, B., A. de la Fuente, and I. Hoeschele, 2008 Gene network inference via structural equation modeling in genetical genomics experiments. Genetics 178: 1763–1776.
Liu, M., A. Liberzon, S. W. Kong, W. R. Lai, P. J. Park *et al.*, 2007 Network-based analysis of affected biological processes in type 2 diabetes models. PLoS Genet. 3: e96.
Logsdon, B. A., and J. Mezey, 2010 Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations. PLoS Comput. Biol. 6: e1001014.
Madigan, D., J. York, and D. Allard, 1995 Bayesian graphical models for discrete data. Int. Stat. Rev. 63: 215.
McKenzie, M., A. K. Henders, A. Caracella, N. R. Wray, and J. E. Powell, 2014 Overlap of expression quantitative trait loci (eQTL) in human brain and blood. BMC Med. Genomics 7: 31.
Moon, J. Y., E. Chaibub Neto, B. S. Yandell, and X. Deng, 2014 Bayesian causal phenotype network incorporating genetic variation and biological knowledge, pp.165–195 in, *Probabilistic Graphical Models in Genetics, Genomics, and Postgenomics*, edited by C. Sinoquet, and R. Mourad. Oxford University Press, London/New York/Oxford.
Pearl, J., 1988 *Probabilistic Inference in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA.
Peila, R., B. L. Rodriguez, and L. J. Launer, 2002 Type 2 diabetes, APOE gene, and the risk for dementia and related pathologies: The Honolulu-Asia Aging Study. Diabetes 51: 1256–1262.
Pinna, A., N. Soranzo, I. Hoeschele, and A. de la Fuente, 2011 Simulating systems genetics data with SysGenSIM. Bioinformatics 27: 2459–2462.
Rhinn, H., R. Fujita, L. Qiang, R. Cheng, J. H. Lee *et al.*, 2013 Integrative genomics identifies APOE ε4 effectors in Alzheimer's disease. Nature 500: 45–50.
Schadt, E. E., J. Lamb, X. Yang, J. Zhu, S. Edwards *et al.*, 2005 An integrative genomics approach to infer causal associations between gene expression and disease. Nat. Genet. 37: 710–717.
Spirtes, P., C. Glymour, and R. Scheines, 2000 *Causation, Prediction, and Search*. MIT Press Cambridge, MA
Stegle, O., L. Parts, R. Durbin, and J. Winn, 2010 A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. PLoS Comput. Biol. 6: e1000770.
Stranger, B. E., S. B. Montgomery, A. S. Dimas, L. Parts, O. Stegle *et al.*, 2012 Patterns of cis regulatory variation in diverse human populations. PLoS Genet. 8: e1002639.
Uhler, C., G. Raskutti, P. Bühlmann, and B. Yu, 2013 Geometry of the faithfulness assumption in causal inference. Ann. Stat. 41: 436–463.
Veyrieras, J.-B., S. Kudaravalli, S. Y. Kim, E. T. Dermitzakis, Y. Gilad *et al.*, 2008 High-resolution mapping of expression-QTLs yields insight into human gene regulation. PLoS Genet. 4: e1000214.
Wagner, G. P., and J. Zhang, 2011 The pleiotropic structure of the genotype-phenotype map: the evolvability of complex organisms. Nat. Rev. Genet. 12: 204–213.
Yang, S., Y. Liu, N. Jiang, J. Chen, L. Leach *et al.*, 2014 Genome-wide eQTLs and heritability for gene expression traits in unrelated individuals. BMC Genomics 15: 13.
Yang, X., J. L. Deignan, H. Qi, J. Zhu, S. Qian *et al.*, 2009 Validation of candidate causal genes for obesity that affect shared metabolic pathways and networks. Nat. Genet. 41: 415–423.
Zhang, B., and S. Horvath, 2005 A general framework for weighted gene co-expression network analysis. Stat. Appl. Genet. Mol. Biol. 4: 1–45.
Zhang, B., C. Gaiteri, L.-G. Bodea, Z. Wang, J. McElwee *et al.*, 2013 Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease. Cell 153: 707–720.
Zhang, L., and S. Kim, 2014 Learning gene networks under SNP perturbations using eQTL datasets. PLoS Comput. Biol. 10: e1003420.
Zhu, J., M. C. Wiener, C. Zhang, A. Fridman, E. Minch *et al.*, 2007 Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. PLoS Comput. Biol. 3: e69.
Zhu, J., B. Zhang, E. N. Smith, B. Drees, R. B. Brem *et al.*, 2008 Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. Nat. Genet. 40: 854–861.

