

JOINT CLUSTERING OF PROTEIN INTERACTION NETWORKS BY BLOCK MODELING

Yijie Wang and Xiaoning Qian

Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA

ABSTRACT

Identification of functional modules in protein protein interaction (PPI) networks may help better understand cell functions. Many existing computational methods focus on identifying modules based on either individual PPI networks or protein sequence similarities within the species. As both interaction data and sequence similarities may not be either complete or accurate with respect to revealing protein functionalities, we propose a joint clustering framework based on block modeling to integrate the available information across different species to utilize both protein interaction data and sequence similarities. The motivation is to borrow strengths from multiple data sources for more accurate module identification as evolutionally different species may share similar cellular organization. Our blockmodel joint clustering enables the identification of not only densely connected modules but also those modules containing proteins with similar interaction patterns to the rest of the networks. We develop a simulated annealing (SA) algorithm based on Potts-Models for the blockmodel problem to solve the non-convex combinatorial optimization. Our method is validated using synthetic networks as well as yeast and fruit fly PPI networks. The experimental results conclude that joint clustering outperforms clustering of individual networks separately.

Index Terms— Joint clustering; PPI network; Protein sequence similarity; Simulated annealing

1. INTRODUCTION

In computational biology, network clustering helps identify groups of molecules that share similar constituent, topological, or functional properties and collaborate to achieve coherent and distinct cellular functions [1]. Many existing algorithms focus on identifying “clusters” as functional modules which have more than expected internal interactions. However, the mathematically “optimal” clustering based on this *modularity*-based definition may not necessarily capture the actual functional organization of biological networks [2]. As pointed out in [3], there are more structures in biological networks other than densely self-connected modules. To identify biologically meaningful functional modules with diverse topological structures, blockmodel module identification [3, 4, 5, 6] have been proposed by taking into account the roles of proteins in networks reflected by their interaction patterns.

XQ is supported by NSF/MCB #1244068 and NIH/NIDDK DK092845.

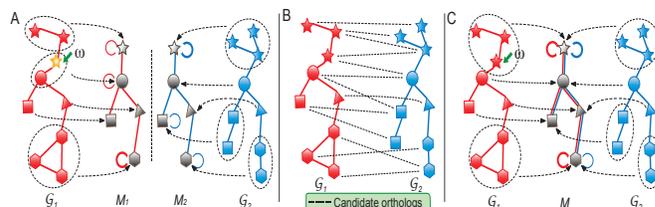


Fig. 1. **A.** Network clustering separately for two networks \mathcal{G}_1 and \mathcal{G}_2 ; **B.** Sequence similarities between \mathcal{G}_1 and \mathcal{G}_2 ; **C.** Joint network clustering. The width of edges in the virtual modular space is proportional to the number of aggregated edges in original networks.

Recently, comparative network analysis [7, 8, 9] has demonstrated to be able to identify functional modules that are evolutionarily conserved across species by identifying similar subnetworks or cohesive mappings based on similarity between biomolecules and their interactions. This motivates our work to jointly cluster PPI networks, which integrates available large-scale constituent, topological, and functional information regarding proteins and their interactions to obtain biologically meaningful clustering results. We propose a new joint network clustering method for a pair of PPI networks which integrates both protein interaction and sequence similarities. We formulate network clustering as a block modeling problem by mapping the original networks to an *image graph* in which nodes represent potential functional modules with specific functionalities. The *image graph* optimally preserves the interaction patterns among nodes of the original networks across corresponding modules (Figure 1A), and enables to capture the functional interdependences between biomolecules based on the ways they interact with each other. We adopt a simulated annealing (SA) algorithm for the corresponding Potts-model [3] to solve the non-convex problem of simultaneous module identification across two networks, which has been shown to yield high quality results. Our experimental results with both synthetic and real-world PPI networks demonstrate that our new joint clustering algorithm solved by SA (JointSA) outperforms separate block modeling clustering algorithm (SingleSA).

