Zamani Dadaneh and Qian *EURASIP Journal on Bioinformatics and Systems Biology* (2016) 2016:5
DOI 10.1186/s13637-016-0038-9

EURASIP Journal on Bioinformatics and Systems Biology
a SpringerOpen Journal

RESEARCH Open Access

Bayesian module identification from multiple noisy networks

Siamak Zamani Dadaneh and Xiaoning Qian*

Abstract

Background and motivations: Module identification has been studied extensively in order to gain deeper understanding of complex systems, such as social networks as well as biological networks. Modules are often defined as groups of vertices in these networks that are topologically cohesive with similar interaction patterns with the rest of the vertices. Most of the existing module identification algorithms assume that the given networks are faithfully measured without errors. However, in many real-world applications, for example, when analyzing protein-protein interaction networks from high-throughput profiling techniques, there is significant noise with both false positive and missing links between vertices. In this paper, we propose a new model for more robust module identification by taking advantage of multiple observed networks with significant noise so that signals in multiple networks can be strengthened and help improve the solution quality by combining information from various sources.

Methods: We adopt a hierarchical Bayesian model to integrate multiple noisy snapshots that capture the underlying modular structure of the networks under study. By introducing a latent root assignment matrix and its relations to instantaneous module assignments in all the observed networks to capture the underlying modular structure and combine information across multiple networks, an efficient variational Bayes algorithm can be derived to accurately and robustly identify the underlying modules from multiple noisy networks.

Results: Experiments on synthetic and protein-protein interaction data sets show that our proposed model enhances both the accuracy and resolution in detecting cohesive modules, and it is less vulnerable to noise in the observed data. In addition, it shows higher power in predicting missing edges compared to individual-network methods.

Keywords: Module identification, Stochastic block model, Multiple-network clustering, Bayesian clustering, Variational Bayes algorithm

1 Introduction

Identifying modular structures within large-scale networks has attracted significant attention in many research fields, including social science, biology, and information technology, just to name a few. For these applications, the ultimate goal is to group vertices in given networks into cohesive modules or communities, in which the vertices share similar properties, specifically their interaction patterns. Typically, densely connected sub-networks in given networks are considered desirable modular structures [1]. There have been many existing approaches proposed to study this problem in the literature, including spectral clustering algorithms based on graph cut [2, 3], modularity-based algorithms [4, 5], as well as matrix factorization algorithms for network clustering [6, 7].

In addition to these optimization algorithms based on graph theory and mathematical programming, in statistical inference, stochastic block models (SBM) originally proposed by [8] adopt a multinomial-Bernoulli probabilistic model to capture the inherent modular structures in observed networks. Hofman and Wiggins [9] developed a Bayesian framework to find the module or community memberships of vertices in networks under study and took advantage of variational approximation to efficiently sample from the corresponding posterior distributions.

Extending the analysis to dynamic networks has attracted major attention recently. Authors in [10] studied community evolution in blogosphere based on graph characteristics such as in-degrees and out-degrees.

*Correspondence: xqian@ece.tamu.edu
Department of Electrical and Computer Engineering, Texas A&M University, MS 3128, TAMU, College Station, TX, USA

Chi et al. [11] used graph cut size as a measure of community evolution and proposed a dynamic version of spectral clustering. In [12], an algorithm called FacetNet was developed by extending the graph factorization method for analysis of evolutionary networks. A Markov model [13] was adopted to capture temporal community variation in stochastic block models with Gibbs sampling implemented for inference of unknown model parameters.

In this paper, we focus on module identification in biological networks. On one hand, it is often the case that biological networks in public databases [14, 15], by either high-throughput profiling techniques or laborious manual curations, contain significant errors (both false positives and false negatives). On the other hand, usually, several independent such empirical networks are available for studying the species of interest, creating the opportunity to integrate information from different sources and gain higher accuracy and better reproducibility. With these noisy networks, we aim to develop an integrated stochastic model and solution methods to improve the accuracy of module identification by combining information from multiple observed networks. Figure 1 provides the schematic illustration of our basic idea. With multiple noisy observations in the top row of Fig. 1a, the proposed stochastic model assumes that there is a consistent virtual graph that captures the coherent root modular structure. As a specific application of our approach, we can think of multiple networks profiled with different techniques or characterized by multiple types of interactions between vertices. Figure 1b shows the graphical representation of our extended SBM from traditional SBM for analyzing multiple noisy networks. In our model, every observed network $A^{(t)}$ is associated with a latent modular structure $\vec{z}^{(t)}$. These instantaneous structures are considered as the results from stochastic transitions from a latent root modular structure $\vec{z}$ that is coherent in all networks. Note that this is in contrast to previous dynamic models that concentrate on the evolution of modular structures rather than embedding them. The probabilistic inference task is to *simultaneously* learn the root as well as instantaneous modular structures from multiple noisy networks. With such a probabilistic model, we are able to elicit the essential modular structure in all the observed networks. By combining information from these various sources, we can compensate for the perturbation effect from noisy observations. To infer this extended SBM for multiple-network clustering, a variational Bayes method is derived to efficiently quantify uncertainties over unknown model parameters. We apply our method to protein-protein interaction (PPI) data sets and show that by taking advantage of different sources of information, our method outperforms the existing SBM-based methods implemented on individual networks in predicting new protein complexes. Furthermore, the capability of predicting missing edges from our Bayesian modeling creates the opportunity for our method to be used in active learning scenarios, where the task is to efficiently infer