Communicating editor: I. Hoeschele

Appendix

Bayesian Networks Background and Notation

A *Bayesian network* (Pearl 1988) is a multivariate probabilistic model whose conditional independence relations can be represented graphically by a directed acyclic graph (DAG) with *vertices* $V = (V_1, ..., V_p)$ and *directed edges* $(i,\ j) \in E \subset V \times V$ (note that we use the notations i and V_i interchangeably, to refer to a node). If $(i, j) \in E$, we say i is a *parent* of j and j is a *child* of i. We define a *directed path* as any unbroken, nonintersecting sequence of vertices in a graph that go along the direction of the edges. We say that a *descendant* of a vertex i is any vertex j such that there is a directed path from i to j, whereas a *nondescendant* of i is any vertex k such that there is no directed path from i to k. A vertex j in a DAG G corresponds to a random variable X_j in the Bayesian network. Assuming the local directed Markov property that states that each variable is independent of its nondescendant variables conditional on its parent variables, we can factor the joint distribution as

$$P(X \mid G) = \prod_{j=1}^{p} P(X_j \mid X_{G_j}), \tag{A1}$$

where $X = (X_1, ..., X_p)^T$, G_j is the set of parents of j, and $X_{G_j} = \{X_i : i \in G_j\}$. Note that G can be decomposed into a set of parent sets such that $G = (G_1, ..., G_p) = (G_j, G_{-j})$, where $G_{-j} = (G_1, ..., G_{j-1}, G_{j+1}, ..., G_p)$, and that the prior predictive distribution in (A1) factorizes across vertices into local components $P(X_j \mid X_{G_j})$ that are functions of these parent sets. The network scores are computed according to the BIC approximation to the prior predictive distribution (Kass and Raftery 1995). For structure learning, we focus on the posterior distribution of DAG structures

$$P(G \mid X) = \frac{P(X \mid G)P(G)}{\sum_{G \in \mathcal{G}} P(X \mid G)P(G)}, \tag{A2}$$

where $P(G)$ is a prior on the network structure G, and $\mathcal{G}$ represents the space of all DAGs with p vertices. This posterior is the target equilibrium distribution for the MCMC samplers described in the next section.

Checking for Cycles

The main computational cost for structure samplers is due to checking for cycles. Following Goudie and Mukherjee (2011), we adopt the algorithm from King and Sagert (2002), for cycle checking, in our implementations. This algorithm tracks the transitive closure of the current state of the sampler, represented by matrix T^G, which, for a graph G, is represented as the directed graph (V, E^*), where $(V_i, V_j) \in E^*$ if and only if a path exists from V_i to V_j. Inspection of the adjacency matrix of the transitive closure shows which network modifications can be made without introducing a cycle in the following way: the addition of an edge (i,j) will introduce a cycle if and only if $T^G_{ji} = 1$, whereas the removal of an edge can never introduce a cycle. An efficient implementation of this algorithm keeps track of a path count matrix C^G (with rows indexing parents, columns indexing children, off-diagonal entries representing the number of distinct paths from V_i to V_j in G, and diagonal entries set to 1). Observe that $T^G_{ij} = 1$ if and only if $C^G_{ij} > 0$, and query operations can be performed by simply checking whether the relevent entries are positive. As pointed by Goudie and Mukherjee (2011), updating C^G is straightforward. Consider a graph G' formed by adding an edge (i,j) to graph G. Let $C^G_{\cdot i}$ represent the ith column of C^G and $C^G_{j\cdot}$ represent the jth row of C^G. Then the updated count matrix for the addition of an edge (i,j) is computed as $C^{G'} = C^G + C^G_{\cdot i} \otimes C^G_{j\cdot}$, whereas the updated count matrix after the deletion of an edge (i,j) is computed as $C^{G'} = C^G - C^G_{\cdot i} \otimes C^G_{j\cdot}$.

Summary Characteristics of Structure MCMC Samplers for Learning Bayesian Networks

The STR sampler (Madigan et al. 1995; Giudici and Castelo 2003)

The STR sampler performs simple modifications to the current DAG state. At each iteration, it adds, drops, or reverses a single edge of the network and accepts the modified network according to the standard Metropolis–Hastings acceptance probability. Each time an edge is added or reversed, it is necessary to check whether the new network has a cycle, and if it does, the move needs to be discarded. [This sampler was proposed by Giudici and Castelo 2003 as an improvement over the Markov chain Monte Carlo model composition (MC^3) sampler proposed by Madigan *et al.* 1995 that considered only addition and deletion moves.] A consequence of using the simplest possible network alterations that are made without consideration for the larger structure of the network around them is that the mixing of the Markov chain is usually very slow, and the chain tends to get trapped in local maxima.

The REV sampler (Grzegorczyk and Husmeier 2008)

The REV sampler adopts an alternative edge reversal move ("REV move") to improve the mixing of the Markov chain. The steps to perform this network alteration are (i) select an edge whose direction is to be reversed; (ii) for each of the nodes connected by the selected edge, drop the incoming edges (that is, "orphan" the nodes); (iii) reverse the direction of the selected edge; and (iv) sample new parents for the nodes involved in the edge reversal (in addition to keeping the reversed edge) with a probability proportional to their scores. Note that contrary to the STR sampler, which allows the reversal of an edge only if it leads to a new valid DAG, the REV move always leads to a DAG. Even for those edges that could be reversed by the STR reversal move, it is still advantageous to use the REV move since it usually leads to higher acceptance rates. It does this by sampling completely new parent sets, instead of simply changing the direction of a single edge. Because the adoption of the REV move alone does not guarantee the ergodicity of the Markov chain, the REV sampler is actually implemented as a combination of REV and STR moves. In this article we adopt a sampler with REV moves performed in 50% of the iterations.

