RESEARCH

Fusing gene expressions and transitive protein-protein interactions for inference of gene regulatory networks

Wenting Liu[1,2] and Jagath C. Rajapakse[3]*

Abstract

Background: Systematic fusion of multiple data sources for Gene Regulatory Networks (GRN) inference remains a key challenge in systems biology. We incorporate information from protein-protein interaction networks (PPIN) into the process of GRN inference from gene expression (GE) data. However, existing PPIN remain sparse and transitive protein interactions can help predict missing protein interactions. We therefore propose a systematic probabilistic framework on fusing GE data and transitive protein interaction data to coherently build GRN.

Results: We use a Gaussian Mixture Model (GMM) to soft-cluster GE data, allowing overlapping cluster memberships. Next, a heuristic method is proposed to extend sparse PPIN by incorporating transitive linkages. We then propose a novel way to score extended protein interactions by combining topological properties of PPIN and correlations of GE. Following this, GE data and extended PPIN are fused using a Gaussian Hidden Markov Model (GHMM) in order to identify gene regulatory pathways and refine interaction scores that are then used to constrain the GRN structure. We employ a Bayesian Gaussian Mixture (BGM) model to refine the GRN derived from GE data by using the structural priors derived from GHMM. Experiments on real yeast regulatory networks demonstrate both the feasibility of the extended PPIN in predicting transitive protein interactions and its effectiveness on improving the coverage and accuracy the proposed method of fusing PPIN and GE to build GRN.

Conclusion: The GE and PPIN fusion model outperforms both the state-of-the-art single data source models (CLR, GENIE3, TIGRESS) as well as existing fusion models under various constraints.

Keywords: Gene regulatory network (GRN), Gene expressions, Gaussian mixture model (GMM), Protein-protein interaction networks, Transitive protein-protein interactions

Background

Gene regulations describe the interactions among genes during cellular activity. Through regulation, genes orchestrate the level of synthesized mRNA and thereby control the expression of other genes and the rates at which proteins are produced, eventually deciding the state of the cell. Gene expression (GE) microarrays provide quantitative or semi-quantitative data on the cell state at a specific time and condition. By "reverse-engineering" GE data, regulatory interactions among genes can be identified and gene regulatory network can be mapped using computational methods [1, 2].

Vast majority of functional analysis approaches to modelling microarray GE data assume that genes with similar expression profiles have similar cellular functions [3–5]. A molecular pathway is a set of genes that activate together to achieve a specific task and thus share similar expression profiles. In this paper, we use a data-driven method - the model-based clustering - to model genes in distinct pathways. Specifically, we model each pathway

*Correspondence: asjagath@ntu.edu.sg
[3]School of Computer Engineering, Nanyang Technological University, Singapore, Singapore
Full list of author information is available at the end of the article

as a Gaussian model as it allows modelling correlations among gene expressions in a data-driven way. It is better suited for situations where the prior knowledge of the regulatory pathways is unknown. In addition, because genes naturally participate in more than one regulatory pathway, soft-clustering is used to allowed so that genes can have memberships in multiple pathways. Hence, we adopt the Gaussian mixture model (GMM) on GE data so that different regulatory pathways can be identified.

The rationale behind clustering is that co-expressed genes, i.e., genes in the same cluster are more likely to be functionally related and belong to the same cluster. However, regulatory processes of the genes in a cluster could not necessarily be direct as it could refer to an indirect regulation via proteins, metabolites, or ncRNAs. In cases where two interaction partners are transcription factors or where two proteins are in the same complex, the interactions are direct. In order to identify indirect regulations in GRN, evidences from multiple data sources should be used. For example, medical literature, protein-protein interaction (PPI) data, gene ontology, etc., have all been used to supplement wet lab data in the inference of GRN [6–9]. When more than one source are available, an essential step is to optimally combine evidences from multiple sources to derive a coherent GRN [10–12].

Since proteins are products of genes, protein interactions provide useful evidence for gene regulation. PPIN data have been fused with GE data for GRN inference in previous studies [13–17]. Most of these works considered only binary links of PPIN: if the link is consistent with the predicted edge from GE, the link is accepted as a true regulation. This approach throws away valuable information, so an accurate quantitative scoring scheme is needed to evaluate consistency between PPIN and gene regulation. On the other hand, existing PPINs are sparse and many real protein interactions are missing in current PPIN databases. Suggested by previous PPI prediction works [18, 19], there exist a large number of interactions between proteins in complexes, which have not yet been observed or recorded in current PPIN. We therefore propose a heuristic to quantitatively extend sparse PPIN by using transitive linkages. We then propose a novel way to score protein interactions by combining topological properties of extended PPIN and correlations of GE. Our experiments demonstrate that transitive protein interactions indeed play an important role in predicting protein interactions. We fuse the extended PPIN scores with GE data, using a Gaussian hidden Markov model (GHMM) to identify gene regulatory pathways, which are found to more consistent with PPIN than those produced by GMM. We further refine PPIN confidence scores by including gene interaction scores from GHMM, which makes the PPIN score more consistent with the existing GRN.

Since there exists no widely accepted model that universally fits GE data well [20–23], and different models capture different GE properties leading to different or complementary GRN structures [21], fusion of different models should lead to better GRN. The GHMM identifies regulatory pathways and obtain possible interacting genes by considering linear correlations between genes but misses conditional dependencies, i.e., non-linear relations, among genes in the same regulatory pathway. The Bayesian network (BN) model is good at capturing these conditional dependences but suffers from poor computational efficiency. We thus propose a systematic probabilistic framework that fuse these two models and derive coherent GRN closer to biological reality. Specifically, our framework takes a coarse-to-fine approach: GHMM generates regulatory pathways (i.e., a coarse GRN having high coverage) and obtains refined interaction scores, both of which are then used to constrain the GRN structure of a BN model (i.e., the Bayesian Gaussian mixture model). This generates GRN that are of good coverage and high precision. Furthermore, GRN structural constrains help greatly reduce the search space for BGM model, thereby reducing the overall computational complexity. Figure 1 shows the flow chart of our GRN inference process.
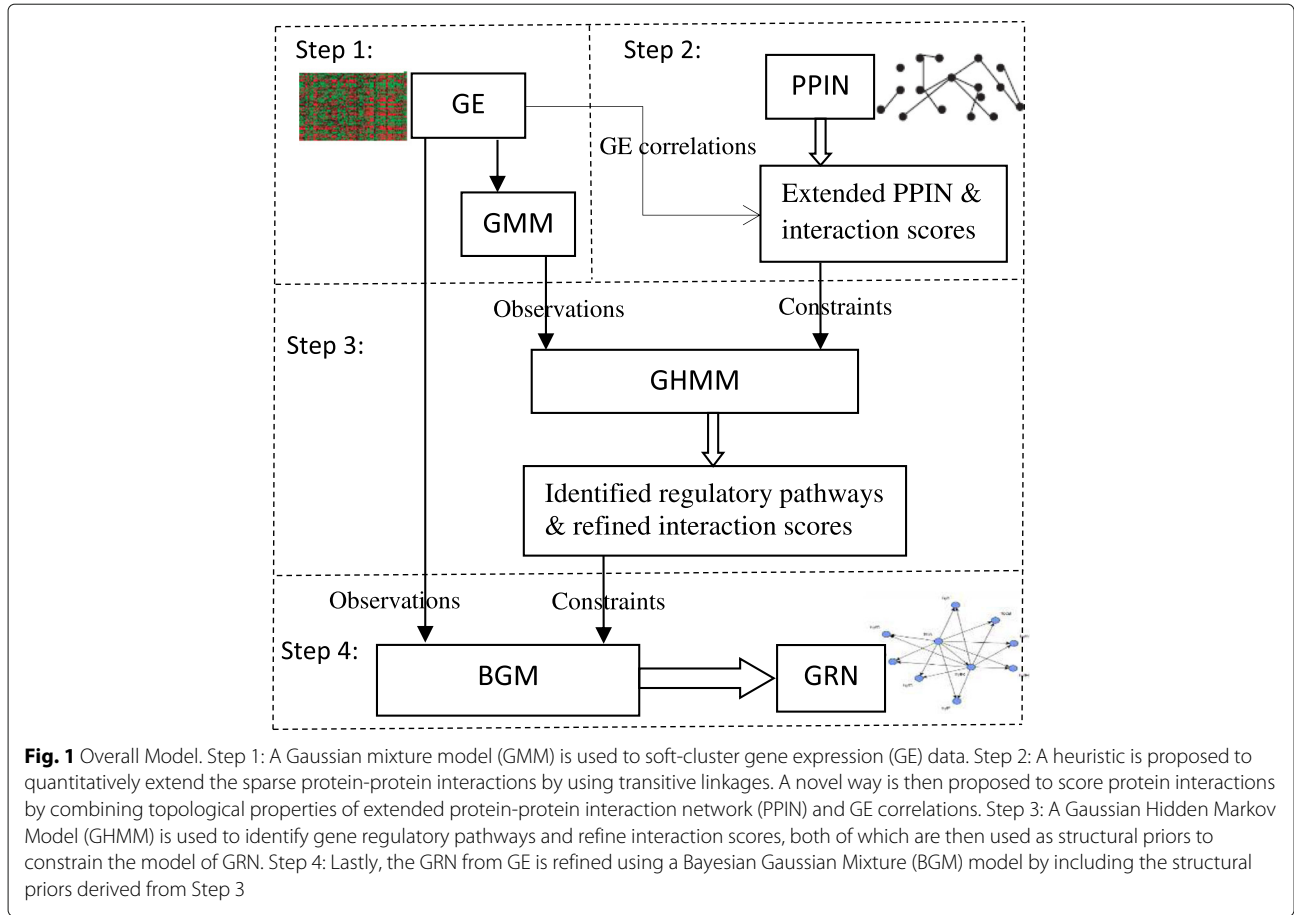
Methods
Soft-Clustering of GE by Gaussian mixture model
Gaussian mixture model
Since gene expression measurement can be viewed as an expression of every gene over all the possible pathways, we employ a Gaussian mixture model (GMM) [24] to describe gene expression (GE) data. For simplicity, we assume that the expression of genes in a pathway follows a Gaussian model, and gene expression data are generated by a finite mixture of underlying probability distribution, that is, by multivariate normal distributions. The key difference between our work and existing GMM hard clustering [13, 24] of GE data lies in our assumption that each gene can participate in multiple pathways. That is, we allow soft-clustering membership for genes so that they can participate in multiple pathways. When one gene participates multiple pathways, it can have soft-clustering membership values for different pathways, representing its contributions of gene expressions in the Gaussian model in different pathways.