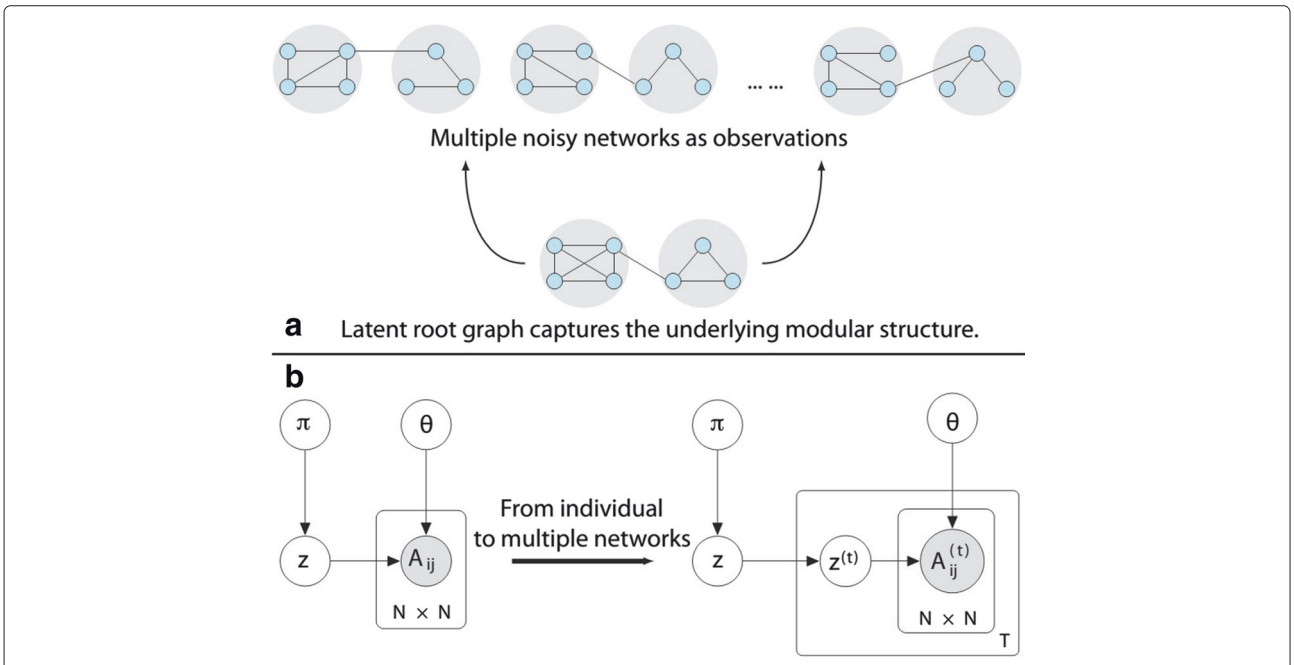

Fig. 1 Schematic illustration. **a** An example of multiple noisy networks with a coherent modular structure. **b** Graphical representation of the proposed probabilistic model for module identification across multiple networks as an extension of the module identification model for individual networks

protein-protein interactions from new sets of experiments and concurrently by taking advantage of prior knowledge from the existing experimental results, which the individual network SBM lacks.

2 Background

We first briefly review the stochastic block model for module identification in individual networks to study the modular structures [8], for which a Bayesian module identification algorithm has recently been proposed to efficiently solve the problem [9].

Given a network in a graph representation, $G = (V, E)$, where V denotes the set of all N vertices in the given network and E is the set of edges connecting the corresponding vertices in the network G. Let A be the observed $N \times N$ adjacency matrix whose elements take the values 0 or 1: $A_{ij} = 1$ indicating that there is a corresponding edge $e_{ij} \in E$ between vertices v_i and $v_j \in V$ and $A_{ij} = 0$ otherwise. We introduce the latent variable $z_i \in \{1, 2, \ldots, K\}$ to represent the module assignment of vertices v_i, and K is the total number of desirable modules. In SBM, the probability that an edge exists between two vertices depends on their module memberships. Conditioning on module assignments, the probabilities that corresponding vertices are linked follow Bernoulli distributions with the corresponding bias parameters $\theta_c = p(A_{ij} = 1 | z_i = z_j)$ and $\theta_d = p(A_{ij} = 1 | z_i \neq z_j)$, which are called within- and between-module edge probabilities, respectively. Also, SBM assumes a multinomial distribution over module assignment probabilities with parameters $\pi_k = p(z_i = k|\vec{\pi})$. With these assumptions, the joint probability of an adjacency matrix A and the corresponding module assignment vector $\vec{z}$ can be written as

$$\begin{aligned} p\left(A, \vec{z} | \vec{\theta}, \vec{\pi}, K\right) &= p\left(A | \vec{z}, \vec{\theta}\right) p\left(\vec{z} | \vec{\pi}\right) \\ &= \theta_c^{c^+} (1-\theta_c)^{c^-} \theta_d^{d^+} (1-\theta_d)^{d^-} \prod_{k=1}^{K} \pi_k^{n_k}, \end{aligned} \tag{1}$$

in which $c^+ = \sum_{i>j} A_{ij} I\left[z_i = z_j\right]$ is the number of edges contained within potential modules; $c^- = \sum_{i>j} (1 - A_{ij}) I\left[z_i = z_j\right]$ is the number of non-edges contained within modules; $d^+ = \sum_{i>j} A_{ij} I\left[z_i \neq z_j\right]$ is the number of edges between vertices across different modules; $d^- = \sum_{i>j} (1 - A_{ij}) I\left[z_i \neq z_j\right]$ is the number of non-edges across potential modules; and n_k denotes the number of vertices assigned to the kth potential module with $\sum_k n_k = N$. $I[x]$ denotes the indicator function, which equals to one if its argument x is a true logic statement and zero otherwise. The factorization of the joint probability follows from the fact that the probability of the observed adjacency matrix can be completely determined based on the given model parameters, including module assignment probabilities $\vec{\pi}$ and within- and between-module edge probabilities θ_c, θ_d.

3 Methods

We extend the above Bayesian framework for individual networks to more robust and accurate module identification across multiple networks. A variational Bayes approach is then derived to infer the unknown parameters of our extended model to identify significant modules across multiple noisy networks.

3.1 Multiple-network stochastic block model

Given multiple observed noisy networks with corresponding adjacency matrices $\left\{A^{(1)}, A^{(2)}, \ldots, A^{(T)}\right\}$, we aim to study the hidden modular structures across these networks. Without loss of generality, we assume that the set of vertices is fixed in all adjacency matrices. To infer the modular structures of these observed networks, we introduce a latent root module assignment $\vec{z}$, which can be considered to determine the connectivity of a virtual image graph illustrated in Fig. 1. For T observed networks, the corresponding instantaneous module assignments $\vec{z}^t$ for $A^{(t)}$ evolve under a transition probability matrix $P^{(t)}$. This model allows an inherent modular structure to unify all other observations to borrow strengths from each other when inferring modules of a certain network and thereby compensates for the potential detrimental effect of noise mixed with observations.

With the underlying assumption that multiple observed networks have modular structures with similar within- and between-module edge densities, we fix the edge probabilities θ_c and θ_d to be the same for all the observed networks. To fully specify this new stochastic block model, we set the root assignment matrix $\vec{z}$ to be multinomial with assignment probabilities $\vec{\pi}$. We can write the joint distribution of assignment matrices and observed adjacency matrices of this model as follows:

$$\begin{aligned} &p\left(A^{(1:T)}, \vec{z}, \vec{z}^{(1:T)} | \vec{\theta}, \vec{\pi}, P^{(1:T)}, K\right) \\ &= \left[\prod_{t=1}^{T} p\left(A^{(t)} | \vec{z}^{(t)}, \vec{\theta}\right) p\left(\vec{z}^{(t)} | \vec{z}, P^{(t)}\right)\right] p(\vec{z}|\vec{\pi}) \\ &= \theta_c^{\sum_{t=1}^{T} c_t^+} (1-\theta_c)^{\sum_{t=1}^{T} c_t^-} \theta_d^{\sum_{t=1}^{T} d_t^+} (1-\theta_d)^{\sum_{t=1}^{T} d_t^-} \\ &\quad \times \left[\prod_{t=1}^{T} \prod_{i=1}^{N} \prod_{r,s=1}^{K} P_{rs}^{I\left[z_i = r\right] \cdot I\left[z_i^{(t)} = s\right]}\right] \prod_{k=1}^{K} \pi_k^{n_k}, \end{aligned} \tag{2}$$

where a concise index representation $(1 : T)$ is adopted to denote the indices of the corresponding components in the model for multiple networks. For example, $A^{(1:T)}$ stands for T adjacency matrices $\left\{A^{(1)}, \ldots, A^{(T)}\right\}$. The corresponding numbers of edges c_t^+, c_t^-, d_t^+, and d_t^- for the tth network are defined similarly as in the model (1) for individual networks, except that the adjacency matrix A is replaced with $A^{(t)}$. Similarly, $I\left[z_i = r\right] \cdot I\left[z_i^{(t)} = s\right]$ counts the vertice v_i when it is assigned to the sth module for

the tth network and in the rth module in the root assignment; n_k is calculated from the root assignment. One immediate consequence of such modeling is that the edges that frequently appear in multiple observations have a higher chance of being true positives. Such an intuition is reflected in the likelihood function in our model. In addition, the model makes sure that the vertices connected by these edges are more likely to be assigned to the same modules in different observed networks by the proper choice of transition probabilities, which is clarified in the subsequent section.