The single-parent set block Gibbs sampler—1PB (Goudie and Mukherjee 2011)

Because our novel samplers are extensions of the block Gibbs samplers proposed by Goudie and Mukherjee (2011), we present their method in greater detail in this section and the next section. For Bayesian networks the most natural blocks are given by the parent sets of each node. Because the prior predictive distribution factorizes according to each node and its parent set, we can parameterize a DAG G according to the parent sets G_j of each vertex $j = 1, ..., p$, such that the posterior distribution of G is represented by

$$P(G_1, \ldots, G_p \,|\, X) \propto P\big(G_1, \ldots, G_p\big) \prod_{j=1}^{p} P(X_j \,|\, X_{G_j}). \tag{A3}$$

To construct a block Gibbs sampler over the parent set blocks, we need to construct the full conditional distributions of each parent set, conditional on all other parent sets. Because Bayesian networks are DAGs, we have that parent sets G_j, for which $G = (G_j, G_{-j})$ is cyclic, must have probability zero, and the full conditional distribution of G_j given G_{-j} is given by

$$P(G_j \,|\, G_{-j}, X) = \frac{P(G_j, G_{-j} \,|\, X)}{\sum_{G_j \in K_j^*} P(G_j, G_{-j} \,|\, X)}, \tag{A4}$$

where K_j^* represent the set of parent sets G_j such that G is acyclic. The computation of K_j^* can be done efficiently, using the path count matrix. Recall that adding an edge (i, j) will introduce a cycle if and only if $C_{ji}^G > 0$. Therefore, the set of nodes that can be added as parents of V_j is given by the set $K_j = \{V_i : C_{ji}^G = 0\}$ Since any subset of K_j can also be added as parents of V_j, we have that $K_j^* = \mathcal{P}(K_j)$, the power set of K_j.

In this article we assume a uniform prior for network structures and adopt the BIC approximation for the marginal likelihood (Kass and Raftery 1995) so that

$$P\big(G_j, G_{-j} \,|\, X\big) \propto \exp\left\{-\frac{\mathrm{BIC}(G_j, G_{-j})}{2}\right\} = S\big(G_j, G_{-j}\big), \tag{A5}$$

where $\mathrm{BIC}(G_j, G_{-j})$ is computed as the sum of the piecewise local BIC scores associated with the factorization of G according to G_j and G_{-j} and where each local score corresponds to the BIC score of the regression of each node on its parents.

The full conditional distributions for this sampler are then given by

$$P\Big(G_j^{(i)} = g_j \Big| G_{-j}^{(i)} = g_{-j}^{(i)}, X\Big) = \frac{S\Big(g_j, g_{-j}^{(i)}\Big)}{\sum_{g_j \in K_j^*} S\Big(g_j, g_{-j}^{(i)}\Big)}, \tag{A6}$$

where $g_{-j}^{(i)}$ corresponds to the configuration of the sampled parent sets of the nodes other than j at iteration i of the algorithm. For instance, for node V_4, $g_{-4}^{(i)} = (g_1^{(i)}, g_2^{(i)}, g_3^{(i)}, g_5^{(i)}, g_6^{(i)})$.

The two-parent sets block Gibbs sampler—2PB (Goudie and Mukherjee 2011)

Goudie and Mukherjee (2011) pointed out that the 1PB sampler described above might still suffer from slow convergence when the parent sets are highly correlated. To circumvent this problem, the authors proposed the 2PB sampler, where pairs of parent sets are blocked together. In their original implementation, both the parent set pairs and the updating order are randomly selected at each iteration of the algorithm.

The full conditional distributions for this sampler are given by

$$P\left(G_{j_1}^{(i)} = g_{j_1}, G_{j_2}^{(i)} = g_{j_2} \middle| G_{-(j_1 j_2)}^{(i)} = g_{-(j_1 j_2)}^{(i)}, X\right) = \frac{S\left(g_{j_1}, g_{j_2}, g_{-(j_1 j_2)}^{(i)}\right)}{\sum_{(g_{j_1}, g_{j_2}) \in K_{j_1 j_2}^*} S\left(g_{j_1}, g_{j_2}, g_{-(j_1 j_2)}^{(i)}\right)}, \tag{A7}$$

where $g_{-(j_1 j_2)}^{(i)}$ corresponds to the configuration of the sampled parent sets of the nodes other than j_1 and j_2 at iteration i of the algorithm, and $K_{j_1 j_2}^*$ represents the set of parent set pair configurations, (G_{j_1}, G_{j_2}), such that $G = (G_{j_1}, G_{j_2}, G_{-\{j_1 j_2\}})$ is acyclic.