2. METHODS

We first develop an integrated mathematical model for joint clustering of two PPI networks. The motivation is to obtain biologically meaningful results by integrating useful in-

formation and accrued knowledge from two networks across species. The essence of the integrated model is to introduce a common virtual network as the image graph for both networks. From the perspective of evolution, different species may evolve from the same ancestor and share the same functional modular structure. The virtual network, where each node represents a potential module (either densely connected or sparsely connected) with groups of proteins with similar cellular functions, provides insights into the cellular functional organization by the derived modular structure as shown in Figure 1C. We solve the module identification for two networks at the same time by mapping each network into this virtual network. By this integrated model, evolutionarily conserved molecules are more likely to possess independent but coherent functions.

2.1. Joint network clustering

Let two given PPI networks from two species be $\mathcal{G}_k = \{V_k, E_k\}$ ($k = 1, 2$), where $V_k = \{u_1^k, u_2^k, \dots, u_{N_k}^k\}$ are sets of nodes that represent proteins in \mathcal{G}_k and E_k are sets of edges representing interactions among proteins in \mathcal{G}_k . The network topology of \mathcal{G}_k can be represented by adjacency matrix A^k , where $A_{ij}^k \in \{0, 1\}$ denotes the interactions between nodes u_i^k and u_j^k in \mathcal{G}_k . Let $S_{12}(u_i^1, u_j^2)$ represent the sequence similarity between node $u_i^1 \in V_1$ and node $u_j^2 \in V_2$. We aim to find mappings from the given networks to the introduced virtual network $M = \{V_M, E_M\}$ as illustrated in Figure 1C, where $V_M = \{v_1^m, v_2^m, \dots, v_{N_m}^m\}$ denotes the virtual nodes in M and N_m is the total number of virtual nodes in M ($N_m \leq \min_{k \leq 2} \{N_k\}$). The adjacency matrix of M is A^M . For each given network \mathcal{G}_k , we define a many-to-one mapping $\Psi_k : V_k \mapsto V_M$ to the virtual network M . For a node u_i^k in \mathcal{G}_k , $\Psi_k(i)$ assigns it to a virtual node v_i^m , $1 \leq i \leq N_m$.

In order to consider the sequence similarities between the nodes across networks and the interaction similarities shared by proteins in two networks, we formulate our joint clustering objective function as follows:

$$\max_{\Psi_1, \Psi_2, A^M} \lambda U(S_{12}, \Psi_1, \Psi_2) + (1 - \lambda) Q(A^1, A^2, A^M, \Psi_1, \Psi_2), \quad (1)$$

in which λ is a weighting coefficient. Function $U(S_{12}, \Psi_1, \Psi_2)$ computes the total similarity score based on the sequence similarity between corresponding proteins assigned to the same virtual node according to Ψ_1 and Ψ_2 in two networks.

$$U(S_{12}, \Psi_1, \Psi_2) = \sum_{\substack{1 \leq j \leq N_2 \\ 1 \leq i \leq N_1}} S_{12}(u_i^1, u_j^2) \delta_{\Psi_1(i), \Psi_2(j)}, \quad (2)$$

where $\delta_{v, v'}$ is the indicator function, which equals to 1 when $v = v'$ and 0 otherwise. Function $Q(A^1, A^2, A^M, \Psi_1, \Psi_2)$ measures the conservation of interaction patterns shared by corresponding proteins assigned to the same module, for which we develop a pairwise block modeling formulation to jointly consider the mapping quality between two networks as well as the interaction patterns within both networks.