3.2 Bayesian inference

To predict module assignments to assign memberships to all the vertices in the given networks, we resort to Bayesian inference to draw a joint posterior distribution of all the latent variables and unknown model parameters. To facilitate the computation of the posterior, we prefer more efficient variational Bayes algorithms instead of directly implementing Monte Carlo (MC) simulations. In order to derive closed-form updates for variational Bayes algorithms, we adopt conjugate prior distributions in our multiple-network clustering model. The conjugate prior for the root assignment probability distribution $\vec{\pi}$ is a Dirichlet distribution with a hyper-parameter vector $\vec{n}_0$:

$$p\left(\vec{\pi}|\vec{n}_0\right) = \frac{\Gamma\left(\sum_{k=1}^{K} n_{k,0}\right)}{\prod_{k=1}^{K}\Gamma\left(n_{k,0}\right)} \prod_{k=1}^{K} \pi_k^{n_{k,0}-1}. \tag{3}$$

Here $n_{k,0}$ is the kth component of vector $\vec{n}_0$ and $\Gamma(\cdot)$ is the gamma function. The conjugate priors for edge weights θ_c and θ_d are beta distributions with hyper-parameters $(\alpha_{c,0}, \beta_{c,0})$ and $(\alpha_{d,0}, \beta_{d,0})$, respectively,

$$\begin{aligned} p\left(\vec{\theta}|\vec{\alpha}_0, \vec{\beta}_0\right) &= p(\theta_c|\alpha_{c,0}, \beta_{c,0}) p(\theta_d|\alpha_{d,0}, \beta_{d,0}) \\ &= \frac{\Gamma(\alpha_{c,0}+\beta_{c,0})}{\Gamma(\alpha_{c,0})\Gamma(\beta_{c,0})} \theta_c^{\alpha_{c,0}-1}(1-\theta_c)^{\beta_{c,0}-1} \\ &\quad \times \frac{\Gamma(\alpha_{d,0}+\beta_{d,0})}{\Gamma(\alpha_{d,0})\Gamma(\beta_{d,0})} \theta_d^{\alpha_{d,0}-1}(1-\theta_d)^{\beta_{d,0}-1}. \end{aligned} \tag{4}$$

The underlying assumption here is that prior to observing the data, within- and between-module edge weights are independent, so their joint prior distribution factorizes. The transition probability matrices $P^{(t)}$ are stochastic, and therefore, their rows add up to 1. For each matrix $P^{(t)}$, where $t \in \{1, 2, \ldots, T\}$, we use Dirichlet prior distributions with a hyper-parameter vector $\vec{\eta}_k^{(0)}$ on rows

$$\begin{aligned} p\left(P^{(t)}|\vec{\eta}_1^{(0)}, \ldots, \vec{\eta}_K^{(0)}\right) &= \prod_{k=1}^{K} p\left(\vec{P}_k^{(t)}|\vec{\eta}_k^{(0)}\right) \\ &= \prod_{k=1}^{K} \frac{\Gamma\left(\sum_{m=1}^{K}\eta_{k,m}^{(0)}\right)}{\prod_{m=1}^{K}\Gamma\left(\eta_{k,m}^{(0)}\right)} \times \prod_{m=1}^{K}\left(P_{km}^{(t)}\right)^{\eta_{k,m}^{(0)}-1}, \end{aligned} \tag{5}$$

where $\vec{P}_k^{(t)}$ is the kth row of the transition probability matrix $P^{(t)}$, $P_{km}^{(t)}$ is its mth element, and $\eta_{k,m}^{(0)}$ is the mth element of $\vec{\eta}_k^{(0)}$. The rows of transition probability matrices are assumed to be independent, and also, we set their hyper-parameter vectors to be identical.

To further ensure that our model captures the modular structure inherent in the observed networks, we set hyper-parameters of prior beta distributions over edge weights to bias towards edge weights with within-module edge weights being greater than between-module edge weights, and this is controlled through appropriate settings of hyper-parameters of prior beta distributions over edge weights. For the model to be capable of benefiting from the structural information inferred from other networks, we prefer that the diagonal entries of transition probability matrices $P^{(t)}$ to be higher than the off-diagonal entries of those matrices, which can be achieved by setting higher hyper-parameters in the corresponding Dirichlet distributions.

With these incorporated conjugate priors, their functional forms are preserved in the posterior, a variational Bayes algorithm with closed-form updates can be derived to infer the model parameters, and, more importantly, module memberships from the aforementioned model (2) in the subsequent section.

3.3 Variational Bayes solution

Variational Bayes method is an efficient alternative to Monte Carlo sampling methods [16, 17] for statistical inference over complicated models as direct sampling is not tractable and computationally prohibitive. Under appropriate settings, variational Bayes algorithms can be derived to infer the desired posterior distributions with comparable accuracies at a greater speed, which is essential for the analysis of large-scale networks. The variational Bayes method seeks a restricted family of approximation distributions $q(\cdot)$, which minimize the Kullback-Leibler (KL) divergence between the joint probability distributions of unknown parameters and their approximate joint probability distributions [18]. For our proposed model, the quantity to be minimized takes the following form:

$$\begin{aligned} F\left\{q, A^{(1:T)}\right\} = -\sum_{\vec{z},\vec{z}^{(1:T)}} \int\int \left[q\left(\vec{z}, \vec{z}^{(1:T)}, \vec{\theta}, \vec{\pi}\right) \right. \\ \left. \times \ln \frac{p\left(A^{(1:T)}, \vec{z}, \vec{z}^{(1:T)}, \vec{\theta}, \vec{\pi}|K\right)}{q\left(\vec{z}, \vec{z}^{(1:T)}, \vec{\theta}, \vec{\pi}\right)} \right] d\vec{\theta} d\vec{\pi}. \end{aligned} \tag{6}$$

To simplify this optimization problem of minimizing the free energy $F\left\{q, A^{(1:T)}\right\}$, we follow the mean field approximation framework developed in physics [9]. To

be specific, we factorize the variational or approximate distribution $q(\cdot)$ with respect to its arguments:

$$q\left(\vec{z},\vec{z}^{(1:T)},\vec{\theta},\vec{\pi}\right)=q_{\vec{\theta}}(\vec{\theta})q_{\vec{\pi}}(\vec{\pi})q_{\vec{z}}(\vec{z})\prod_{t=1}^{T}q_{\vec{z}^{(t)}}\left(\vec{z}^{(t)}\right). \quad (7)$$

After this simplification, it can be shown that the optimal approximate distribution $q_{\vec{z}}$ for the root module assignment $\vec{z}$ satisfies the following equation [18]:

$$\ln q^{*}_{\vec{z}}(\vec{z})\propto E_{-\vec{z}}\left[\ln p\left(A^{(1:T)},\vec{z},\vec{z}^{(1:T)},\vec{\theta},\vec{\pi}\,|K\right)\right], \quad (8)$$

where $E_{-\vec{z}}[\cdot]$ denotes the expectation taken over all the parameters and latent variables except $\vec{z}$. Similar equations can be derived for $\vec{\pi}$, $\vec{\theta}$, and $\vec{z}^{(t)}$ for $t\in\{1,2,\ldots,T\}$. Solving the above Eq. (8) for all the unknown parameters leads to the complete derivation of the approximate distributions.