Efficient implementation of the 2PB sampler depends on the fast computation of $K_{j_1 j_2}^*$, and, once again, we make use of the path count matrix. Let $K_{j_1} = \{V_i : C_{j_1 i}^G = 0\}$ and $K_{j_2} = \{V_i : C_{j_2 i}^G = 0\}$ represent the set of nondescendants of nodes V_{j_1} and V_{j_2}, respectively. Similarly, define the respective complement sets as $K_{j_1}^c = \{V_i : C_{j_1 i}^G > 0\}$ and $K_{j_2}^c = \{V_i : C_{j_2 i}^G > 0\}$ As before, let $K_{j_1}^* = \mathcal{P}(K_{j_1})$ and $K_{j_2}^* = \mathcal{P}(K_{j_2})$ represent the sets of parent sets that can be added separately to V_{j_1} and V_{j_2} without creating a cycle. As pointed out by Goudie and Mukherjee (2011), $K_{j_1, j_2}^* \neq K_{j_1}^* \times K_{j_2}^*$, since the Cartesian product of $K_{j_1}^*$ and $K_{j_2}^*$ might contain networks where a descendant of V_{j_1} is added as a parent of V_{j_2} and a descendant of V_{j_2} is added as parent of V_{j_1}, leading to cycles that do not exist when we consider $K_{j_1}^*$ and $K_{j_2}^*$ separately.

The solution proposed by Goudie and Mukherjee (2011) was to consider the following partition of $K_{j_1 j_2}^*$: (i) parent set pairs that lead to DAGs with no path connecting V_{j_1} and V_{j_2}; (ii) parent set pairs where a descendant of V_{j_1} is a parent of V_{j_2}, but no descendant of V_{j_2} is a parent of V_{j_1}; and (iii) parent set pairs where a descendant of V_{j_2} is a parent of V_{j_1}, but no descendant of V_{j_1} is a parent of V_{j_2}. This partition can be represented graphically by three DAGs: (i) $V_{j_1}\ V_{j_2}$, (ii) $V_{j_1} \Rightarrow V_{j_2}$, and (iii) $V_{j_1} \Leftarrow V_{j_2}$, where the double arrows represent not directed edges between vertices but the existence of a path connecting the vertices. An enumeration of parent sets for each of the three cases described above is given by

path	**parent sets for V_{j_1}**		**parent sets for V_{j_2}**
$V_{j_1}\quad V_{j_2}$,	$\Omega \backslash H_\Omega\left(K_{j_1}^c \cup K_{j_2}^c\right)$	$\times$	$\Omega \backslash H_\Omega\left(K_{j_1}^c \cup K_{j_2}^c\right)$,
$V_{j_1} \Rightarrow V_{j_2}$,	$\Omega \backslash H_\Omega\left(K_{j_1}^c \cup K_{j_2}^c\right)$	$\times$	$H_{K_{j_2}^*}\left(K_{j_1}^c\right)$,
$V_{j_1} \Rightarrow V_{j_2}$,	$H_{K_{j_1}^*}\left(K_{j_2}^c\right)$	$\times$	$\Omega \backslash H_\Omega\left(K_{j_1}^c \cup K_{j_2}^c\right)$,

(A8)

where $\Omega = \mathcal{P}(\{1, 2, ..., p\})$ is the power set of all node indexes, and $H_A(B)$ is a set function defined (for any set B, and any set of sets A) as

$$H_A(B) = \{a \in A : a \supset b \in B\}, \tag{A9}$$

where (in words) we select the sets on A that contain an element of the set B. For instance, for $A = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$, $B_1 = \{2\}$, and $B_2 = \{1, 3\}$, we have that $H_A(B_1) = \{\{2\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}$ and $H_A(B_1) = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}, \{3\}, \{2, 3\}\}$

Note that for case i, we have that the set of parent sets of both V_{j_1} and V_{j_2} is given by the remaining parent sets of Ω, after we removed all sets containing V_{j_1}, V_{j_2}, or their descendants. [In other words, we restrict our attention to parent sets composed of nondescendants of both V_{j_1} and V_{j_2}. Note that an alternative expression for the set $\Omega \backslash H_\Omega(K_{j_1}^c \cup K_{j_2}^c)$ is $K_{j_1}^* \cap K_{j_2}^*$.] For case ii, we have that the parent sets of node V_{j_1} are still given by $\Omega \backslash H_\Omega(K_{j_1}^c \cup K_{j_2}^c)$, since we do not allow node V_{j_2}, or any of its descendants, to be a parent of V_{j_1} or a descendant of V_{j_1} to be one of its parents. The set of parents of node V_{j_2}, on the other hand, is given by $H_{K_{j_2}^*}(K_{j_1}^c)$, since the existence of a path connecting V_{j_1} to V_{j_2} implies that V_{j_1}, or one of its descendants at least, must be a parent of V_{j_2}. The rationale of case iii is analogous to that of case ii and follows by replacing V_{j_1} by V_{j_2}. For any pair of parent sets (G_{j_1}, G_{j_2}), $K_{j_1 j_2}^*$ is given by the union of the three Cartesian products in (A8).