Mathematically, for each network \mathcal{G}_k , Ψ_k should minimize the mismatch between the given network \mathcal{G}_k and the introduced virtual network M [3]:

$$\min_{\Psi_k, A^M} \sum_{\substack{i \neq j \\ i, j=1, \dots, N_k}} \left[A_{ij}^k - A_{\Psi_k(i), \Psi_k(j)}^M \right] (A_{ij}^k - p_{ij}^k), \quad (3)$$

in which $\left[A_{ij}^k - A_{\Psi_k(i), \Psi_k(j)}^M \right]$ calculates the number of mismatched edges between \mathcal{G}_k and M and $(A_{ij}^k - p_{ij}^k)$ denotes the penalty for the corresponding mismatch. We set $p_{ij}^k = \frac{\sum_{i' \neq i} A_{i'j}^k \sum_{j' \neq j} A_{ij'}^k}{\sum_{i' \neq j'} A_{i'j'}^k}$ to make the total mismatch error on existing edges equal to the error on absent edges ($\sum_{i \neq j} A_{ij}^k (A_{ij}^k - p_{ij}^k) = \sum_{i \neq j} (1 - A_{ij}^k) p_{ij}^k$).

Although A^M is unknown before clustering, algebraic manipulations can lead to the absorption of optimizing A^M in the following optimization problem [3]:

$$\max_{\Psi_k} \sum_{m, n} \left| \sum_{\substack{i \neq j \\ i, j=1, \dots, N_k}} (A_{ij}^k - p_{ij}^k) \delta_{\Psi_k(i), m} \delta_{\Psi_k(j), n} \right|. \quad (4)$$

Once we derive Ψ_k [3], A^M can be estimated based on the interaction preservation in (3) in a straightforward manner. Therefore, for \mathcal{G}_1 and \mathcal{G}_2 , we have $Q(A^1, A^2, A^M, \Psi_1, \Psi_2)$ equal to

$$\sum_{m, n} \left| \sum_{k=1, 2} \sum_{i \neq j} (A_{ij}^k - p_{ij}^k) \delta_{\Psi_k(i), m} \delta_{\Psi_k(j), n} \right|. \quad (5)$$

To summarize, the final formulation for joint blockmodel clustering can be written as:

$$\begin{aligned} & \max_{\Psi_1, \Psi_2} \lambda \sum_{\substack{1 \leq j \leq N_2 \\ 1 \leq i \leq N_1}} S_{12}(u_i^1, u_j^2) \delta_{\Psi_1(i), \Psi_2(j)} \\ & + (1 - \lambda) \sum_{m, n} \left| \sum_{k=1, 2} \sum_{i \neq j} (A_{ij}^k - p_{ij}^k) \delta_{\Psi_k(i), m} \delta_{\Psi_k(j), n} \right|. \end{aligned} \quad (6)$$

2.2. Optimization for joint network clustering

With the new mathematical model for joint network clustering, we now turn to the problem of solving the optimization problem (6). To obtain high quality solutions for this highly non-convex problem, we implement a simulated annealing (SA) algorithm based on the heat-bath algorithm for Potts-Models [3].

To derive the SA algorithm, we assume that each potential virtual node v_i^m in M represents a spin state and Ψ_k assigns each original network node to an arbitrary spin state. We use $H_{v_\phi^m}$ to denote the energy of the system represented in the objective function (6) with $\Psi_k(i) = v_\phi^m$ at temperature T . We apply the single spin heat-bath update rule, which updates the system energy when making a state change for a given node u_i^k from a state v_ϕ^m to v_α^m : $H_{v_\alpha^m} = H_{v_\phi^m} + \Delta H_{\Psi_k(i): v_\phi^m \rightarrow v_\alpha^m}$.

The probability of making state assignment change is proportional to the exponential of the corresponding energy change of the entire system with all other nodes' states fixed.