Particularly, these distributions belong to the same family as prior distributions, i.e., the approximate distributions of θ_c, θ_d, and $\vec{\pi}$ are respectively beta, beta, and Dirichlet distributions with hyper-parameters $\left(\tilde{\alpha}_c,\tilde{\beta}_c\right)$, $\left(\tilde{\alpha}_d,\tilde{\beta}_d\right)$, and $\tilde{\vec{n}}$. In order to calculate the posterior approximate distribution of module assignments, we factorize them as $q(z_i=k)=Q_{ik}$ and $q\left(z_i^t=k\right)=Q_{ik}^{(t)}$ for $i\in\{1,2,\ldots,N\}$, $t\in\{1,2,\ldots,T\}$, and $k\in\{1,2,\ldots,K\}$. Q and $Q^{(t)}$ are $N\times K$ matrices, in which the ith row denotes the probability of assigning vertex v_i to different potential modules.

The variational Bayes algorithm iterates between two stages. In the first step, the current distributions over the model parameters are used to evaluate the module assignment matrices Q and $Q^{(t)}$; and in the second step, these memberships are fixed and variational distributions over model parameters are updated. The resulting iterative algorithm then can be summarized as:

Initialization. Initialize $N\times K$ matrices Q and $Q^{(t)}$ for $t\in\{1,2,\ldots,T\}$ and set $\tilde{\alpha}_c=\alpha_{c,0}$, $\tilde{\beta}_c=\beta_{c,0}$, $\tilde{\alpha}_d=\alpha_{d,0}$, $\tilde{\beta}_d=\beta_{d,0}$, and $\tilde{\vec{n}}=\vec{n}_0$.

(i) Update the following expected values:

$$E\left[\ln\pi_k\right]=\psi(\tilde{n}_k)-\psi\left(\sum_{k=1}^{K}\tilde{n}_k\right); \quad (9)$$

$$E\left[\ln P_{km}^{(t)}\right]=\psi\left(\tilde{\eta}_{k,m}^{(t)}\right)-\psi\left(\sum_{m=1}^{K}\tilde{\eta}_{k,m}^{(t)}\right); \quad (10)$$

$$E\left[\ln\frac{1-\theta_d}{1-\theta_c}\right]=\psi\left(\tilde{\beta}_d\right)-\psi\left(\tilde{\alpha}_d+\tilde{\beta}_d\right)-\psi\left(\tilde{\beta}_c\right)+\psi\left(\tilde{\alpha}_c+\tilde{\beta}_c\right); \quad (11)$$

$$E\left[\ln\frac{1-\theta_d}{1-\theta_c}+\ln\frac{\theta_c}{\theta_d}\right]=\psi(\tilde{\alpha}_c)-\psi(\tilde{\beta}_c)-\psi(\tilde{\alpha}_d)+\psi(\tilde{\beta}_d), \quad (12)$$

where $\psi(\cdot)$ is the digamma function.

(ii) Update the variational distribution over the root module assignment:

$$Q_{ik}\propto\exp\left\{E\left[\ln\pi_k\right]+\sum_{t=1}^{T}\sum_{m=1}^{K}Q_{im}^{(t)}E\left[\ln P_{km}^{(t)}\right]\right\}. \quad (13)$$

Normalize Q such that $\sum_{k=1}^{K}Q_{ik}=1$ for all vertices v_i.

(iii) Update the variational distributions over instantaneous module assignments for $t\in\{1,2,\ldots,T\}$:

$$Q_{ik}^{(t)}\propto\exp\left\{\sum_{j\neq i}\left(E\left[\ln\frac{1-\theta_d}{1-\theta_c}+\ln\frac{\theta_c}{\theta_d}\right]A_{ij}^{(t)}-E\left[\ln\frac{1-\theta_d}{1-\theta_c}\right]\right)Q_{jk}^{(t)}+\sum_{s=1}^{K}Q_{is}\left[\ln P_{sk}^{(t)}\right]\right\}. \quad (14)$$

Normalize $Q^{(t)}$ such that $\sum_{k=1}^{K}Q_{ik}^{(t)}=1$ for all vertices v_i.

(iv) Update the posterior hyper-parameters of the Dirichlet distribution over the root module assignment of vertices:

$$n_k=\sum_{i=1}^{N}Q_{ik}+n_{k,0}. \quad (15)$$

(v) Consider $\eta^{(t)}$ for $t\in\{1,2,\ldots,T\}$ as a matrix whose elements are $\eta_{k,m}^{(t)}$. Then, update the matrix $\eta^{(t)}$ as follows:

$$\eta^{(t)}=Q'Q^{(t)}+\eta^{(0)}, \quad (16)$$

where Q' is the transpose of the matrix Q and $\eta^{(0)}$ is the matrix of prior hyper-parameters of transition probability matrices.

(vi) Update the hyper-parameters of beta distributions over edge weights:

$$\tilde{\alpha}_c=\frac{1}{2}\sum_{t=1}^{T}Tr\left(Q^{(t)'}A^{(t)}Q^{(t)}\right)+\alpha_{c,0}; \quad (17)$$

$$\tilde{\beta}_c=\frac{1}{2}\sum_{t=1}^{T}Tr\left(Q^{(t)'}\left(\vec{u}\vec{v}^{(t)'}-Q^{(t)}\right)\right)-\frac{1}{2}\sum_{t=1}^{T}Tr\left(Q^{(t)'}A^{(t)}Q^{(t)}\right)+\beta_{c,0}; \quad (18)$$

$$\tilde{\alpha}_d = \sum_{t=1}^{T}\sum_{i>j} A_{ij}^{(t)} - \frac{1}{2}\sum_{t=1}^{T} Tr\left(Q^{(t)'}A^{(t)}Q^{(t)}\right) + \alpha_{d,0}; \quad (19)$$

$$\tilde{\beta}_d = \sum_{t=1}^{T}\sum_{i>j}\left(1 - A_{ij}^{(t)}\right) - \frac{1}{2}\sum_{t=1}^{T} Tr\left(Q^{(t)'}\left(\vec{u}\vec{v}^{(t)'} - Q^{(t)}\right)\right) + \frac{1}{2}\sum_{t=1}^{T} Tr\left(Q^{(t)'}A^{(t)}Q^{(t)}\right) + \beta_{d,0}, \quad (20)$$

where $\vec{u}$ is a $N \times 1$ vector of ones and $\vec{v}^{(t)}$ is a vector with elements $v_k^{(t)} = \sum_{i=1}^{N} Q_{ik}^{(t)}$.