Higher-order parent sets block Gibbs samplers—3PB and 4PB

Here we extend the 2PB sampler to blocks formed by an arbitrary number u of parent sets. Clearly, the full conditional distributions for the Gibbs sampler are given by

$$P\left(G_{j_1}^{(i)} = g_{j_1}, G_{j_2}^{(i)} = g_{j_2}, \ldots, G_{j_u}^{(i)} = g_{j_u} \middle| G_{-(j_1, j_2, \ldots, j_u)}^{(i)} = g_{-(j_1 j_2 \ldots j_u)}^{(i)}, X\right) = \frac{S\left(g_{j_1}, g_{j_2}, \ldots, g_{j_u}, g_{-(j_1, j_2, \ldots, j_u)}^{(i)}\right)}{\sum_{(g_{j_1}, g_{j_2}, \ldots, g_{j_u}) \in K_{j_1 j_2 \ldots j_u}^*} S\left(g_{j_1}, g_{j_2}, \ldots, g_{j_u}, g_{-(j_1, j_2, \ldots, j_u)}^{(i)}\right)}, \tag{A10}$$

where $K^*_{j_1,j_2,\ldots,j_u}$ represents the set of parent set configurations, $(G_{j_1}, G_{j_2}, \ldots, G_{j_u})$, such that $G = (G_{j_1}, G_{j_2}, \ldots, G_{j_u}, G_{-\{j_1,j_2,\ldots,j_u\}})$ is acyclic.

Similarly to the 2PB sampler, the computation of $K^*_{j_1,j_2,\ldots,j_u}$ is facilitated by considering a partition of $K^*_{j_1,j_2,\ldots,j_u}$ into d separate sets, where d corresponds to the number of distinct DAGs composed of u nodes. As before, a double arrow pointing from V_{j_i} to V_{j_k} represents the existence of a path connecting V_{j_i}, or one of its descendants, to V_{j_k}. For instance, for $u = 3$, we have that the partition of $K^*_{j_1 j_2 j_3}$ in the 3PB sampler is done across 25 DAGs

$$\begin{array}{ccccccccc} (1) & V_{j_2} & & (2) & V_{j_2} & & (3) & V_{j_2} & \ldots & (25) & V_{j_2} \\ & \swarrow & & & \swarrow & & & \nearrow & & & \swarrow \quad \nwarrow \\ V_{j_1} & & V_{j_3} & V_{j_1} & & V_{j_3} & V_{j_1} & \Longrightarrow V_{j_3} & \ldots & V_{j_1} & \Longleftarrow V_{j_3} \end{array} \tag{A11}$$

and enumeration of the respective parent sets is given by

$$\begin{array}{lccccc} \textbf{path} & \textbf{parent sets for } \boldsymbol{V_{j_1}} & & \textbf{parent sets for } \boldsymbol{V_{j_2}} & & \textbf{parent sets for } \boldsymbol{V_{j_3}} \\ (1) & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right) & \times & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right) & \times & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right), \\ (2) & H_{K^*_{j_1}}\left(K^c_{j_2}\right) & \times & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right) & \times & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right), \\ (3) & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right) & & H_{K^*_{j_2}}\left(K^c_{j_1}\right) & & H_{K^*_{j_3}}\left(K^c_{j_2}\right), \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (25) & H_{K^*_{j_2}}\left(K^c_{j_2} \cap K^c_{j_3}\right) & \times & H_{K^*_{j_2}}\left(K^c_{j_3}\right) & \times & \Omega\backslash H_\Omega\left(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3}\right). \end{array} \tag{A12}$$

Note that for any node that does not have an arrowhead pointing in, the set of parent sets is given by $\Omega\backslash H_\Omega(K^c_{j_1} \cup K^c_{j_2} \cup K^c_{j_3})$, since the set of parent sets of V_{j_1}, V_{j_2}, and V_{j_3} is given by the remaining parent sets of Ω, after we removed all sets containing V_{j_1}, V_{j_2}, V_{j_3}, or any of their descendants. For nodes that have one arrowhead pointing to them, the set of parent sets is given by $H_{K^*_{j_r}}(K^c_{j_k})$, when $V_{j_k} \Rightarrow V_{j_r}$, since the existence of a path connecting V_{j_k} to V_{j_r} implies that V_{j_k}, or one of its descendants, must be a parent of V_{j_r}. For nodes that have two arrowheads pointing in, we have that the set of parent sets is given by $H_{K^*_{j_r}}(K^c_{j_k} \cap K^c_{j_s})$, when $V_{j_k} \Rightarrow V_{j_r} \Leftarrow V_{j_s}$,, since in this case V_{j_k} (or one of its descendants) and V_{j_s} (or one of its descendants) must both be parents of V_{j_r}.