$$p(\Psi_k(i) = v_\alpha^m) = \frac{\exp\left\{-\beta\Delta H_{\Psi_k(i):v_\phi^m \rightarrow v_\alpha^m}\right\}}{\sum_{n=1}^{N_m} \exp\left\{-\beta\Delta H_{\Psi_k(i):v_\phi^m \rightarrow v_n^m}\right\}}, \quad (7)$$

in which $\beta = 1/T$. In order to compute the energy change $\Delta H_{\Psi_k(i):v_\phi^m \rightarrow v_\alpha^m}$ at T , we first decompose the energy change into two terms based on (1):

$$\Delta H_{\Psi_k(i):v_\phi \rightarrow v_\alpha} = \lambda \Delta U_{\Psi_k(i):v_\phi \rightarrow v_\alpha} + (1-\lambda) \Delta Q_{\Psi_k(i):v_\phi \rightarrow v_\alpha}, \quad (8)$$

where $\Delta U_{\Psi_k(i):v_\phi^m \rightarrow v_\alpha^m}$ computes the energy change with respect to node similarities caused by switching u_i^k from the state v_ϕ^m to v_α^m . We can write $\Delta U_{\Psi_k(i):v_\phi^m \rightarrow v_\alpha^m}$ as:

$$\Delta U_{\Psi_k(i):v_\phi^m \rightarrow v_\alpha^m} = \sum_j^{N_l} S_{ki} (u_i^k, u_j^l) \left[\delta_{\Psi_l(j), v_\alpha^m} - \delta_{\Psi_l(j), v_\phi^m} \right], \quad (9)$$

Each potential state change based on (9) takes $O(N^{v_\phi^m} + N^{v_\alpha^m})$ operations where $N^{v_\phi^m}$ and $N^{v_\alpha^m}$ denote the number of nodes assigned to states v_ϕ^m and v_α^m . Thus, each local update takes $O((N_1 + N_2)(N^{v_\phi^m} + N^{v_\alpha^m}))$ operations.

The energy change with respect to network structure $\Delta Q_{\Psi_k(i):v_\phi \rightarrow v_\alpha}$ can be calculated similarly as in [3] by the following equation

$$\begin{aligned} \Delta Q_{\Psi_k(i)=v_\phi^m \rightarrow v_\alpha^m} = & \left(\left| a_{v_\phi^m - i, v_\phi^m - i}^k + a_{v_\phi^m, v_\phi^m}^l - \left| a_{v_\phi^m, v_\phi^m}^k + a_{v_\phi^m, v_\phi^m}^l \right| \right) \right. \\ & + \left(\left| a_{v_\alpha^m + i, v_\alpha^m + i}^k + a_{v_\alpha^m, v_\alpha^m}^l - \left| a_{v_\alpha^m, v_\alpha^m}^k + a_{v_\alpha^m, v_\alpha^m}^l \right| \right) \right) \\ & + 2 \left(\left| a_{v_\phi^m - i, v_\alpha^m + i}^k + a_{v_\phi^m, v_\alpha^m}^l - \left| a_{v_\phi^m, v_\alpha^m}^k + a_{v_\phi^m, v_\alpha^m}^l \right| \right) \right) \\ & + 2 \sum_{s \neq v_\phi^m, v_\alpha^m}^{N_M} \left(\left| a_{v_\phi^m - i, s}^k + a_{v_\phi^m, s}^l - \left| a_{v_\phi^m, s}^k + a_{v_\phi^m, s}^l \right| \right) \right) \\ & + 2 \sum_{s \neq v_\phi^m, v_\alpha^m}^{N_M} \left(\left| a_{v_\alpha^m + i, s}^k + a_{v_\alpha^m, s}^l - \left| a_{v_\alpha^m, s}^k + a_{v_\alpha^m, s}^l \right| \right) \right) \end{aligned}, \quad (10)$$

where $a_{r,s}^k$ is the overall mismatch penalty between modules r and s in \mathcal{G}_k . $m_{r,s}^k$ represents the total interactions between modules r and s in G_k and D_r^k is the summation of the degrees of all the nodes in module r in G_k . The subscript $v_\phi^m - i$ in (10) stands for the operation of removing the corresponding node u_i^k from the set of nodes assigned to v_ϕ^m while $v_\alpha^m + i$ denotes adding the node to v_α^m . These values in (10) can be efficiently computed by the following equations:

$$a_{r,s}^k = m_{r,s}^k - \frac{D_r^k D_s^k}{2M^k}; \quad (11)$$

$$m_{r,s}^k = \sum_{ij} A_{ij}^k \delta_{\Psi_k(i), r} \delta_{\Psi_k(j), s}; \quad (12)$$

$$D_r^k = \sum_{ij} A_{ij}^k \delta_{\Psi_k(i), r}; \quad M^k = \sum_{ij} A_{ij}^k; \quad (13)$$

$$a_{v_\phi^m - i, v_\phi^m - i}^k = m_{v_\phi^m, v_\phi^m}^k + 2d_{i \rightarrow v_\phi^m}^k - \frac{(D_{v_\phi^m}^k - d_i^k)^2}{2M^k}; \quad (14)$$

$$a_{v_\alpha^m + i, v_\alpha^m + i}^k = m_{v_\alpha^m, v_\alpha^m}^k + 2d_{i \rightarrow v_\alpha^m}^k - \frac{(D_{v_\alpha^m}^k + d_i^k)^2}{2M^k}; \quad (15)$$

$$a_{v_\phi^m - i, v_\alpha^m + i}^k = m_{v_\phi^m, v_\alpha^m}^k - d_{i \rightarrow v_\alpha^m}^k + d_{i \rightarrow v_\phi^m}^k - \frac{((D_{v_\phi^m}^k)^2 - (d_i^k)^2)}{M^k}; \quad (16)$$

$$a_{v_\phi^m - i, s}^k = m_{v_\phi^m, s}^k - d_{i \rightarrow v_\alpha^m}^k - \frac{(D_{v_\phi^m}^k - d_i^k) D_s^k}{2M^k}. \quad (17)$$

where $d_{i \rightarrow v_\phi^m}^k = \sum_j A_{ij}^k \delta_{j, v_\phi^m}$ denotes the number of interactions between node u_i^k and nodes in state v_ϕ^m and d_i^k denotes the degree of the node u_i^k . Based on the equations (10) to (17), the local update for ΔQ takes $O((N_1 + N_2)N_m^2)$ operations at each temperature.

3. EXPERIMENTS

To demonstrate that our joint network clustering algorithm (JointSA) is superior to separate network clustering algorithms, we compare our algorithm with the block modeling clustering algorithm of single networks solved by simulated annealing (SingleSA) [3] on synthetic networks and two PPI networks collected from the Database of Interacting Proteins (DIP) [10].

3.1. Synthetic networks

We test JointSA and SingleSA on a set of synthetic networks. We first generate noise-free networks based on the virtual network shown in Figure 2A. In the virtual network, virtual nodes ‘‘a’’, ‘‘b’’ and ‘‘e’’ represent densely connected modules and virtual nodes ‘‘c’’ and ‘‘d’’ represent the modules having the bi-partite structure with interactions running mainly between nodes assigned to them. We set the sizes of the modules corresponding to the virtual nodes ‘‘a’’, ‘‘b’’, ‘‘c’’, ‘‘d’’ and ‘‘e’’ to 16, 48, 32, 32 and 80 respectively. Additionally, we can add noise to networks with the noise level as the percentage of interactions that do not adhere to the topology of these virtual nodes. A similar setting has been used for benchmarking in [11, 3]. For separate network clustering, we apply SingleSA to these randomly generated synthetic networks with different noise levels. For joint network clustering, we simply use two same synthetic networks and further introduce a node similarity S_{12} between them, in which we randomly assign 8 similar pairs in average for each node. We change the difficulty of the joint clustering task by using different noise levels for both network interactions and the nodes similarities. As our JointSA also uses simulated annealing for optimization, we use the same parameters for both SingleSA and JointSA. Both SingleSA and JointSA converge to the final solutions within a few minutes.