(vii) Calculate the updated free energy:

$$F\left\{q^*, A^{(1:T)}\right\} = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{K} Q_{ik}^{(t)} \ln Q_{ik}^{(t)} + \sum_{i=1}^{N}\sum_{k=1}^{K} Q_{ik} \ln Q_{ik} - \sum_{t=1}^{T}\sum_{k=1}^{K} \ln \frac{B\left(\tilde{\vec{\eta}}_k^{(t)}\right)}{B\left(\vec{\eta}_k^{(0)}\right)} - \ln \frac{B\left(\tilde{\alpha}_c, \tilde{\beta}_c\right) B\left(\tilde{\alpha}_d, \tilde{\beta}_d\right) B\left(\tilde{\vec{n}}\right)}{B\left(\alpha_{c,0}, \beta_{c,0}\right) B\left(\alpha_{d,0}, \beta_{d,0}\right) B(\vec{n}_0)}, \quad (21)$$

where $B(\cdot)$ is a beta function with the vector argument.

The optimized free energy in (21) decreases in consecutive iterations, and thereby, this algorithm is guaranteed to converge to a local optimum. In the case where the posterior is multi-modal, several initializations should be tested to ensure the quality of the returned solutions.

4 Experimental results

In this section, we evaluate our Bayesian module identification across multiple networks by testing the derived variational Bayes algorithm on both synthetic and real-world PPI data sets. The obtained results also are compared with Bayesian module identification with individual networks and another state-of-the-art network clustering algorithm—ClusterOne [19]. For synthetic data, we have the ground-truth module memberships and therefore we use normalized mutual information to assess the performance of our model. Normalized mutual information (NMI) is defined as follows [3, 13]:

$$\mathrm{NMI}(\mathcal{C}, \mathcal{C}') = \frac{\hat{\mathrm{MI}}\left(\mathcal{C}, \mathcal{C}'\right)}{\max(H(\mathcal{C}), H(\mathcal{C}'))}, \quad (22)$$

where $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ denotes the true assignments of vertices to corresponding modules and $\mathcal{C}' = \left\{C_1', C_2', \ldots, C_K'\right\}$ denotes the inferred module memberships of vertices by the implemented algorithms. $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of the ground truth and inferred modules. $\hat{\mathrm{MI}}(\mathcal{C}, \mathcal{C}')$ is the mutual information calculated by $\hat{\mathrm{MI}}(\mathcal{C}, \mathcal{C}') = \sum_{C_i, C_j'} p\left(C_i, C_j'\right) \ln \frac{p\left(C_i, C_j'\right)}{p\left(C_i\right)p\left(C_j'\right)}$.

For real-world data sets, we analyze two budding yeast (*Saccharomyces cerevisiae*) PPI networks obtained from the Database of Interaction Proteins (DIP) [14] and the Biological General Repository for Interaction Datasets (BioGRID) [15] to predict protein complexes. The *predicted protein complexes* as inferred modules by the selected algorithms are then verified against the Saccharomyces Genome Database (SGD) [20] and Munich Information Center for Protein Sequences (MIPS) [21] golden standards as the *reference complexes*. To validate the predicted protein complexes by the selected algorithms, we adopt the same performance metrics introduced in [3, 19]. The first metric is the fraction of pairs between the predicted and reference complexes with an overlap score of larger than 0.25. We represent this metric in the results with Frac. The overlap score ω between two sets of vertices, proteins in this case, V_1 and V_2, is defined as:

$$\omega(V_1, V_2) = \frac{|V_1 \cap V_2|^2}{|V_1||V_2|}, \quad (23)$$

where $|\cdot|$ denotes the cardinality of a set. The threshold 0.25 used for ω is achieved when two equally sized protein complexes have an intersection set with half of their size.

The second performance metric is the geometric accuracy (Acc), which is the geometric mean of two measures: the module-wise sensitivity (Sn) and module-wise positive predictive value (PPV): $\mathrm{Acc} = \sqrt{\mathrm{PPV} \times \mathrm{Sn}}$. Given n reference and m predicted protein complexes, t_{ij} denotes the number of proteins that are the members of both the reference complex i: $1 \le i \le n$ and predicted complex j: $1 \le j \le m$. Furthermore, let n_i represent the total number of proteins in the reference complex i. The two measures Sn and PPV for computing the geometric accuracy are defined as:

$$\mathrm{Sn} = \frac{\sum_{i=1}^{n} \max_{j\in\{1,\ldots,m\}} t_{ij}}{\sum_{i=1}^{n} n_i};$$

$$\mathrm{PPV} = \frac{\sum_{j=1}^{m} \max_{i\in\{1,\ldots,n\}} t_{ij}}{\sum_{j=1}^{m}\sum_{i=1}^{n} t_{ij}}.$$

Since Sn can be maximized by putting every protein in the same module and PPV can be maximized by assigning each protein in a distinct module, the Acc is considered a better performance metric that we adopt.

When some proteins appear in either none of the predicted complexes or in more than one complexes, the value of PPV can be misleading [3, 19]. To mitigate such deficiencies, the authors in [22] have introduced an additional metric called module-wise separation (Sep) for fair comparison. To define it, we first introduce the marginal

row-wise and column-wise relative frequencies that can be computed as:

$$F_{ij}^{r} = \frac{t_{ij}}{\sum_{j=1}^{m} t_{ij}};$$
$$F_{ij}^{c} = \frac{t_{ij}}{\sum_{i=1}^{n} t_{ij}}.$$

The separation of the predicted complex i and reference complex j then equals to $\text{Sep}_{ij} = F_{ij}^{r} F_{ij}^{c}$. The reference-wise and inferred-module-wise scores are calculated for the whole set of the reference and predicted complexes as:

$$\text{Sep}_{\text{ref}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \text{Sep}_{ij}}{m};$$
$$\text{Sep}_{\text{inf}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \text{Sep}_{ij}}{n}.$$

The final separation score is computed from these two quantities as $\text{Sep} = \sqrt{\text{Sep}_{\text{ref}}\text{Sep}_{\text{inf}}}$. $\text{Sep}_{ij} = 1$ indicates that the reference complex j is a perfect match for predicted complex i and both of them contain identical proteins.

4.1 Synthetic networks

Using the same procedure as in [1], we generate a synthetic network with $N = 128$ vertices and $K = 4$ modules, each module containing 32 vertices. The average degree of vertices is set to 16, and the average number of between-module edges of each vertex is set to 6. To generate the network, we first assign vertices to different modules by following a multinomial distribution with the equal weights for all modules. Then, each pair of vertices are connected with the probabilities equal to θ_c or θ_d if they belong to the same or different modules, respectively.

To simulate multiple observed noisy networks, we implement the Sneppen and Maslov re-wiring method [23] to construct new networks and adjacency matrices with instilled noise based on the original network generated as described above. In this method, a pair of edges $v_i \leftrightarrow v_j$ and $v_k \leftrightarrow v_\ell$ are randomly selected and then re-wired such that v_i becomes connected to v_ℓ, while v_j to v_k, provided that none of these edges existed previously. This method preserves the degree of each vertex and thus global topological properties, including edge densities in perturbed networks, do not change significantly.

Following this procedure, we generate two different sets of simulated networks. In the first experiment, we generate $T = 10$ adjacency matrices where the number of randomly selected re-wirings increases linearly from 5 to 50 % of the total number of edges in nine steps gradually. The adjacency matrices for two extreme cases at $t = 1$ and $t = 10$ are shown in Fig. 2a respectively on top and bottom rows, which reflect different levels of introduced noise. In the second experiment, $T = 10$ adjacency matrices are generated from an original adjacency matrix by randomly re-wiring 25 % of the total number of edges. Thereby, here, the noise levels are consistent across all the networks. Note that the re-wirings are independent from each other. Intuitively, in the first set of networks, module identification becomes more difficult with increasing noise levels while in the second experiment, it is similarly difficult when identifying modules in respective networks.