In general, for the order u parent sets block Gibbs, the set of parent sets of any node that does not have any arrowhead pointing in is given by

$$\Omega\backslash H_\Omega\left(\cup_{k=1}^{u} K^c_{j_k}\right), \tag{A13}$$

whereas the parent sets of any node V_{j_r} that has one or more arrowheads pointing in are given by

$$H_{K^*_{j_r}}\left(\cup_{k \in \mathcal{K}} K^c_{j_2}\right), \tag{A14}$$

where $\mathcal{K}$ represents the set of nodes in the tails of the arrows pointing to V_{j_r}.

Parent sets block Gibbs samplers with biased update—c2PB, c3PB, and c4PB

The connected 2PB sampler (c2PB) is a simple modification of the standard 2PB algorithm. Instead of selecting the pair of parent sets G_{j_1} and G_{j_2} at random and independently, the c2PB sampler first selects an edge $(j_1, \ j_2) \ \in E$ at random and then it blocks together the parent sets G_{j_1} and G_{j_2}. Similarly, the connected 3PB (c3PB) updates parent sets G_{j_1}, G_{j_2}, and G_{j_3}, where V_{j_1}, V_{j_2}, and V_{j_3} are connected linearly with one-way causal flow, $V_{j_1} \rightarrow V_{j_2} \rightarrow V_{j_3}$, in the current DAG. The connected 4PB (c4PB) is the extension of c3PB to update the parent sets of four nodes connected as $V_{j_1} \rightarrow V_{j_2} \rightarrow V_{j_3} \rightarrow V_{j_4}$. Contrary to the PB samplers, where the parent sets of all nodes are updated at each Gibbs sampling iteration, the cPB samplers update a single parent set per iteration. Hence, cPB samplers represent heuristic approximations to formal Gibbs samplers (which, nevertheless, achieve state-of-the-art empirical performance, as shown in our simulation study).

GENETICS

Supporting Information
http://www.genetics.org/lookup/suppl/doi:10.1534/genetics.114.172619/-/DC1

Bayesian Network Reconstruction Using Systems Genetics Data: Comparison of MCMC Methods

Shinya Tasaki, Ben Sauerwine, Bruce Hoff, Hiroyoshi Toyoshiba, Chris Gaiteri, and Elias Chaibub Neto