Given a gene expression dataset, let $\mathcal{I} = \{i\}_{i=1}^{I}$ denote the set of genes, $\mathcal{L} = \{l\}_{l=1}^{L}$ be the set of pathways, and $\gamma_i \in \mathcal{L}$ denote the regulatory pathways in which gene i participates, i.e., the pathway assignment for gene i. The regulatory pathway assignments of gene set $\mathcal{I}$ is denoted by the set $\Gamma = \{\gamma_i\}_{i=1}^{I}$. The pathway assignment variable γ_i follows a multinomial distribution parameterized by vector $\Theta = \{\theta_l\}_{l=1}^{L}$, with assignment probabilities $p(\gamma_i = l) = \theta_l, \theta_l \in [0, 1]$ where $\sum_{l=1}^{L} \theta_l = 1$. We assume

Fig. 1 Overall Model. Step 1: A Gaussian mixture model (GMM) is used to soft-cluster gene expression (GE) data. Step 2: A heuristic is proposed to quantitatively extend the sparse protein-protein interactions by using transitive linkages. A novel way is then proposed to score protein interactions by combining topological properties of extended protein-protein interaction network (PPIN) and GE correlations. Step 3: A Gaussian Hidden Markov Model (GHMM) is used to identify gene regulatory pathways and refine interaction scores, both of which are then used as structural priors to constrain the model of GRN. Step 4: Lastly, the GRN from GE is refined using a Bayesian Gaussian Mixture (BGM) model by including the structural priors derived from Step 3

that the set of weights $\Theta = \{\theta_l\}_{l=1}^L$ follow a Dirichlet distribution $p(\Theta) = Dir(\alpha_1, \ldots, \alpha_L)$ with Dirichlet weights having uniform priors.

Given gene expression data $\mathcal{X} = \{x_i\}_{i \in \mathcal{I}}$, each instance i has J continuous-valued attributes $x_i = \{x_{ij}\}_{j=1}^J$, where x_{ij} represents the gene expression value measured for gene i in experiment j out of a total of J experiments. We represent each pathway with a Gaussian model wherein the tuple representing the gene expression levels of each experiment (conditioned on a pathway) is a multivariate sample. The probability of observing a given tuple of gene expression levels conditioned on a pathway is $p(x_i | \gamma_i = l) \sim N(\mu_l, \Sigma_l)$, where μ_l is the mean vector and Σ_l is the $J \times J$ covariance matrix.

The pathway assignments are determined by $p(\Gamma | \mathcal{X}) \propto p(\mathcal{X}, \Gamma)$. Assuming independence between genes, the joint distribution over $\mathcal{X}$ and pathway assignments Γ is

$$p(\mathcal{X}, \Gamma) = \prod_{i=1}^I p(x_i, \gamma_i) = \prod_{i=1}^I \sum_{l=1}^L p(x_i, \gamma_i = l) \quad (1)$$

Learning parameters of the Gaussian mixture model

Expectation Maximization (EM) algorithm [25] is used to estimate the parameters of the joint likelihood in (1). The

EM algorithm generates a sequence of parameter approximations that eventually maximizes the observed likelihood. The parameters are initialized to random values. The t-th iteration comprises the following two steps.

E-step: Compute the probability of gene $\{i : i = 1, \ldots, I\}$ conditioned on regulatory pathway $\{l : l = 1, \ldots, L\}$ by using current parameter estimates as:

$$p\left(x_i | \mu_l^t, \Sigma_l^t\right) \sim N\left(\mu_l^t, \Sigma_l^t\right);$$

$$\tau_{il}^t = p(x_i, \gamma_i = l) = \frac{\theta_l^t p\left(x_i | \mu_l^t, \Sigma_l^t\right)}{\sum_{l'=1}^L \theta_{l'}^t p\left(x_i | \mu_{l'}^t, \Sigma_{l'}^t\right)}. \quad (2)$$

This requires $I \times L$ computations.

M-step: Re-estimate the GMM parameters $\{\theta_l, \mu_l, \Sigma_l\}_{l=1}^L$ for all L pathways as:

$$\theta_l^{t+1} = \frac{1}{I} \sum_{i=1}^I \tau_{il}^t;$$

$$\mu_l^{t+1} = \frac{\sum_{i=1}^I \tau_{il}^t x_i}{\sum_{i=1}^I \tau_{il}^t}; \quad (3)$$

$$\Sigma_l^{t+1} = \frac{\sum_{i=1}^I \tau_{il}^t \left(x_i - \mu_l^{t+1}\right) \left(x_i - \mu_l^{t+1}\right)'}{\sum_{i=1}^I \tau_{il}^t}.$$

Once EM procedure converges, we accumulate the joint probability of each pair of genes (i, i') over all the pathways. This gives the pairwise interaction probability of two genes $G_{i,i'}$, which measures how likely both genes belong to the same pathway (assuming that genes are mutually independent):

$$G_{i,i'} = p\left(x_i, x_{i'}, \gamma_i = \gamma_{i'}\right) = \sum_{l=1}^{L} \tau_{il}\tau_{i'l}. \tag{4}$$

where $\tau_{il} = p(x_i, \gamma_i = l)$ be the joint probability of gene i belonging to the l-th regulatory pathway. That is, from genes' pathway assignments matrix $\tau = \{\tau_{il}\}_{I \times L}$, we obtain the first version of GRN as a multiplication of τ and its transpose τ^T:

$$G = \tau \times \tau^T. \tag{5}$$

Note that $G = \{G_{i,i'}\}$ consists of the probabilities of gene pairs belonging to the same regulatory pathways. Each entry $G_{i,i'}$ denotes the interaction probability of gene i and i', estimated from GE data.

The number of mixture components L can be determined by the component-wise EM algorithm [25] automatically, instead of trying all possible $L \in [L_{min}, L_{max}]$ via the time-consuming EM algorithm. The idea is to make use of the non-increasing property of $p(\mathcal{X}, \Gamma | \mathcal{M}_L)$ with respect to L, implying that the minimum message length (MML) criterion $\text{MML}(\mathcal{X}, \Gamma, \mathcal{M}_L)$ also decreases with L. Starting from $L = L_{max}$, we run the EM algorithm and compute the MML. At the next iteration, L is decremented by eliminating the smallest or empty components (setting the smallest θ_l as 0). The E and M steps are repeated to compute an updated MML value at each iteration until $\text{MML}(\mathcal{X}, \Gamma, \mathcal{M}_L)$ converges. In this way, not all values of L need to be evaluated, saving unnecessary computations.