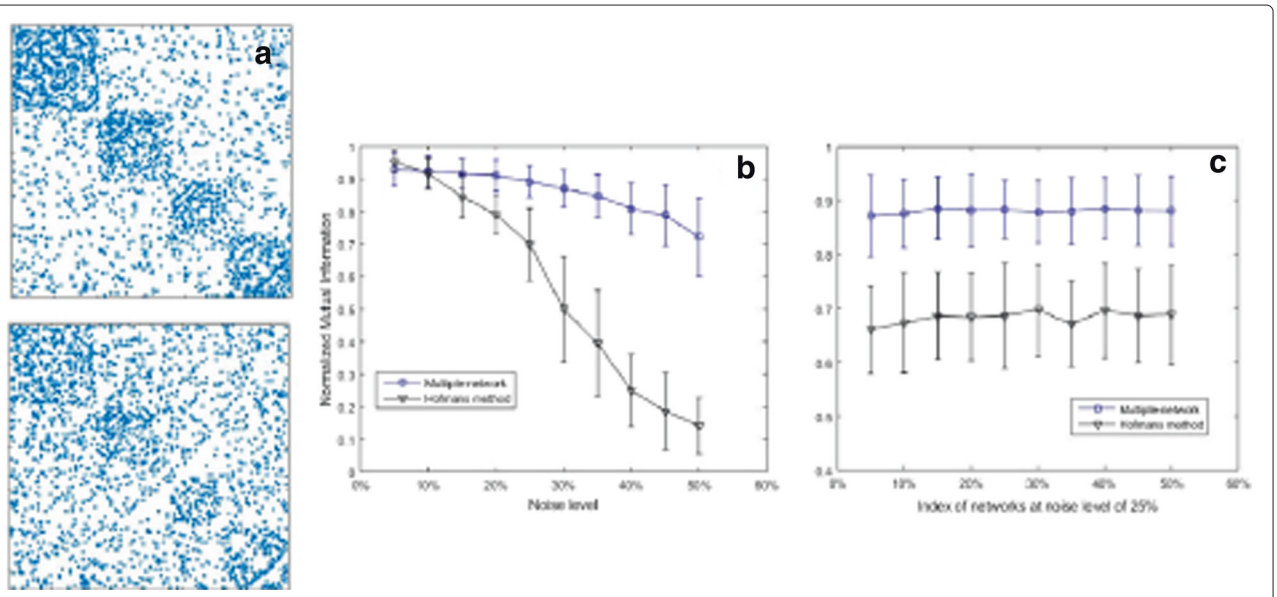

Fig. 2 Performance evaluation with synthetic networks. **a** Examples of noisy networks: The top network is with the lowest noise level (5 %) and the bottom network is with the highest noise level (50 %). **b** Performance results of the first set of experiments, where the instilled noise level increases gradually. **c** Performance results of the second set of experiments, where the noise level remains constant

To demonstrate that our Bayesian module identification across multiple networks can better identify modules by borrowing strength across networks, we compare the results by our algorithm on the set of ten randomly perturbed networks with those of Hofman's algorithm [9] applied to individual networks. Since we assume no prior knowledge on module memberships of the vertices, the initial hyper-parameters for the Dirichlet distributions for module assignments are set to equal values for all $K = 4$ modules. Empirically, neither of the algorithms is sensitive to hyper-parameters for beta distributions over edge weights, provided that within-module edge weights are larger than the between-module counterpart. Based on our experiments, Hofman's implementation on individual networks may not converge to the global optimum, especially when we have high levels of introduced noise, for example, when we have 20 % random re-wirings ($t > 4$) in our experiments. On the contrast, we find that multiple random initializations may not be necessary for our multiple-network clustering algorithm. In order to have a fair comparison with the satisfactory solution quality, we take 100 random initializations for both algorithms.

Figure 2b and c depict the average normalized mutual information in two experiments between module detection results of both algorithms and the true module membership, by which data has been generated, based on averaged 100 independent repeats with the aforementioned settings. As it can be seen in these figures, as the noise level increases, the difference between the performances of two algorithms gets more significant. For highly noisy adjacency matrices, Hofman's algorithm indeed fails to recover the module memberships accurately. Nonetheless, our algorithm by borrowing information from other observations returns satisfying results. In the second experiment, we can clearly observe the superiority of our model as there is an approximate 0.2 difference in the normalized mutual information measure in favor of our algorithm across all networks. Thus, aggregating information across networks has led to higher accuracy in predicting the module membership of vertices achieved by our method.

4.2 Edge prediction

To further validate our model, we simulate two networks from a single "ground truth" network with the specifications identical to the networks generated in the previous section, by instilling 25 % noisy edges based on the Sneppen and Maslov re-wiring method. To test the capability of our model to predict missing edges of the ground truth network, we randomly hold out different percentages of edges of the first network and use the remaining edges (and the other network in the case of our multiple-network model) to predict the probability of each missing edge to exist. Compared to previous experiments, we only need to slightly modify the inference by discarding the held-out edges from the likelihood. The probability of an edge existing between nodes i and j in the model can be calculated by

$$p(A_{ij}=1)=\left(\frac{\tilde{\alpha}_c}{\tilde{\alpha}_c+\tilde{\beta}_c}-\frac{\tilde{\alpha}_d}{\tilde{\alpha}_d+\tilde{\beta}_d}\right)\sum_{k=1}^{K}Q_{ik}^{(t)}Q_{jk}^{(t)}+\frac{\tilde{\alpha}_d}{\tilde{\alpha}_d+\tilde{\beta}_d}.$$

We consider different training ratios (the percentage of remaining edges in one network) ranging from 20 to 80 % with 10 % increments. Figures 3 and 4 show the error bar plots of Area Under the Curves (AUC) of both the Receiver Operating Characteristic (ROC) and Precision-recall (PR) for different training ratios, respectively. As expected, our multiple-network method outperforms Hofman's model in terms of both ROC and PR as it takes advantage of the shared information across observations. As shown in Figs. 3 and 4, decreasing the number of held-out edges leads to the growing margin between the performances of the two approaches.

4.3 Protein complex prediction

We further have applied our Bayesian module identification to unweighted yeast PPI networks, extracted from DIP and BioGrid, to predict protein complexes. Each of these networks have 4540 proteins and the number of edges in DIP and BioGrid networks are 21,326 and 49,128, respectively. Besides our algorithm, ClusterOne [19] and Hofman's method [9] also have been applied to the networks for comparison. ClusterOne is a greedy algorithm that can be considered as an overlapping extension of normalized cut spectral clustering. For both Hofman's and our algorithm, we need to decide the value of K for the number of potential modules. However, we note that both algorithms are in the Bayesian framework and thereby