DOI: 10.1534/genetics.114.172619

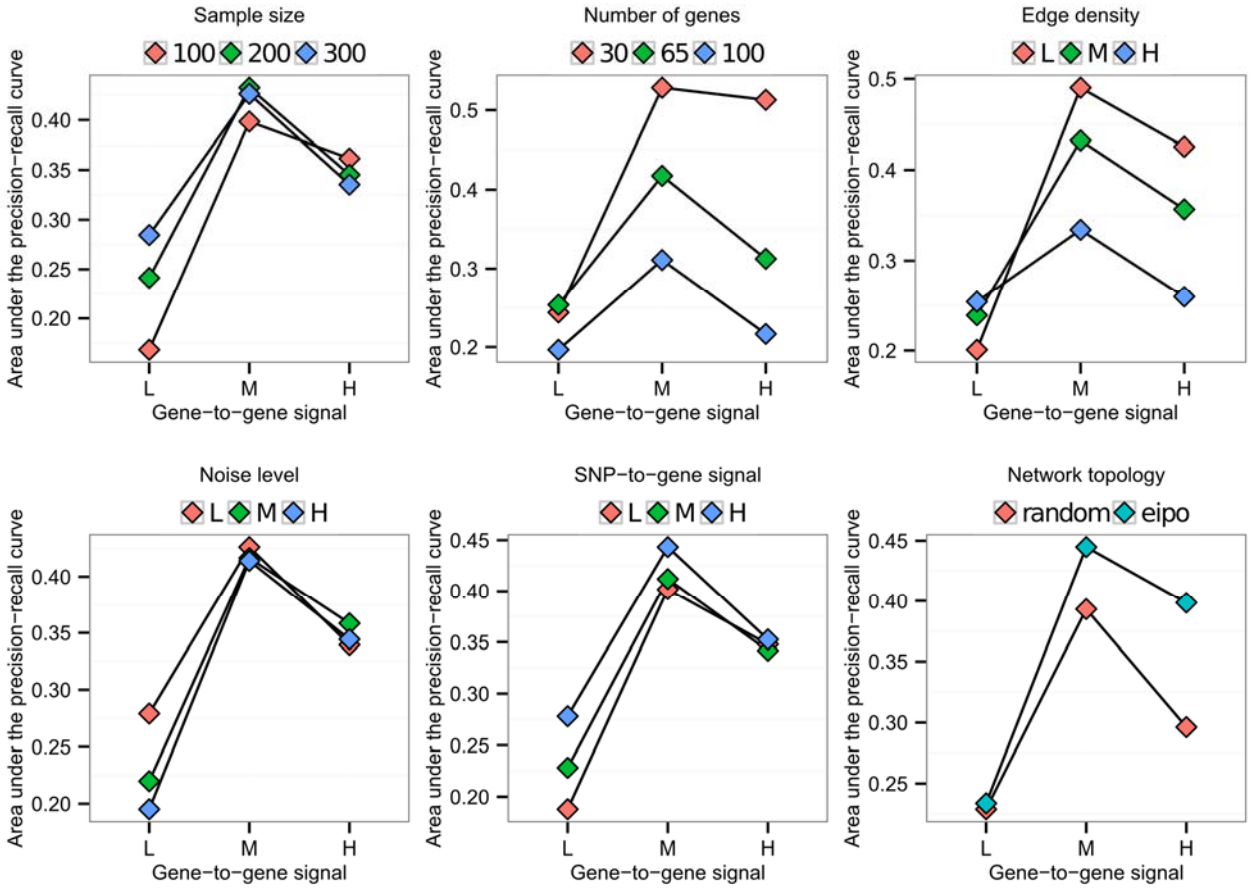

Figure S1 Interaction plots for gene-to-gene signal and six simulation parameters.

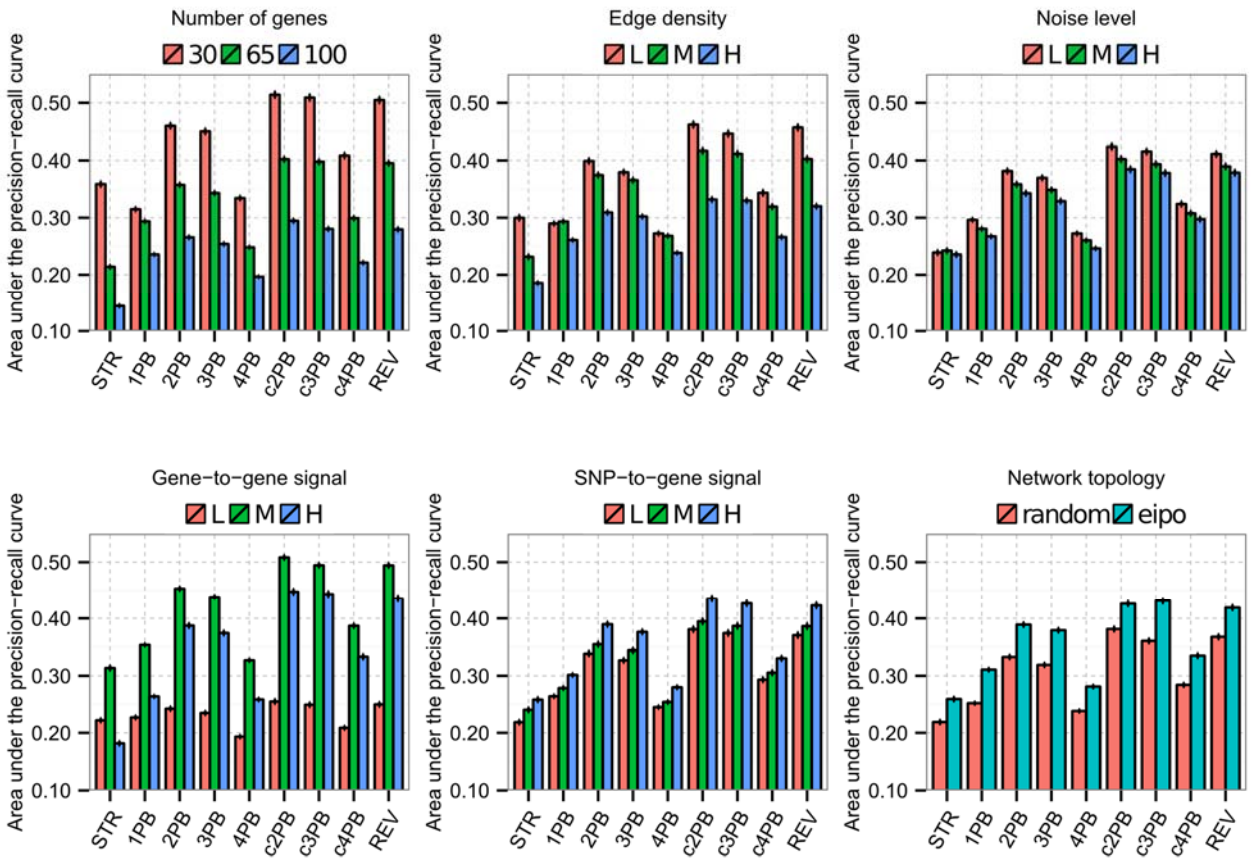

Figure S2 Interaction plots for MCMC samplers and six simulation parameters.