Structural inferences from PPIN
Implicit PPI derived from transitivity
Suggested by [18], there exist a large number of interactions between protein complex components that are not yet observed or recorded. Yamada et al. [19] showed that PPIN evolutionary properties, e.g., shortest path, clustering coefficient, give some clues for potential protein interactions. We therefore propose a heuristic to augment protein-interaction networks (PPIN) by assuming transitivity among known protein interactions. The rationale behind this heuristic is also the base for a lot of similarity-based approaches of predicting GRNs, e.g., the relevance networks based algorithms [26] assumed that "if A is similar to B, then A interacts with B". Hence, "if A is similar to B, and B is similar to C, then A is similar to C, implying that A interacts with C." As such, we infer implicit interactions based on the first-order transitivity assumptions of

protein interactions, i.e., if A interacts with B and B interacts with C, we infer that A interacts with C. Suthram et al. [27] designed methods to assign confidence scores of predicted potential protein interactions from multiple data sources, i.e., by combining GE data, literature, and PPIN data. Inspired by their work, we assign confidence scores of implicit protein-protein interactions (PPI) based only on GE and PPIN data. We use a confidence scoring scheme similar to that of [27]. The novelty of our score lies in its combination of the shortest path score and the Markov clustering score (a graph property) instead of the clustering coefficient.

We extract the shortest path of length $d_{ii'}$ from protein/gene i to protein/gene i' in protein interaction network P whose gene set is $\mathcal{I}$ by using Dijkstra's shortest path algorithm [28]. Let $\mathcal{E}$ denote the set of shortest paths $d_{ii'} < \infty$. When the interaction dataset P is quite sparse, we choose a larger PPI dataset of the same species, whose gene set is a superset of $\mathcal{I}$. Use of a larger PPI dataset incorporates additional biological hints for the analysis. By choosing the shortest path as a potential PPI, we obtain a higher confidence in deriving inferred implicit interactions. We also avoid the need to consider many paths by introducing all the transitive PPIs. For example, we can avoid the hassle of considering protein interactions in a clique.

Scoring the shortest path
Implicit interactions are designed to have lower scores compared to explicit interactions; scores are assigned inversely proportional to the number of intermediaries, i.e., implicit interactions with a larger number of intermediaries have proportionately lower scores. The path length $d_{ii'}$ indicates the strength of the implicit association between gene i and i'. Let the confidence score $c_d \in [0, 1]$ be a non-increasing number series over path length d, which quantifies the implicit protein interaction between i and i'. Clearly, c_1 should be 1 for 1-hop connections. For $d \geq 2$, we set $c_d = \zeta^{d-1}$ where ζ is the probability of extending the path by one hop. We assume that each additional hop is independent of the previous hop.

Using the equivalence assumption, we deduce the closed form approximation for ζ as follows. Suppose there are paths of length $d = 2, \ldots, \infty$ between proteins i and i', then there should be a direct interaction between protein i and i', i.e., $\sum_{d=2}^{\infty} c_d = 1$. Thus, the solution to this infinite summation $\sum_{d=1}^{\infty} \zeta^d = \frac{\zeta}{1-\zeta} = 1$ is $\zeta = 0.5$. We therefore assign a confidence score of $c_d = 0.5^{d-1}$ to each shortest path of length d. This score decreases exponentially with the number of hops, e.g., $c_2 = 0.5, c_3 = 0.25, c_4 = 0.125$, etc.

For a gene pair (i, i') in $\mathcal{I}$ with a corresponding shortest transitive path length $d_{ii'}$, a confidence score of $c_{d_{ii'}}$ is assigned omitting $\sum_{pl>d_{ii'}}^{\infty} 0.5^{pl-1}$. In fact, this confidence

score is a conservative estimate of the real confidence score because it assumes that there exists path lengths of length $d > pl$. In practice, there may exist only one path between two genes. The shortfall is at most $\sum_{pl=d>2}^{\infty} 0.5^{pl-1} < 0.5$. If there exist many paths of varying lengths (overwhelming evidence) between a genes pair, then the confidence score becomes an underestimate. On the other hand, if there exists only one path, which is also the shortest path, then the confidence score is an accurate depiction of the strength of regulation between two genes.

We use the method of [27] to estimate PPI scores by modeling protein interactions as a function of two random variables: (1) the implicit PPI confidence score based on the shortest path and (2) the Pearson correlation coefficient of expression measurements for the corresponding genes. Unlike the approach of [27], which learns the weight of each random variable from a training set of positive and negative examples, we simply use uniform weights. Our formulation involves only two variables: the implicit PPI confidence score and Pearson correlation coefficient derived from GE data. Let W with elements W_{ij} denote the graph with connection strengths computed from these two variables, then

$$W_{ij} = \rho_{ij} + c_{d_{ij}} = \rho_{ij} + 0.5^{d_{ij}-1}, \quad (6)$$

where ρ_{ij} is the correlation of gene i and j in the gene expression data, and d_{ij} is the shortest path length between gene/protein i and j in the PPIN.

Until now, W collected the evidence from GE correlation and transitive protein-protein interaction for predicting direct gene-gene interactions. In fact, there are other evidences for gene-gene interactions. Similar to [29] that used a random walk model to consider the incompleteness of current gene ontology (GO) or PPIN evidences, we propose a random walk model to allow collecting additional evidence in a random fashion for predicting gene-gene interactions.

Topological connectivity via random walk transitions

The random walk models including Markov clustering algorithm [30] and PageRank [31] have been successfully used to model the link structure of graphs. Likewise, we extract the topological structure of the extended PPIN by using a random walk model instead of modelling with a small-world clustering coefficient [27].

Given an undirected graph W, random walk transition matrix T is defined as

$$T = \begin{cases} c \frac{W_{ij}}{\sum_{j=1}^{I} W_{ij}} + \frac{1-c}{I} & \text{if } W_{ij} \neq 0, \\ \frac{1-c}{I} & \text{otherwise.} \end{cases} \quad (7)$$

where $c \in (0.5, 1)$ is the fusion parameter (a.k.a. the damping factor in PageRank, typically set to 0.85) that determines the probability of the next transition from one of the outgoing links versus the transition from going to any random link.

Enright et al. [30] has shown that transition matrix T converges quadratically to an equilibrium state representing the topological connectivity of the graph. The converged matrix, denoted by $\hat{T} = \lim_{k \to +\infty} T^k$, can be computed by Markov CLustering (MCL) algorithm [30]. In our context, the converged matrix of extended PPIN shows how likely protein pairs are related to one another.

Extended PPIN

Recall that the confidence score of how likely the predicted PPI occurs are inferred from the shortest path evidences from existing database, which can be treated as the confidence from biological knowledge. Here, the topological connectivities of the extended PPIN are inferred from the graph structure or the topological properties by random walk transitivity. We combine both of them to refine the confidence score of how likely all protein pairs will interact with each other in the PPIN as $C = W + W * T$.

Combining W from (6) and the transition matrix from (7), the final confidence PPI scores are arrived as follows.

$$C_{ij} = 0.5 \times W_{ij} + 0.5 \times \sum_{k=1}^{I} W_{ik} \hat{T}_{kj}, \quad (8)$$

where W_{ij} denotes the confidence score derived from the PPIN database and GE correlations and $\hat{T}_{kj}$ represents the converged random walk probability. Hence, $W \times \hat{T}$ yields the confidence on the link structures of the original network. The summation term actually compensates for the confidence score W_{ij} that could have been underestimated in the previous step using the shortest path.

To summarize, score C_{ij} starts from the values estimated from PPIN and GE correlations and then updated with converged values of a random walk model. Thus, the confidence score takes into account topological properties of GRN. The extended PPIN is derived from thresholding the C scores.

Gaussian hidden Markov model

When genes collaborate to achieve a specific task, the corresponding protein products generally interact [13]. The PPI serve as valuable hints to underlying regulations among genes in GRN pathways. To fuse PPI and GE data, we treat gene expressions as observations, pathways as hidden states, and protein interactions as transitions in a Gaussian Hidden Markov Model (GHMM) [24]. In the GHMM, gene expression is the observed variable, the pathway it belongs to is the hidden state, given a hidden state (pathway), the expression of genes in the pathway follows a Gaussian model. The are gene-gene interactions are

treated as transitions between their corresponding hidden states (pathways): if they are in the same pathway, the gene-pathway assignments are reinforced; otherwise, they are penalized. In this way, the GHMM combines the extended PPIN in the gene-pathway assignments.