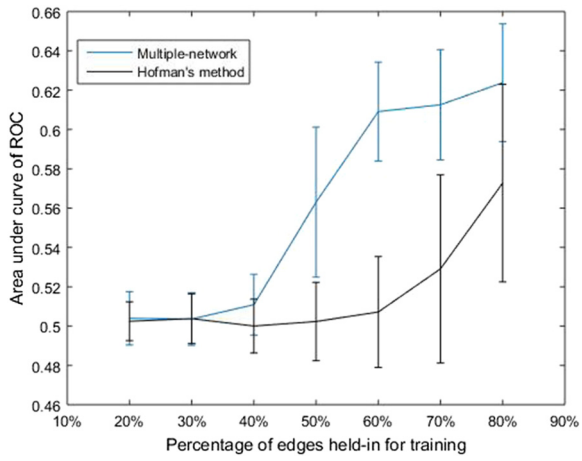

Fig. 3 Error bar plots of Area Under the Curve (AUC) of ROC for training ratios ranging from 20 to 80 %

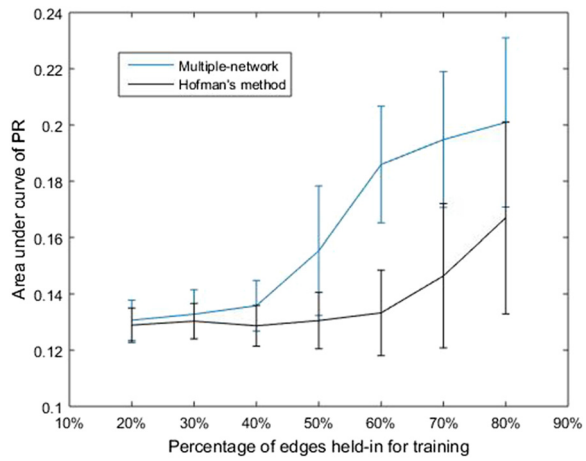

Fig. 4 Error bar plots of Area Under the Curve (AUC) of Precision-recall (PR) for training ratios ranging from 20 to 80 %

the full probability of memberships of different modules can be determined. With the large enough K, model likelihoods for different Ks can be evaluated to determine the optimal K. In the current experiments, we also focus on non-overlapping module identification as done in [9] for fair comparison by assigning each vertex v_i to the kth module that maximizes $Q_{ik}^{(t)}$ in the tth network. Based on the average size of protein complexes given in yeast golden standards, which is approximately 5 in both SGD and MIPS, we set K =1000 considering the size of our PPI networks.

The results of these three methods are compared based on the parameters introduced earlier. Table 1 contains the performance comparison between these algorithms based on the SGD golden standard while Table 2 provides the performance comparison based on the MIPS standard. One advantage of Bayesian methods is that the identified complexes cover the whole set of proteins present in the data sets, while ClusterOne only discovers overlapping complexes that include 1372 and 2340 proteins for DIP

Table 1 Performance comparison of different algorithms based on SGD golden standard

Data set	Metric	Multiple network	ClusterOne	Hofman
DIP	Acc	**0.5435**	0.4731	0.4561
	Frac	0.2129	**0.3194**	0.1000
	PPV	0.4648	**0.5528**	0.3295
	Sep	**0.3511**	0.3329	0.3146
BioGRID	Acc	**0.6110**	0.5961	0.5549
	Frac	0.2097	**0.4839**	0.1871
	PPV	0.4738	**0.5663**	0.4612
	Sep	**0.3999**	0.3325	0.3505

The best indices are highlighted with bold fonts

Table 2 Performance comparison of different algorithms based on MIPS golden standard

Data set	Metric	Multiple network	ClusterOne	Hofman
DIP	Acc	**0.3933**	0.3178	0.3403
	Frac	0.2381	**0.3598**	0.1111
	PPV	0.3567	**0.4076**	0.2651
	Sep	**0.2535**	0.2216	0.2020
BioGRID	Acc	**0.4614**	0.4336	0.4383
	Frac	0.2975	**0.4974**	0.2275
	PPV	0.3713	**0.4207**	0.3649
	Sep	**0.2536**	0.2193	0.2189

The best indices are highlighted with bold fonts

and BioGRID datasets respectively out of the total 4540 proteins in the original networks. As a consequence, complexes found by Bayesian methods have larger sizes and therefore possess lower fraction of matched proteins compared to the reference complexes, which are reflected as low PPV values in both tables. Nonetheless, the results with respect to the Acc have shown that our new Bayesian module identification with multiple networks can better predict potential protein complexes, in which proteins are densely connected; and thereby, these methods are more useful when the final objective is to predict new protein complexes. Another observation in the results is that combining information from different observed networks in our model enhances both the Acc and Sep metrics compared with ClusterOne and Hofman's method, especially for the DIP network, a relatively sparse network, for which detecting inherent modules is a difficult task. Table 3 depicts the number of predicted complexes for different data sets by different algorithms.

It is clear that, compared to its individual-network counterpart—Hofman's method—our algorithm is able to find significantly more protein complexes and thus obtaining a better understanding of both networks' modular structures. One example is illustrated in Fig. 5. This figure demonstrates the interactions among a group of proteins in the DIP data set. All of these proteins are assigned to the same complex by Hofman's method. However, our multiple-network method identified a more densely connected subset of proteins. This discovered protein complex contains all of the proteins in the reference RNA polymerase II mediator protein complex in

Table 3 Number of identified protein complexes by different algorithms for DIP and BioGRID data sets