As we know the ground truth of the structure in synthetic networks, we use normalized mutual information (NMI) [12] between the ground truth and the clustering results obtained

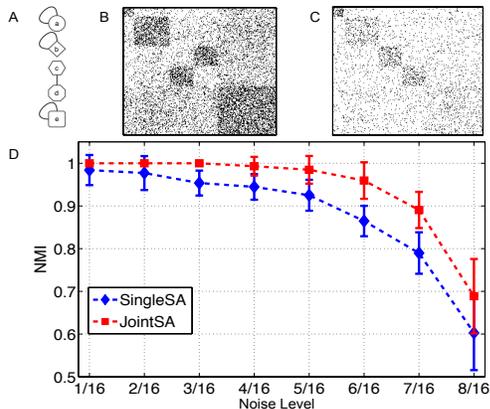


Fig. 2. Results for synthetic networks with a known underlying structure: A. The structure of the virtual network. B. One example of the adjacency matrix for a generated network at the noise level 0.5; C. One example of the similarity matrix at the noise level 0.5; D. Performance comparison.

by both algorithms to evaluate the clustering accuracy. Figure 2D shows the performance comparison between JointSA and SingleSA. At each noise level, we randomly generate 50 networks. And for each network, we take the best out of 10 runs with different random initializations. The average and the standard deviation of the NMI obtained from 50 randomly generated networks are plotted in Figure 2D. Clearly, our JointSA significantly outperforms SingleSA, which implies that the integration of the information across two network including node similarities with reasonable accuracy may significantly improve joint clustering results.

3.2. Protein interaction networks

In order to validate that joint network clustering can detect more biologically meaningful modules than the separate clustering, we further compare JointSA and SingleSA on real-world PPI networks. We use the PPI networks of *S. cerevisiae* (Sce) and *D. melanogaster* (Dme) extracted from DIP [10]. These two networks have 4,990 nodes with 21,911 edges (Sce) and 7,390 nodes with 22,695 edges (Dme) respectively. The similarities among proteins across two networks are computed by SSEARCH routine in the FASTA package [13]. The final protein similarity is binary by setting to 1 when the e-value between two protein sequences is lower than 10^{-5} , and 0 otherwise. With such large PPI networks, it takes several hours for JointSA and SingleSA to converge.

We annotate each node in PPI networks by its corresponding gene name and use Ontologizer [14] to perform Gene Ontology (GO) enrichment analysis for the results obtained by JointSA and SingleSA. GO enrichment analysis helps interpret the corresponding cellular functions for the proteins in derived modules by statistically detecting whether they correspond to a specific gene ontology category (GO term). Figure 3A shows the number of significantly enriched modules detected by both JointSA and SingleSA. From Figure 3A, we find that JointSA identifies more GO

enriched modules than SingleSA for both *S. cerevisiae* and *D. melanogaster* PPI networks for different number of modules (N_m). Figure 3B illustrates the number of enriched GO terms that cover fewer than 100 proteins for the identified modules by both JointSA and SingleSA. We observe that the modules detected by JointSA have more annotated GO terms with smaller sizes (< 100), which implies that the identified modules by JointSA are enriched with more specific cellular functionalities. Hence, JointSA can identify biologically more significant modules with known cell functions.

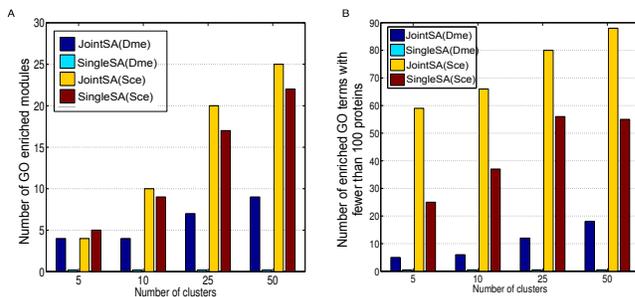


Fig. 3. A. Number of modules with statistically significantly enriched GO terms below 1% after Bonferroni correction for different N_m . B. Number of statistically significantly enriched GO terms that cover fewer than 100 proteins.

