2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

GSCA: Reconstructing biological pathway topologies using a cultural algorithms approach

Thair Judeh, Thaer Jayyousi, Lipi Acharya, Robert G. Reynolds, and Dongxiao Zhu

Abstract— **With the increasing availability of gene sets and pathway resources, novel approaches that combine both resources to reconstruct networks from gene sets are of interest. Currently, few computational approaches explore the search space of candidate networks using a parallel search. In particular, search agents employed by evolutionary computational approaches may better escape false peaks compared to previous approaches. It may also be hypothesized that gene sets may model signal transduction events, which refer to linear chains or cascades of reactions starting at the cell membrane and ending at the cell nucleus. These events may be indirectly observed as a set of unordered and overlapping gene sets. Thus, the goal is to reverse engineer the order information within each gene set to reconstruct the underlying source network using prior knowledge to limit the search space.**

We propose the Gene Set Cultural Algorithm (GSCA) to reconstruct networks from unordered gene sets. We introduce a robust heuristic based on the arborescence of a directed graph that performs well for random topological sort orderings across gene sets simulated for four *E. coli* networks and five *Insilico* networks from the DREAM3 and DREAM4 initiatives, respectively. Furthermore, GSCA performs favorably when reconstructing networks from randomly ordered gene sets for the aforementioned networks. Finally, we note that from a set of 23 gene sets discretized from a set of 300 *S. cerevisiae* expression profiles, GSCA reconstructs a network preserving most of the weak order information found in the KEGG Cell Cycle pathway, which was used as prior knowledge.

I. Introduction

Recently, a wave of publications has emerged incorporating pathway topologies into the analysis of molecular profiling data sets and their derivatives including Paradigm [1], SubpathwayMiner [2], and TEAK [3]. These approaches and others use existing pathway database resources such as Reactome [4] and KEGG [5]. Given the abundance of gene expression data sets and their derived gene sets, novel algorithms that reliably infer biological pathways topologies may be of use. Furthermore, reconstructing a biological network may be an important piece for further analysis such as network partitioning and network querying.

Previous approaches to reconstruct biological networks include Probabilistic Boolean networks [6], Bayesian networks [7], mutual inference based methods [8], and ordinary differential equations [9]. While useful, these approaches may not exploit signaling cascades as illustrated in Figure 1. In Figure 1, the underlying signaling pathway may have different components activated in response to various biological conditions. Various components may be activated through linear signaling cascade mechanisms. In one paradigm, a cell membrane receptor is bounded by a growth factor. This in turn causes a signal to be transmitted to the nucleus, which results in a change in gene expression levels [10]. In particular, linear signaling cascades may be thought of as ordered sets of genes but are observed as unordered sets of genes. Approaches that are specifically designed for gene sets may be of use.

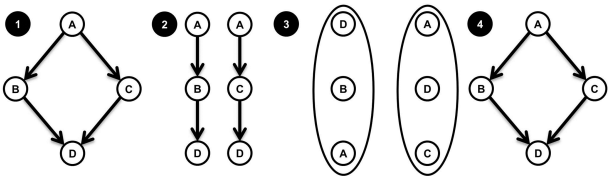

Fig. 1. 1) The underlying signaling pathway. 2) A signaling pathway may consist of several overlapping signaling transduction events that may be best represented using ordered and linear chains of genes. These signal transduction events whose orders are known are denoted as ordered gene sets. 3) The indirect observed measurements are available as unordered gene sets. 4) Using the unordered gene sets in (3), the goal is to reconstruct the underlying network found in (1). This figure originally appeared in [11].

Gene Set Enrichment Analysis [12] and Gene Set Analysis [13] are some of the many approaches currently available that focus on the analysis of gene sets, which may be obtained via databases such as the Molecular Signatures Database [12] or by discretizing time series data [14] and steady state data. Gene sets are more interpretable as they correspond to lists of biological processes [15] and may be thought of as derived sample features that succinctly summarize the original gene expression data [16]. Furthermore, by using gene sets, data sets from multiple platforms may be integrated [16]. These previous approaches, however, may focus only on gene sets individually in relation to gene expression data sets and may not necessarily focus on the interactions that various gene sets may have with one another. In particular, for a set of highly overlapping gene sets, sufficient information may be present that allows for the reconstruction of the underlying biological network that may have emitted the gene sets.

Prior knowledge may also be exploited and used to reduce the search space and to improve the reconstructed networks. In the work of Liu and Zhao [17], gene expression data was utilized to better delineate the pathway components of the *S. cerevisiae* MAPK signaling pathway found in protein-protein interaction data. In the work of Hashemikhabir et al [18], the problem of reconstructing a signaling pathway was framed as finding the minimum number of operations to modify a

Thair Judeh, Thaer Jayyousi, Robert G. Reynolds, and Dongxiao Zhu are with the Department of Computer Science at Wayne State University, Detroit, MI (email: {tjudeh, al6854, robert.reynolds, dzhu}@wayne.edu). Lipi Acharya is with Dow AgroSciences LLC, Indianapolis, IN (email: lacharya@dow.com).