Hidden Markov random field (HMRF) models assume that the conditional distribution of a variable obeys Markov property, i.e., the probability of a variable only depends on the neighbouring variables (see [32] for a complete description of HMRF). In the present context, HMRF graph is represented by a set of nodes where node i represents observation x_i with hidden variable γ_i and the neighborhood graph is represented by constraints C_{ij}, each of which indicates the edge weight between node i and node j. Specifically, the extended PPIN with confidence score C is used as constraints (prior knowledge) to the GHMM, i.e., they are considered neighbourhood structural priors of the corresponding gene-gene associations. Figure 2 shows a sample GHMM model with PPI priors.

The prior probability of a particular cluster assignment Γ follows a Gibbs distribution [33] as $p(\Gamma|C) = \frac{1}{Z}\exp\left(\sum_{i=1}^{I}\sum_{j\neq i}^{I} -C_{ij}\delta(\gamma_j \neq \gamma_i)\right)$ where $\delta(\cdot)$ is the indicator function and $Z = \sum_{\Gamma} p(\Gamma|C)$ is the normalizing function, and C_{ij} is the PPIN confidence score between gene i and j. Exact inference of the posterior requires the complete evaluation of $p(\Gamma|\mathcal{X}, C) \propto p(\mathcal{X}, \Gamma)p(\Gamma|C)$, where $p(\mathcal{X}, \Gamma)$ follows the Gaussian mixture model defined in (1). The posterior [34] probabilities for the multivariate Gaussian case are approximated as

$$
\begin{aligned}
\tau_{il} &= p(x_i, \gamma_i = l|C) \\
&= \frac{\theta_l p(x_i|\mu_l, \Sigma_l)}{\sum_{l'}^{L} \theta_{l'} p(x_i|\mu_{l'}, \Sigma_{l'})} \exp\left(\sum_{j\neq i} -C_{ij}(1-\tau_{jl})\right),
\end{aligned} \quad (9)
$$

where θ_l is the mixture weight and $p(x_i|\mu_l, \Sigma_l)$ is the probability density function of the component model.
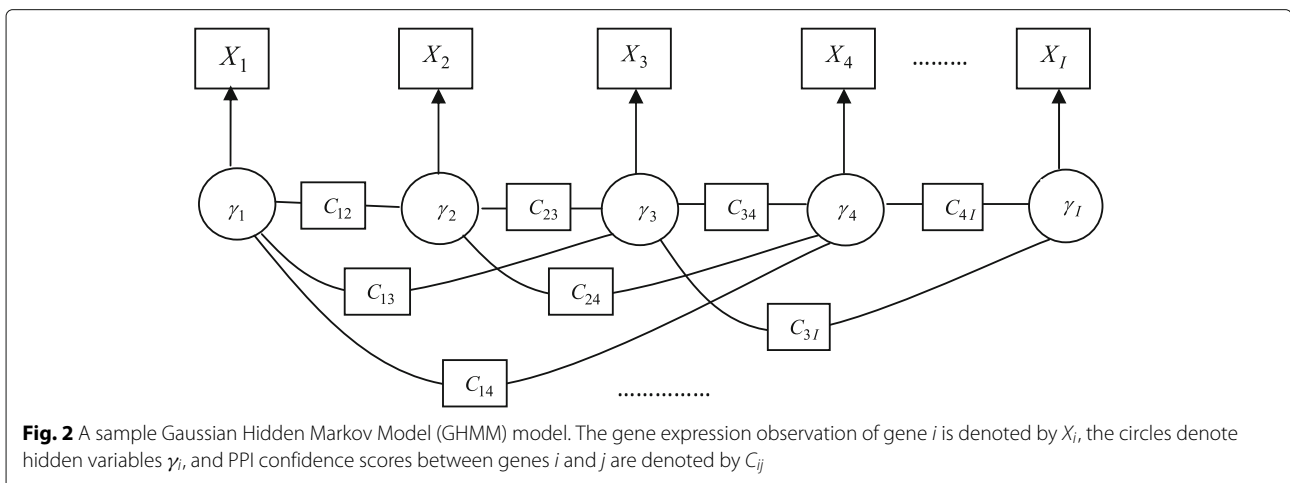
Based on the above, we learn a GHMM by using component-wise EM algorithm of "Soft-Clustering of GE by Gaussian mixture model" section, from which we derive the GRN from the pathway assignments: $G = \tau \times \tau^T$. The gene interaction score is then refined as $R = 0.5 \times G + 0.5 \times C$ where C is the confidence score of PPIN in (8).

Bayesian Gaussian network model

The GHMM assumes that the genes in a pathway are mutually and conditionally independent and do not consider partial correlation of their mRNA expression levels. To model this inter-gene dependency in a pathway, we use a Bayesian network (BN), a directed acyclic graph with local conditional distributions. Here, nodes represent genes, proteins, and/or metabolites, while edges represent molecular interactions such as protein-DNA and protein-protein interactions, including indirect relationships like those from inferred PPI.

BN are prone to overfitting of noisy or sparse training data such as gene expression data. Overfitting could lead to a vastly incorrect graph structure of GRN. Furthermore, since protein levels are unobservable from microarray data, vast majority of BN models of GRN proposed so far only include mRNA levels of genes as nodes but do not include protein levels. Thus, one way to reduce overfitting the BN is to incorporate the prior knowledge extracted from PPI. We thus propose a novel scoring scheme to feed implicit protein interactions (derived from GHMM) as structural priors into BN, thereby enhancing the robustness of predicted GRN.

Another major limitation of BN is its exponentially growing solution space with respect to the size of the network and thus approximate solutions such as those using Monte Carlo Markov Chain (MCMC) or genetic

Fig. 2 A sample Gaussian Hidden Markov Model (GHMM) model. The gene expression observation of gene i is denoted by X_i, the circles denote hidden variables γ_i, and PPI confidence scores between genes i and j are denoted by C_{ij}

algorithms (GA) have been used. Such solutions are prone to errors and suboptimal. Since gene regulations exist among the genes in the same regulatory pathway, we constrain the parent set of each gene in the BN by first detecting the pathways, thereby greatly reducing the solution search space for subsequent computations. In a GRN, every gene appears in multiple pathways and therefore the relationship between genes in each pathway cannot be solved independently, i.e., by modelling each pathway as an independent BN. We thus use a Bayesian Gaussian Mixture (BGM) [35] that simply implements a BN with Gaussian Mixture observations. Here, observations for the Gaussian mixture reflect experimental observations, i.e., the experiments/observations are hard-clustered by using Gaussian Mixture Models. We thus use a MCMC inference method to learn the BGM model with structural priors derived from R scores in "Gaussian hidden Markov model" section, as shown in Fig. 1.

The BN is defined by a graph $\mathcal{G}$ with a family of conditional probability distributions and their parameters $\mathcal{Q} = \{q_i\}$, which together specify the joint distribution over the variables $p(\mathcal{X}|\mathcal{G}, \mathcal{Q})$. The joint distribution in a static BNs is factorized as $p(\mathcal{X}|\mathcal{G}, \mathcal{Q}) = \prod_{i=1}^{I} p(x_i|\pi_i, q_i)$ where each node x_i depends only on its parent nodes π_i. The parameter matrix $\mathcal{Q}$ is comprised of I vectors where each vector q_i specifies a local probability distribution.

If we assume a Gaussian Bayesian Network (GBN), the parameter vector $q_i = \{\mu_{\pi_i}, \sigma_{\pi_i}\}$ consists of the mean and standard deviation of the local probability distributions $p(x_i|\pi_i, q_i) \sim N(\mu_{\pi_i}, \sigma_{\pi_i})$. If we assume conditional independence among the J experiments, then $p(x_i|\pi_i, q_i) = \prod_{j=1}^{J} p(x_{ij}|\pi_i = x_{\pi_i,j}, q_i)$ where $p(x_{ij}|\pi_i = x_{\pi_i,j}, q_i) \sim N(\mu_{\pi_i,j}, \sigma_{\pi_i,j}^2)$. Assuming parameter independence, prior distribution $p(\mathcal{Q}|\mathcal{G})$ of the unknown parameters is expressed in terms of I local prior distributions: $p(\mathcal{Q}|\mathcal{G}) = \prod_{i=1}^{I} p(q_i|\mathcal{G}) = \prod_{i=1}^{I} p(q_i|\pi_i)$. The marginal likelihood $p(\mathcal{X}|\mathcal{G})$ is thus the integral over the parameter space:

$$
\begin{aligned}
p(\mathcal{X}|\mathcal{G}) &= \int p(\mathcal{X}, \mathcal{Q}|\mathcal{G}) d\mathcal{Q} \\
&= \int \left(\prod_{i=1}^{I} p(x_i|\pi_i, q_i) p(q_i|\pi_i) \right) d\mathcal{Q},
\end{aligned}
\tag{10}
$$

which can be rewritten as $p(\mathcal{X}|\mathcal{G}) = \prod_{i=1}^{I} \Psi(x_i, \pi_i)$, where $\Psi(x_i, \pi_i) = \int p(q_i|\pi_i) p(x_i|\pi_i, q_i) dq_i$.