4. DISCUSSION

In this paper, we propose a novel mathematical framework for joint clustering two PPI networks simultaneously. Furthermore, our mathematical framework (1) can be extended to multiple networks in a straightforward manner with the following objective function:

$$\max_{\Psi_1, \dots, \Psi_k, A^M} \lambda \sum_{i < j} U(S_{ij}, \Psi_i, \Psi_j) + (1 - \lambda)Q(A^1, \dots, A^k, A^M, \Psi_1, \dots, \Psi_k). \quad (18)$$

We note that the corresponding local update for the SA optimization has $O(\sum_i^k N_i N_m^2)$ computational complexity, which scales linearly with respect to the number of networks k . The convergence of the corresponding simulated annealing solution can be guaranteed by setting the initial temperature high and the cooling down procedure slow [15].

Our joint clustering formulation integrates conserved similarity across networks into a flexible clustering model based on block modeling. The preliminary experimental results have shown that joint clustering outperforms clustering of individual networks separately with respect to obtaining biologically meaningful modules. Our future research will focus on developing more efficient and effective computational tools for studying large-scale biological networks simultaneously so that multiple information resources can be utilized to compensate potential errors and bias from the existing curated data sets.

5. REFERENCES

- [1] JB Pereira-Leal, AJ Enright, and CA Ouzounis, "Detection of functional modules from protein interaction networks," *Proteins*, vol. 54, no. 1, pp. 49–57, Jan 2004.
- [2] R Guimera and LAN Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, pp. 895–900, 2005.
- [3] S Pinkert, J Schultz, and J Reichardt, "Protein interaction networks: More than mere modules," *PLoS Comput Biol*, vol. 6, pp. e1000659, Jan 2010.
- [4] Y. Wang and X. Qian, "A novel subgradient-based optimization algorithm for block model functional module identification," *BMC Bioinformatics*, vol. 14(Suppl 2), no. S23, 2013.
- [5] Y. Wang and X. Qian, "Functional module identification by block modeling using simulated annealing with path relinking," in *ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2012*, Orlando, USA, 2012.
- [6] Y Wang and X Qian, "Functional module identification in protein interaction networks by interaction patterns," *Bioinformatics*, , no. btt569, 2013.
- [7] BP Kelley, R Sharan, RM Karp, T Sittler, DE Root, BR Stockwell, and T Ideker, "Conserved pathways within bacteria and yeast as revealed by global protein network alignment," *Proc Natl Acad Sci USA*, vol. 100, no. 20, pp. 11394–11399, 2003.
- [8] M Koyutürk, A Grama, and W Szpankowski, "An efficient algorithm for detecting frequent subgraphs in biological networks," *Bioinformatics*, vol. 20, pp. SI200–207, 2004.
- [9] BJ Yoon, X Qian, and SME Sahraeian, "Comparative analysis of biological networks using markov chains and hidden markov models," *IEEE Signal Processing Magazine*, vol. 29, no. 1, pp. 22–34, 2012.
- [10] L Salwinski, CS Miller, AJ Smith, FK Pettit, JU Bowie JU, and D Eisenberg, "The Database of Interacting Proteins: 2004 update," *Nucleic Acids Research*, vol. 32, pp. D449–D451, 2004.
- [11] M Newman and E Leicht, "Mixture models and exploratory data analysis in networks," *Proc Natl Acad Sci USA*, vol. 104, pp. 9564–9569, 2007.
- [12] L Danon, J Duch, A Diaz-Guilera, and A Arenas, "Comparing community structure identification," *J Stat Mech*, vol. P09008, 2005.
- [13] WR Pearson and DJ Lipman, "Improved tools for biological sequence comparison," *Proc Natl Acad Sci USA*, vol. 85, no. 8, pp. 2444–2448, 1988.
- [14] S Bauer, S Grossmann, M Vingron, and PN Robinson, "Ontologizer 2.0—a multifunctional tool for go term enrichment analysis and data exploration," *Bioinformatics*, vol. 24, no. 14, pp. 1650–1, 2008.
- [15] S Kirkpatrick, CD Gelatt Jr., and M Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.