Data set	Multiple network	ClusterOne	Hofman
DIP	320	328	112
BioGRID	278	424	189

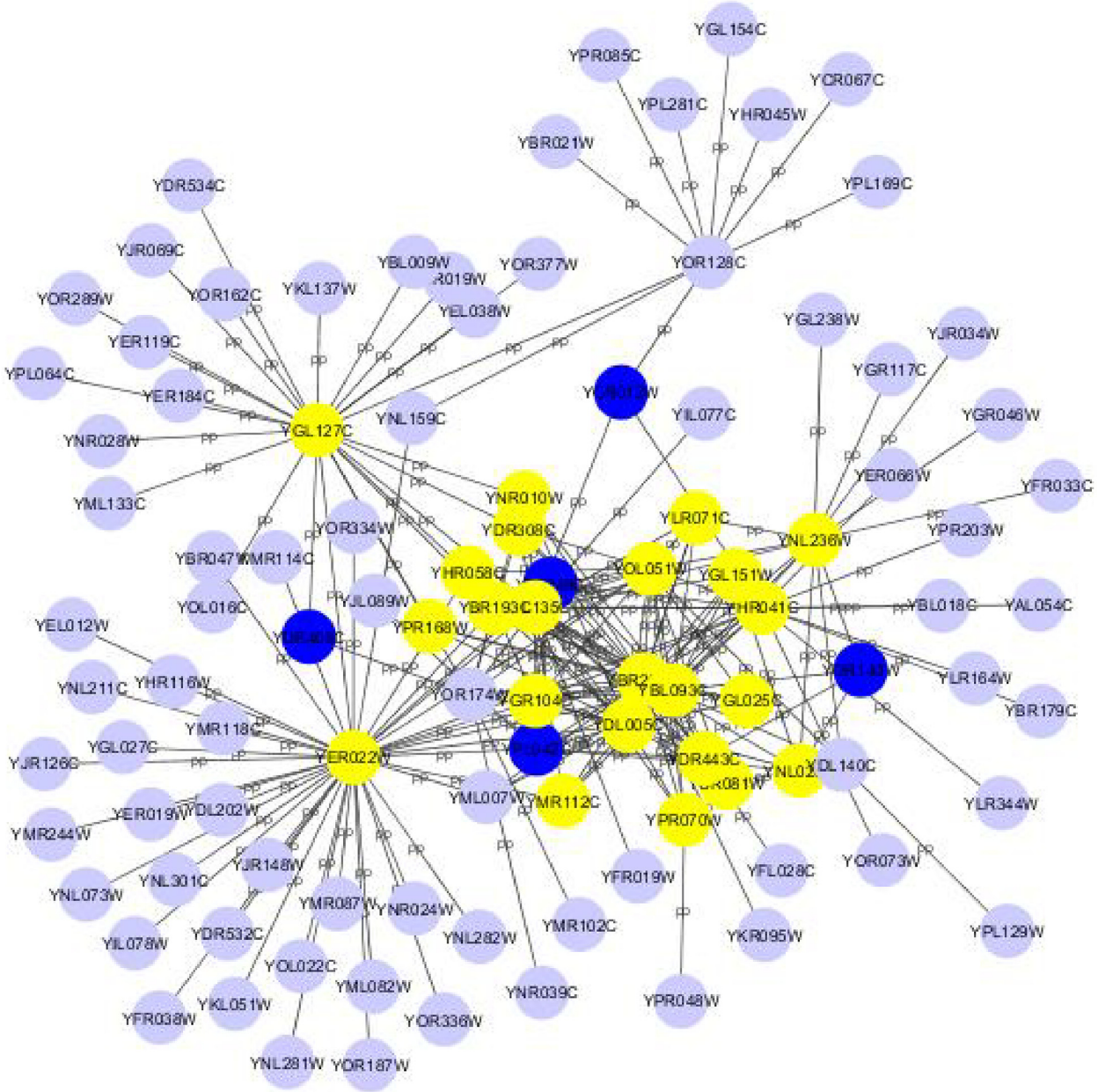

Fig. 5 One example of the predicted protein complexes by Hofman's and our multiple-network clustering algorithms. The whole set of proteins were considered as a single complex by Hofman's method, while the proteins colored in *yellow* and *dark blue* form the predicted complex returned by our new multiple-network clustering method. The colored proteins in *yellow* are the member proteins in the RNA polymerase II mediator protein complex in SGD golden standard. This figure is produced by [24]

SGD golden standard. These proteins are colored in yellow in the figure, and other proteins that do not belong to this reference complex but are members of the predicted complex by our algorithm are colored in blue. Thus, we can observe that our method is capable of increasing the resolution in module identification. Note that the number of modules found by our algorithm is comparable to that of ClusterOne, though the modules in our method are disjoint. Since we have the posterior distribution of module assignments of all vertices, one can easily construct lots of overlapping complexes by allowing each vertex to belong to more than one modules.

5 Conclusions

In this paper, we generalize the variational Bayes algorithm for module identification in individual networks

[9] to a new stochastic block model with the efficient accompanying variational Bayes algorithm for module identification across multiple noisy observed networks. The effectiveness and efficiency of our algorithm with improved accuracy and resolution have been demonstrated on both synthetic and real-world PPI networks. In our future work, we will focus on finding solution methods for module identification from multiple networks with more general noise models.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
XQ conceived and designed the experiments. SZD and XQ designed and implemented the algorithm. SZD performed the experiments. SZD and XQ analyzed the results and wrote the paper. Both authors read and approved the final manuscript.

Acknowledgements
This work was partially supported by Awards #1447235 and #1244068 from the National Science Foundation, as well as Award R21DK092845 from the National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health.

Received: 26 May 2015 Accepted: 20 January 2016
Published online: 05 February 2016

References

1. ME Newman, M Girvan, Finding and evaluating community structure in networks. Phys. Rev. E. **69**(2), 026113 (2004)
2. J Shi, J Malik, Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 888–905 (2000)
3. Y Wang, X Qian, Functional module identification in protein interaction networks by interaction patterns. Bioinformatics. **30**, 569 (2013)
4. S White, P Smyth, A spectral clustering approach to finding communities in graph. SDM. **5**, 76–84 (2005)
5. Y Wang, X Qian, Joint clustering of protein interaction networks through Markov random walk. BMC Syst. Biol. **8**(1), 9 (2014)
6. K Yu, S Yu, V Tresp, in *Advances in Neural Information Processing Systems*. Soft clustering on graphs, (2005), p. 05
7. Y Wang, X Qian, in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. Biological network clustering by robust NMF (ACM, New York, NY, USA, 2014), pp. 688–689
8. PW Holland, S Leinhardt, Local structure in social networks. Sociol. Method. **7**, 1–45 (1976)
9. JK Hofman, CH Wiggins, Bayesian approach to network modularity. Phys. Rev. Lett. **100**, 258701 (2008)
10. R Kumar, J Novak, P Raghavan, A Tomkins, On the bursty evolution of Blogspace. World Wide Web. **8**(2), 159–178 (2005)
11. Y Chi, X Song, D Zhou, K Hino, BL Tseng, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Evolutionary spectral clustering by incorporating temporal smoothness (ACM, New York, NY, USA, 2007), pp. 153–162
12. Y-R Lin, Y Chi, S Zhu, H Sundaram, BL Tseng, in *Proceedings of the 17th International Conference on World Wide Web*. FacetNet: a framework for analyzing communities and their evolutions in dynamic networks (ACM, New York, NY, USA, 2008), pp. 685–694
13. T Yang, Y Chi, S Zhu, Y Gong, R Jin, Detecting communities and their evolutions in dynamic social networks—a Bayesian approach. Mach. Learn. **82**(2), 157–189 (2011)
14. L Salwinski, CS Miller, AJ Smith, FK Pettit, JU Bowie, D Eisenberg, The database of interacting proteins: 2004 update. Nucleic Acids Res. **32**(1), 449–451 (2004)
15. B-J Breitkreutz, C Stark, T Reguly, L Boucher, A Breitkreutz, M Livstone, R Oughtred, DH Lackner, J Bähler, V Wood, *et al*, The BioGRID interaction database: 2008 update. Nucleic Acids Res. **36**(1), 637–640 (2008)
16. S Geman, D Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**(6), 721–741 (1984)
17. S Chib, E Greenberg, Understanding the Metropolis-Hastings algorithm. The American Statistician. **49**(4), 327–335 (1995)
18. CM Bishop, *et al*, *Pattern recognition and machine learning*. (Springer, New York, 2006)
19. T Nepusz, H Yu, A Paccanaro, Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods. **9**(5), 471–472 (2012)
20. EL Hong, R Balakrishnan, Q Dong, KR Christie, J Park, G Binkley, MC Costanzo, SS Dwight, SR Engel, DG Fisk, *et al*, Gene Ontology annotations at SGD: new data sources and annotation methods. Nucleic Acids Res. **36**(1), 577–581 (2008)
21. H-W Mewes, C Amid, R Arnold, D Frishman, U Güldener, G Mannhaupt, M Münsterkötter, P Pagel, N Strack, V Stümpflen, *et al*, MIPS: analysis and annotation of proteins from whole genomes. Nucleic Acids Res. **32**(1), 41–44 (2004)
22. S Brohee, J Van Helden, Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics. **7**(1), 488 (2006)
23. S Maslov, K Sneppen, Specificity and stability in topology of protein networks. Science. **296**(5569), 910–913 (2002)
24. P Shannon, A Markiel, O Ozier, NS Baliga, JT Wang, D Ramage, N Amin, B Schwikowski, T Ideker, Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. **13**(11), 2498–2504 (2003)