GE data are generated under a variety of conditions that may include time-series experiments, i.e., each experiment may be a continuation of preceding experiments. Thus, it is infeasible to assume independence among individual experiments. Taking a step back, we assume the experiments are generated from Gaussian mixtures and adopt a Bayesian Gaussian mixture model to represent GE data.

Suppose there are K mixtures $\{D^{(k)}\}$ where each mixture $D^{(k)} = \{x_{\cdot,j}\}_{I*m}$ represents some m attributes of a gene set $\mathcal{I}$, and $\{\theta_k\}$ mixture weights of experiments in the BGM, then we express

$$
\begin{aligned}
p(\mathcal{X}|\mathcal{G}) &= \sum_{k=1}^{K} \theta_k p\left(D^{(k)}|\mathcal{G}\right) \\
&= \prod_{i=1}^{I} \sum_{k=1}^{K} \theta_k \Psi\left(D^{(k)}, x_i, \pi_i\right),
\end{aligned}
\tag{11}
$$

where mixture weights $\{\theta_k\}$ are estimated with the EM algorithm, assuming independence among genes. Theoretical considerations and more details specifying the GBN parameters estimation can be found in [36].

Constraining the set of parent candidates

The GHMM derived in "Gaussian hidden Markov model" section can provide some structural constraints to BN learning. First, parents of genes should hail from the same regulatory pathway, i.e., if gene i belongs to some regulatory pathways, then other genes in these regulatory pathways could be its potential parents. In this manner, we generate a more accurate parent candidates set for each gene in addition to using the relationship defined in GRN. This is a nice middle ground between an exhaustive search through all the genes for potential parents and the limited set of GRN derived parents.

With the refined confidence score R assigned to every gene pair, the parent candidate set is limited to only top-k neighbours or top-N gene pairs as determined experimentally. One simple way to determine k and N is to use the average number extracted from a real GRN having a comparable size.

Incorporating neighborhood confidence scores

From [33], the prior probability of a particular structure $\mathcal{G}$ with constraints $R = \{R_{ij}\}$ follows a Gibbs distribution,

$$
p(\mathcal{G}|R) = \frac{1}{Z} \exp \left(\sum_{i=1}^{I} \sum_{j \neq i}^{I} -R_{ij} \left(1 - \mathcal{G}_{ij}\right) \right)
\tag{12}
$$

where $Z = \sum_{\mathcal{G}} p(\mathcal{G}|R)$ is a normalizing function.

By using Markov networks framework to represent the correlations between neighbouring links, joint probability of data $\mathcal{X}$, given the BGM structure $\mathcal{G}$ and constraints R, can be written as $p(\mathcal{X}, \mathcal{G}|R) = p(\mathcal{X}|\mathcal{G}) p(\mathcal{G}|R)$. That is,

$$
p(\mathcal{X}, \mathcal{G}|R) = \frac{1}{Z} \prod_{i=1}^{I} \left[\sum_{k=1}^{K} \theta_k \Psi\left(D^{(k)}, x_i, \pi_i\right) \right] \exp \left(\sum_{j \neq i}^{I} -R_{ji}(1 - \mathcal{G}_{ji}) \right)
\tag{13}
$$

Learning BGM with constraints via MCMC

Given GE data $\mathcal{X}$, fusing association scores R computed from GHMM and fixed parameters $\mathcal{Q}$, the structural posterior probability is written as

$$p(\mathcal{G}|\mathcal{X}, \mathcal{Q}, R) = \frac{p(\mathcal{X}, \mathcal{G}|\mathcal{Q}, R)}{p(\mathcal{X})} \tag{14}$$
$$\propto p(\mathcal{X}, \mathcal{G}|\mathcal{Q}, R)$$

In the context of static BNs, different MCMC methods have been proposed for sampling Directed Acyclic Graphs (DAG) from the structural posterior distribution. We adopt the structural MCMC approach of [35] to sample our BGM structure $\mathcal{G}$ from posterior distributions $p(\mathcal{G}|\mathcal{X}, \mathcal{Q}, R)$. The idea is to give preference to the structures of higher posterior probability. The details of MCMC method to learn DAG from BGM are described in the supplementary materials. With model averaging [37], i.e., we run MCMC structural learning for a maximum number of iterations and each predicted edge in the GRN is assigned a confidence score that is measured by the number of occurrences of predicted edges among the generated graphs.

Experiments and results

Datasets

Our methods were tested on a dataset consists of 25 yeast genes, similar to those used by similar studies on fusion of GE and PPIN [6, 16]. The only difference is that we added five more genes (CDC28, CLB6, CLN3, FUS3, FKH2) that are highly connected to the 25 genes, based on current biological databases and literature. The 30 genes network involves in the cell-cycle regulation of yeast. A cell cycle is comprised of four phases: (i) Gap 1 (G1) phase - the checkpoint to ensure that the cell is ready for division, (ii) Synthesis (S) phase - involving DNA replication, (iii) Gap 2 (G2) phase - a checkpoint to ensure that the cell is ready to enter the next phase, and (iv) Mitotic (M) phase referring to cell division.

We collected data from several resources in order to construct a comprehensive target ground-truth network: (i) GeneNetWaver (http://wwwmgs.bionet.nsc.ru/mgs/gnw/genenet/) [38, 39] that includes 12873 transcriptional regulations among 4441 yeast genes where 62 transcriptional regulations are among our 30 cell cycle genes; (ii) 141 literature-reported regulatory relations among the 30 target genes manually collected from [40, 41], etc.; and (iii) Lee et al. [42] which proposed a integrated functional association score from mRNA expression data, PPIN, and literature mining edges of 5552 yeast genes. The linkage score showed good performance on independent benchmark datasets from KEGG (https://www.genome.jp/kegg/), STRING (https://string-db.org/), Gene Ontology (www.geneontology.org/), and experimentally-determined subcellular localization. We collected the "IntNet" from [42] with a high likelihood of regulation score above 0.5, which included co-expression regulations from 717 experiments for yeast (divided into 27 experimental categories), protein-protein interaction experiments, and literature mining of edges. From this network, 166 linkages are among the 30 target genes. In total, the ground-truth network has 317 regulations among 30 cell cycle genes.

The GE data was obtained from [43] that contains 77 experiments collected over 8 yeast cell cycles by using four different synchronisation protocols. The PPI data for *Saccharomyces Cerevisiae* was downloaded from BioGRID (http://thebiogrid.org/), that contains 6263 proteins and 210,996 interactions.

For comparison, true positives (TP), false positives (FP), and false negatives (FN) of edges were computed by comparing the predicted pathways to the target ground truth network. Various performance metrics including Precision, Recall, and F1-score were evaluated. To compare with state-of-the-art methods of predicting GRN from GE data, such as CLR [26], GENIE3 [44], TIGRESS [45], we determined AUROC (The area under the receiver operating characteristic (ROC) curve) and AUPR (The area under the precision-recall (PR) curve) scores as defined in [44]. For these metrics, the best results that are significantly different (p-value < 0.05) from other methods are shown in bold. We implemented t-test to check whether the best result is significantly different from other methods; If the best result is not significantly different from the next best ones, we choose the best results as one group and the others as another group and then implemented the unpaired t-test for the two groups.

Feasibility of extending PPIN

In "Structural inferences from PPIN" section, we described a method to extend PPIN, using transitive protein interactions and assigned confidence scores C to the predicted protein interactions. We will demonstrate the feasibility of predicting protein interactions from this extended PPIN in this section.

To test how well extended PPIN (derived from C scores) recover an incomplete PPIN, we randomly selected 200 yeast genes and assign 3008 PPIs from BioGRID among them. We used 10-fold cross-validation on the PPINs of these 200 randomly selected genes, by randomly removing 10% edges in the target PPIN, deriving extended PPIN based on the remaining 90% edges, and then comparing the extended PPIN with the target PPIN on the missing 10% edges. We repeated 10-fold cross-validation experiments for 10 times to show the robustness of the method. The results demonstrated that the extended PPIN (when the cut-off threshold for C score is set as 0.5) effectively

recover 346 out of the 383 removed edges on average. Further, our extended PPIN based only on 90% PPIN edges predicted the entire PPIN with an average F1-score of 43%, indicating the effectiveness of the method of extending PPIN (C score) and in its utility in predicting the missing PPIN.