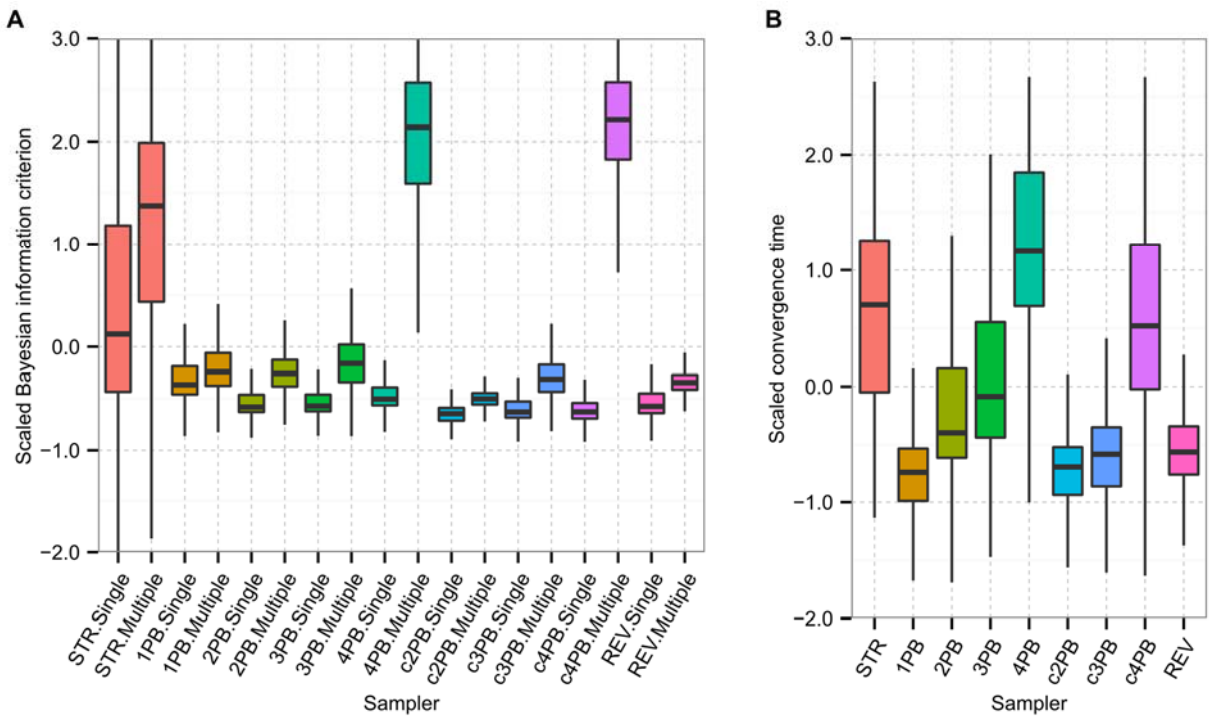

Figure S3 Performance statistics of MCMC samplers for structure learning. (A) Scaled Bayesian information criterion distribution over the MCMC samplers. The average BIC is calculated as mean of BIC of sampled networks in each MCMC run. Then, average BICs of each sampler is scaled over the MCMC runs for the same data set. (B) Scaled convergence time distribution over the nine MCMC samplers. The convergence time is defined as computation time required to reach mean BIC of the last 10% of samples in the MCMC run. The convergence time is scaled over the MCMC runs for the same data set.

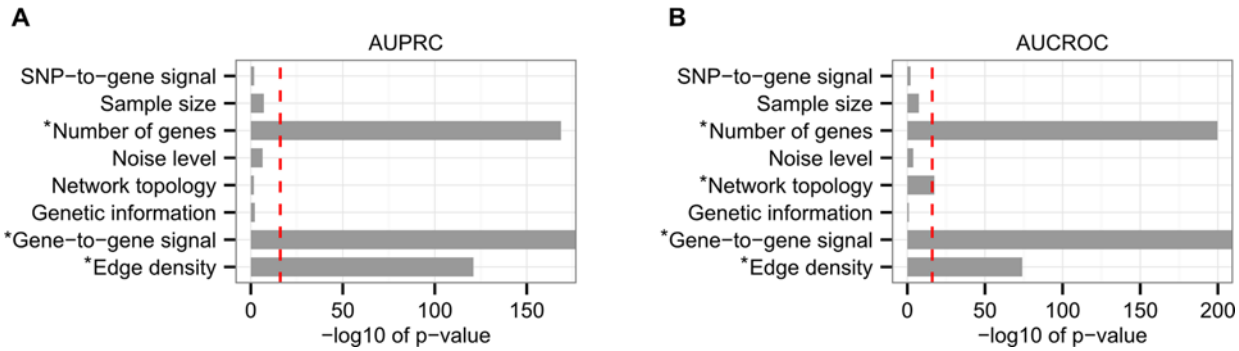

Figure S4 Significance of association between dataset clusters and the eight simulation parameters. Independence between the dataset clusters shown in Figure 5 and the eight simulation parameters were assessed by Chi-square test. Bar plots indicate significance of parameters associated with dataset separation based on (A) the AUCPR result and (B) the AUCROC result. Asterisks indicate the parameters that are significant at p-value < 10E-16.