As BioGrid may contain functional linkages predicted as protein-protein interactions, we also choose another PPIN dataset to show the reliability of the method of extending PPIN (C score) to predict real PPIs. The yeast PPIN data from http://interactome.dfci.harvard.edu/S_cerevisiae consists of the full set of physical interactions that occur in a physiologically relevant dynamic range between all its macromolecules, including protein-protein, DNA-protein, and RNA-protein interactions. By combing all the physical interactions, co-complex membership associations [46] and literature-curated interactions, [47] collected in "Yeast Interactome Datasets" from which we get 11995 interactions of 2234 yeast proteins where only 100 PPIs are found among the same randomly selected 200 yeast genes. The experiments of 10-fold cross-validation on these 100 PPIs showed that our extended PPIN (when the cut-off threshold for C score is set as 0.55) effectively recovers the removed edges on average. It also shows that the extended PPIN based only on 90% PPIN edges predict the entire PPIN with an average F1-score of 36.62% (average precision at 39.56% and average recall at 34.10%). This further validates the reliability of the method of extending PPIN (C score) to predict the real PPIs.

In the BGM with prior model, we added the prior information into the likelihood of gene expression data of the learned BN structure. If the GRN learned from GE data is consistent with prior information, the likelihood is reinforced; otherwise, the likelihood is penalized. Thus, the GRN from the fusion model achieves the maximum consistency between gene expression data and prior information. We choose the informative BioGRID PPINs as priors in our fusion model.

In order to show the effectiveness of predicting gene regulations from extended PPIN, we evaluated the extended PPIN alongside the original PPIN (denoted by "raw PPIN"). As mentioned in "Structural inferences from PPIN" section, we can extend PPIN either from a sub-network or from the global network where the global network is the PPIN for the whole genome of the same organism and the sub-network indicates the subnet PPIN defined by the genes in the target gene set. When the PPIN was extended from subnet information, we only consider transitive links among the target gene set. If the subnet PPIN is quite sparse, we choose the global network to make use of transitivity. However, extended PPIN from global information may introduce more noise especially when PPIN is spurious. We experimentally compared differences in the performance of extending PPIN from the sub-network (i.e., 30-gene PPIN for the benchmark) and the global network (i.e., the complete PPIN from 6263 yeast genes) on the 30-gene benchmark network. Table 1 shows the performance with three PPINs: raw PPIN, extended PPIN with subnet information, and extended PPIN with global information on predicting the GRN of the 30 benchmark genes. The cut-off thresholds to generate GRN from C score for the extended PPINs were set as 0.5 for a better trade-off between precision and recall as seen in Fig. 3.
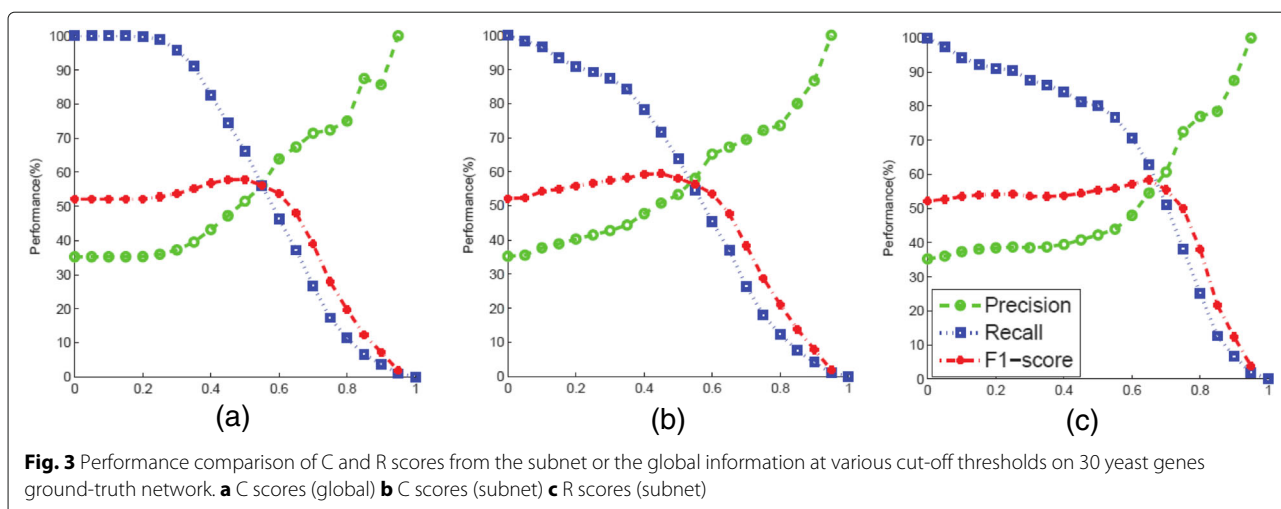
Comparing two extended PPINs, we see that the extended PPIN with global information outperforms the prediction with only subnet information on recall, sacrificing some precision. This indicates that global information may introduce more incoherent genes to make prediction noisy. The observation that both local and global information achieve comparable performance shows that global information can complement subnet information from sparse PPIN data. Clearly, if not much protein interactions are lost in the target PPIN, extended PPIN with subnet information is a good choice to balance the precision and recall, as illustrated by the best F1 performance with extended PPIN from local information. We thus choose extended PPIN with subnet information for the 30-gene benchmark network in subsequent experiments.

Comparing extended PPIN with the raw PPIN, we see that the extended PPIN with subnet information yielded a notable 20% improvement on recall compared to raw PPIN at a hefty decrease of 12% in precision. This shows the feasibility of deriving regulations by using the transitivity present in the PPIN. It is reasonable that raw PPIN predicts GRN with the highest precision, indicating how much protein interactions in the current PPIN database are consistent with gene regulation data. The extended PPINs provide better recall/coverage on

Table 1 Performance of predicting GRN with different PPINs on 30 yeast genes ground-truth network

Method	TP	FP	FN	Prec.(%)	Rec.(%)	F1(%)	AUROC	AUPR
Raw PPIN	165	95	152	63.46	52.05	57.19	0.698	0.591
Extended PPIN (subnet)	229	220	88	51.00	**72.24**	59.79	**0.713**	0.533
Extended PPIN (global)	240	260	77	48.00	**75.71**	58.75	0.696	0.500

Best performance measures that are significantly different are shown in bold

Fig. 3 Performance comparison of C and R scores from the subnet or the global information at various cut-off thresholds on 30 yeast genes ground-truth network. **a** C scores (global) **b** C scores (subnet) **c** R scores (subnet)

predicting GRN even with the subnet information, giving the best F1-score, indicating the incompleteness of the current PPIN database. The precision of extended PPIN can be improved by raising the C score threshold but the recall will then be decreased at the same time. Since our approach is a coarse-to-fine framework to predict GRN where the extended PPIN is used to limit candidate gene interaction set (coarse step), a higher coverage or recall is preferred. As such, we choose 0.5 as the cut-off threshold that gave us a reasonable recall. The high false positive rate from the extended PPIN can be reduced in the subsequent "refinement" step. As subsequent experiments demonstrate, our methods predict the final GRN with a high precision. Therefore, what follows next are demonstrated only with extended PPIN data.

Comparison of performances

We benchmarked the generated GRNs against the ground-truth networks. All the methods except the BGM generated networks with undirected edges. When comparing an undirected graph to a directed one, we considered an existence of a matched link regardless of its direction. For comparison, all the networks generated by the BGM were sampled and averaged for 10 models each of which was learned from 1000 iterations of MCMC structural moves. The performance of the BGMs were further improved by increasing the number of models by model averaging and the iterations of structural moves.

Since the generated graphs from C, R, and $\mathcal{G}$ assigned confidence scores for the edges, i.e., probabilities of edge existence, we chose a cut-off threshold and constructed the GRN, based on the scores: if the edge score is higher than this threshold, we confirm the existence of an edge; and otherwise, the edge was rejected. From experiments, as seen in Fig. 3, we found that the best cut-off thresholds (for C and R scores) that maximize the F1-score (which

is also the best trade-off between precision and recall) of GRN tend to be clustered around 0.5. For GRN structure $\mathcal{G}$ from clustering or the BGM, scores came from the probability of genes belonging to the same pathway or the model averaging where both reflected how likely genes interact. We therefore intuitively fixed 0.5 as the threshold to determine interaction edges. Thus, for consistency, we set the cut-off thresholds for all our graphs to be 0.5. The confidence scores were normalized between 0.0 and 1.0 before averaging. The thresholds can also be experimentally fine-tuned to further improve the performances of our methods. The vast majority of existing works on prediction of GRN choose the cut-off of the confidence score as a trade-off between sensitivity and specificity of prediction [48].

Table 2 compares performances of building GRNs by various methods. The first four methods (CLR, GENIE3, TIGRESS, GMM) are recent state-of-the-art methods of predicting GRN from GE. We use MATLAB implementations of CLR [26], GENIE3 [44], and TIGRESS [45]. The other three methods (GHMM, R scores from GHMM,

Table 2 Performances of prediction of GRNs by various methods on 30 yeast genes ground-truth network

Method	TP	FP	FN	Prec.(%)	Rec.(%)	F1(%)	AUROC	AUPR
CLR	190	312	127	37.85	59.94	46.40	0.555	0.388
GENIE3	128	202	189	38.79	40.38	39.57	0.546	0.395
TIGRESS	140	207	177	40.35	44.16	42.17	0.546	0.392
GMM	172	266	145	39.27	54.26	45.56	0.583	0.412
GHMM	258	329	59	43.95	**81.39**	57.08	0.664	0.461
R scores (GHMM)	250	262	67	**48.83**	78.86	**60.31**	**0.705**	**0.501**
BGM (R scores)	202	237	115	46.01	63.72	53.44	0.627	0.446

Best performance measures that are significantly different are shown in bold

BGM with R scores) are fusion methods that use both the GE and PPIN data as described in the "Methods" section. As seen, all three fusion methods significantly outperformed (5–20% better in all metrics) the methods using GE data only.

To show the effect of extending PPIN by incorporating transitive edges, we also compared three fusion models with and without extending the PPIN (i.e., by using only raw PPIN data). The three fusion methods using extended PPIN outperformed the methods using only raw PPIN. This demonstrates the effectiveness of using extended PPIN in GRN inference.

The GRNs obtained using GHMM and refined R scores are undirected graphs, indicating gene interactions but not specifically the directional regulations. To figure out the directional dependencies among genes, a directional model such as BGM is needed. The BGM model based on R scores as constraints in fact gave the overall best performance and thus deserves further investigation.

BGM with different priors

Using the same set of parent candidates and the BGM model, we evaluated whether the BGM using refined confidence scores R can hold its own when benchmarked against other PPIN fusion methods. Specifically, we compared its performance with two well-known methods. Nariai et al. [14] tested if each added protein pair form a protein complex by considering the complex as a virtual node in the network. The method computes the principal component analysis (PCA) model of the protein pairs from original observations and checks if the protein complex contributes to a higher likelihood. If a higher likelihood is obtained, the complex (pair) is accepted and otherwise rejected. Imoto et al. [15] updated PPIN whenever the learned GRN produced different structure. Each inconsistent edge in the learned DAG is perturbed by either removing or reversing its direction. If a perturbation leads to a better likelihood, the perturbed PPIN is accepted otherwise it is rejected.

Both methods needed significantly more operations than the basic models and increased already substantial computational complexity. Further, they tend to converge to local optima given their greedy hill-climbing nature.

Nariai's method can only increase the protein interactions contributed by a protein complex if the PCA projection of the complex performs better than using raw data. Imoto's method is overly-optimistic as it assumes that the learned DAG is correct. However, the MCMC method is known to only simulate a DAG; it may accept a large number of incorrect edges in one iteration and could accept all of them as updated PPIN in the subsequent iteration. Hence, updating the PPIN based on a learned DAG is unreliable.

To show the utility of each component in the fusion model, we compared the performance of the BGM by using priors from the GMM only, the PPIN only or the GHMM, as shown in Table 3. For the priors from PPIN, we also tested two existing fusion methods with the same priors. We described different components of our method as follows: "BGM (GMM)" denotes the BGM with parental constraints from GMM and predicts GRN only from GE data without PPIN data; "BGM (C scores)" denotes the BGM constrained with C scores from extended PPIN; "Nariai et al. (GHMM)" and "Imoto et al. (GHMM)" use parental constraints from the GHMM with extended PPIN to learn GRNs in BGM with corresponding fusion method; "BGM (R scores)" uses the extended PPIN in the GHMM fusion model and then R constraints for the BGM. For comparison, we also included a BGM without any constraints as the baseline.

The results in Table 3 show that the priors from GHMM significantly improve TPs (highest recall) with comparable F1 scores (higher coverage at the slight expense of precision) compared to the methods using priors only from the PPIN. This indicates that GE clustering help to predict more TPs but at a lower precision. It also demonstrates that the use of the GRN structure from the GMM and extended PPIN in the prediction of GRN. These results further validate the utility of each component in the fusion model. Results in Table 3 show that with same structural constraints, the fusion BGM (with R scores) beats both the BGM proposed by Nariai et al. and by Imoto et al. This means that the R scores are effective measure of the reliability of PPIN edges. Table 3 also shows the running-time of different algorithms implemented with MATLAB on a machine with an Intel Xeon E5-1620 3.6GHz CPU

Table 3 Performance of GRNs generated from the BGM with different priors on 30 yeast genes ground-truth network

Method	TP	FP	FN	Prec.(%)	Rec.(%)	F1(%)	AUROC	AUPR	Time(sec)
BGM	139	201	178	40.88	43.85	42.31	0.553	0.390	636
BGM (GMM)	185	293	132	38.70	58.36	46.54	0.555	0.383	606
BGM (C scores)	170	150	147	**53.13**	53.63	**53.38**	**0.653**	**0.490**	**518**
Nariai et al. (GHMM)	176	273	141	39.20	55.52	45.95	0.570	0.408	1174
Imoto et al. (GHMM)	184	293	133	38.57	58.04	46.35	0.565	0.401	1150
BGM (R scores)	202	237	115	46.01	**63.72**	**53.44**	0.627	0.446	608

Best performance measures that are significantly different are shown in bold

and 8GB RAM. As seen, the priors given by our confidence scores lead to computationally more efficient procedures.

Discussion & conclusion

We proposed an automated method to detect gene regulations by fusing GE data and PPI data. Gene expression data is first soft-clustered into regulatory pathways by a GMM where the number of regulatory pathways is automatically determined via a component-wise EM algorithm. Transitive protein interactions are derived using a novel confidence score and PPIN are refined and extended. The extended PPIN are then fused with the GMM derived from GE data, using a GHMM. This improves the biological relevance of the clustering results and also refines the confidence score of indirect gene regulations/protein interactions. Using refined PPI confidence scores, together with regulatory pathways obtained from the GHMM as structural constraints, a BGM model was used to capture direct gene regulations by using an effective and efficient MCMC procedure. Fusion of the BGM model with GHMM as structural priors generated more accurate GRN compared to those produced by the GHMM and the BGM models. We have experimentally shown that our procedures are more effective than two existing well-known methods for fusing GE and PPIN data.

Furthermore, the fusion framework reduces the overall time complexity of building GRN with Bayesian networks. The GHMM uses a component-wise EM algorithm to soft-cluster genes into pathways at a complexity $O\left(I^2 \times L^2\right)$ (where L denotes the number of components and I denotes the number of genes), indicating that the method scales well on large networks (large I). The BGM structural constraints, i.e., the parent of each node must itself be a node from the same regulatory pathway, help to significantly reduce the search space, i.e., from $O\left(I^I\right)$ to approximately $O\left((I/L)^I\right)$. In fact, for larger I, the reduction can even be more pronounced.

Our approach shows the importance of systematically fusing multiple sources of biological evidences in inferring useful and reliable GRN. In fact, our framework can be extended to fuse more than two data sources, such as gene ontologies, biological pathways (KEGG), transcription factors (known regulators), information mined from literature, and other multi-omics data [4, 5, 10–12]. Our methods may be easily extended by further considering dependencies among the experiments [35] and are worth investigating further.

Abbreviations
AUROC: The area under the receiver operating characteristic (ROC) curve; AUPR: The area under the precision-recall (PR) curve; BGM: Bayesian Gaussian mixture; EM: Expectation maximization; GHMM: Gaussian hidden Markov model; GRN: Gene regulatory networks; HMRF: Hidden Markov random field; MCL: Markov cLustering; MML: Minimum message length; MCMC: Monte Carlo Markov Chain; PPIN: Protein-protein interaction networks

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]School of Public Health and Management, Hubei University of Medicine, Shiyan, Hubei, China. [2]Integrative Biology and Physiology, University of California, Los Angeles, Los Angeles, CA, USA. [3]School of Computer Engineering, Nanyang Technological University, Singapore, Singapore.

Published: 5 April 2019

References
1. Liu F, Zhang S-W, Guo W-F, Wei Z-G, Chen L. Inference of gene regulatory network based on local bayesian networks. PLoS Comput Biol. 2016;12(8):e1005024.
2. Liu Z-P. Reverse engineering of genome-wide gene regulatory networks from gene expression data. Curr genomics. 2015;16:3–22.
3. Langfelder P, Horvath S. Wgcna: an r package for weighted correlation network analysis. BMC Bioinformatics. 2008;9:559.
4. Chan TE, Stumpf MPH, Babtie AC. Gene regulatory network inference from single-cell data using multivariate information measures. Cell Syst. 2017;5:251–67.
5. Aibar S, González-Blas CB, Moerman T, Huynh-Thu VA, Imrichova H, Hulselmans G, Rambow F, Marine J-C, Geurts P, Aerts J, van den Oord J, Atak ZK, Wouters J, Aerts S. Scenic: Single-cell regulatory network inference and clustering. Nat Methods. 2017;14:1083–6.
6. Bernard A, Hartemink AJ. Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. In: Pacific Symposium on Biocomputing. Hawaii: 2005. p. 459–470.
7. Wu M, Chan C. Learning transcriptional regulation on a genome scale: a theoretical analysis based on gene expression data. Brief Bioinform. 2011;13(2):150–61.
8. Wang L, Hou L, Qian M, Deng M. Integrating phosphorylation network with transcriptional network reveals novel functional relationships. PLoS ONE. 2012;7(3):33160.

9. Cheng C, Alexander R, Min R, et al. Understanding transcriptional regulation by integrative analysis of transcription factor binding data. Genome Res. 2012;22(9):1658–67.

10. Zarayeneh N, Ko E, Oh JH, Suh S, Liu C, Gao J, Kim D, Kang M. Integration of multi-omics data for integrative gene regulatory network inference. Int J Data Min Bioinform. 2017;185:223–39.

11. Banf M, Rhee SY. Enhancing gene regulatory network inference through data integration with markov random fields. Sci Rep. 2017;7:1–13.

12. Yugi K, Kubota H, Hatano A, Kuroda S. Trans-omics: How to reconstruct biochemical networks across multiple 'omic' layers. Trends Biotechnol. 2016;34:276–90.

13. Segal E, Wang H, Koller D. Discovering molecular pathways from protein interaction and gene expression data. In: International Conference on Intelligent Systems for Molecular Biology. Brisbane: 2003. p. 264–272.

14. Nariai N, Kim S, Imoto S. Using protein-protein interactions for refining gene networks estimated from microarray data by bayesian networks. In: Pacific Symposium on Biocomputing. Hawaii: 2004. p. 336–347.

15. Imoto S, Higuchi T, Goto T. Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. Stat Methodol. 2006;3(1):1–16.

16. Werhli AV, Husmeier D. Gene regulatory network reconstruction by bayesian integration of prior knowledge and/or different experimental conditions. J Bioinforma Comput Biol. 2008;6(3):543–72.

17. Li M, Wu X, Wang J, Pan Y, Towards the identification of protein complexes and functional modules by integrating ppi network and gene expression data. BMC Bioinformatics. 2012;13:109.

18. Chen Y, Xu D. Global protein function annotation through mining genome-scale data in yeast saccharomyces cerevisiae. Nucleic Acids Res. 2004;32(21):6414–24.

19. Yamada T, Bork P. Evolution of biomolecular networks — lessons from metabolic and protein interactions. Nat Rev Mol Cell Biol. 2009;10(11): 791–803.

20. Emmert-Streib F, Glazko GV, Altay G, Simoes RM. Statistical inference and reverse engineering of gene regulatory networks from observational expression data. Front Genet. 2012;3:8.

21. Bansal M, Belcastro V, Ambesi-Impiombato A. How to infer gene networks from expression profiles. Mol Syst Biol. 2007;3:78.

22. Kuffner R, Petri T, Tavakkolkhah P, Windhager L, Zimmer R. Inferring gene regulatory networks by anova. Bioinformatics. 2012;28(10):1376–82.

23. Zhang X, Zhao X-M, He K, Lu L, Cao Y, Liu J, Hao J-K, Liu Z-P, Chen L. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. Bioinformatics. 2012;28(1):98–104.

24. Vignes M, Forbes F. Gene clustering via integrated markov models combining individual and pairwise features. IEEE Trans Comput Biol Bioinforma. 2009;6:260–70.

25. Figueiredo MAT, Jain AK. Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell. 2002;24(3):381–96.

26. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol. 2010;5(1):8.

27. Suthram S, Shlomi T, Ruppin E. A direct comparison of protein interaction confidence assignment schemes. BMC Bioinformatics. 2006;7:360.

28. Johnson DB. A note on dijkstra's shortest path algorithm. J Assoc Comput Mach (ACM). 1973;20(3):385–8.

29. Yang H, Nepusz T, Paccanaro A. Improving go semantic similarity measures by exploring the ontology beneath the terms and modelling uncertainty. Bioinformatics. 2012;28:1383–9.

30. Enright AJ, Dongen S, Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. 2002;30(7):1575–84.

31. Page L, Brin S, Motwani R. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab. 1999.

32. Li SZ. Markov random field modeling in image analysis, 2nd edn. New York: Springer; 2001.

33. Chandgotia N. Generalisation of the Hammersley-Clifford theorem on bipartite graphs. Trans Amer Math Soc. 2017;369:7107–7137.

34. Lange T, Law MH, Jain AK, Buhmann JM. Learning with constrained and unlabelled data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Beijing; 2005. p. 731–8.

35. Grzegorczyk M, Husmeier D, Edwards KD. Modeling non-stationary gene regulatory processes with a non-homogeneous bayesian network and the allocation sampler. Nucleic Acids Res. 2008;24(18):2071–8.

36. Grzegorczyk M. An introduction to gaussian bayesian networks. Methods Mol Biol. 2010;662:121–47.

37. Battle A, Jonikas M, Walter P. Automated identification of pathways from quantitative genetic interaction data. Mol Syst Biol. 2010;6:379.

38. Ananko EA. Genenet: a database on structure and functional organisation of gene networks. Nucleic Acids Res. 2002;30(1):398–401.

39. Schaffter T, Marbach D, Floreano D. Genenetweaver: In silico benchmark generation and performance profiling of network inference methods. Bioinformatics. 2011;27(16):2263–70.

40. Simon I, Barnett J, Hannett N, Harbison CT, Rinaldi NJ, Volkert TL, Wyrick JJ, Zeitlinger J, Gifford DK, Jaakkola TS, Young Ra. Serial regulation of transcriptional regulators in the yeast cell cycle. Cell. 2001;106(6):697–708.

41. Lee TI, Rinaldi NJ, Robert F, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. Science. 2002;298(5594):799–804.

42. Lee I, Date SV, Adai AT. A probabilistic functional network of yeast genes. Science. 2004;306(5701):1555–8.

43. Spellman PT, Sherlock G, Zhang MQ. Comprehensive identification of cell cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell. 1998;9(12):3273–97.

44. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P. Inferring regulatory networks from expression data using tree-based methods. PLoS ONE. 2010;5:9.

45. Haury A-C, Mordelet F, Vera-Licona P, Vert J-P. Tigress: Trustful inference of gene regulation using stability selection. BMC Syst Biol. 2012;6:145.

46. Collins SR, Kemmeren P, Zhao X-C, Greenblatt JF, Spencer F, Holstege FCP, Weissman JS, Krogan NJ. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. Mol Cell Proteomics. 2007;6(3):439–50.

47. Reguly T, Breitkreutz A, Boucher L, Breitkreutz BJ, Hon GC, Myers CL, Parsons A, Friesen H, Oughtred R, Tong A, Stark C, Ho Y, Botstein D, Andrews B, Boone C, Troyanskya OG, Ideker T, Dolinski K, Batada NN, Tyers M. Comprehensive curation and analysis of global interaction networks in saccharomyces cerevisiae. J Biol. 2006;5(4):11.

48. Djebbari A, Quackenbush J. Seeded bayesian networks: Constructing genetic networks from microarray data. BMC Syst Biol. 2008;2:57.