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

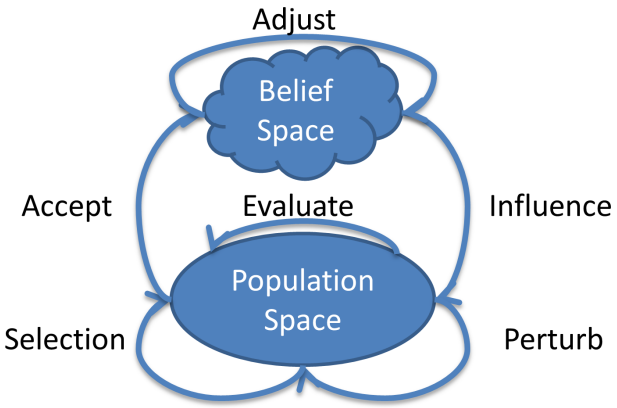

Fig. 2. Reynolds' cultural algorithm framework. Elements in the belief space influence the next generation in the population space. Elements in the population are then perturbed, and their fitness is evaluated. Some elements in the population are then selected to influence the belief space, which in turn may then be adjusted. The process repeats until algorithm termination. Adapted from [19]. This figure originally appeared in [11].

reference pathway that bests corresponded to the input RNAi data. For the Gene Set Cultural Algorithm (GSCA) (a preliminary version that was less streamlined and did not exploit prior knowledge was presented in [11] available at `http://dx.doi.org/10.1145/2506583.2506650`), prior knowledge via the KEGG pathways may be used to hierarchically order the genes using a topological sort ordering.

II. GSCA

At the heart of Gene Set Cultural Algorithm (GSCA) is Reynolds' cultural algorithm framework [20], [21]. The cultural algorithm framework is an evolutionary computational model consisting of three major components: the population space, the belief space, and the communication protocol that allows the population space to influence the belief space and vice-versa as illustrated in Figure 2. Furthermore, newer versions of the cultural algorithm framework may exploit a total of five sources of knowledge [19], [22]. The first knowledge source is situational knowledge, which is responsible for keeping track of the most fit solutions found at each generation. Normative knowledge is then used to provide guidelines and standards for individual behaviors. Domain knowledge is similar to situational knowledge except that it is not updated at the end of each generation. As such, prior knowledge may serve as domain knowledge. History knowledge maintains information about changes within the search space and may be modeled via the use of a tabu list [23]. Finally, topographical knowledge represents the population space as a multi-dimensional grid. Topographical knowledge can thus be used to guide a search towards unexplored areas. GSCA is able to use situational knowledge, domain knowledge, and history knowledge.

The overall framework of GSCA is presented in Algorithm 1. In addition to using the cultural algorithm framework, GSCA uses topological sort orderings to reconstruct a network from unordered linear gene sets. It also uses the KEGG pathways as prior knowledge to reconstruct the latent

Algorithm 1: GSCA

Input: The unordered gene sets U, the number of search agents/ beliefs B, the number of elites T, and the number of generations J.
Output: The directed acyclic graph G of the most fit belief.
Randomly initialize B beliefs of length N (the number of unique genes/nodes in U) in the belief space (Use domain knowledge if available).
Set the exploration status E of all beliefs to false.
Decompose the unordered gene sets U into a set of unique pairs R.
for $j = 1, ..., J$ **do**
  `/* Population Space */`
  for $i = 1, ..., B$ **do**
    if E_i *is true* **then**
      Continue
    end
    Let the set S be the empty set.
    Sort **U** according to a belief B_i.
    Add B_i to the set S.
    Find the fitness of B_i.
    Set the top belief B_T as B_i.
    for $k = 1, ..., R$ **do**
      Swap a pair of nodes in B_i specified by R_k to generate a new belief B_{ik}.
      if *fitness*$(B_{ik}) >$ *fitness*(B_T) **then**
        $B_T = B_{ik}$.
        Empty S.
        Add B_{ik} to S.
      else if *fitness*$(B_{ik}) =$ *fitness*(B_T) **then**
        Add B_{ik} to S.
      end
    end
    With uniform probability, randomly select B_T from S to replace B_i.
    if $B_T = B_i$ **then**
      Set E_i to $true$.
    else
      $B_i = B_T$
    end
  end
  `/* Belief Space */`
  Select the top T beliefs with best fitness values for the next generation.
  Randomly generate $B - T$ new beliefs to be added to the belief space (Use domain knowledge if available).
  Set the exploration status E of the new beliefs as false.
end
Repeat the steps of the Population Space.
Reconstruct the output graph G from the most fit belief.

networks. It should be noted that GSCA makes an additional assumption that the unordered gene sets originated from a directed acyclic graph. While this assumption may lead to loss of representative power (for example, feedback loops cannot be represented by a directed acyclic graph), it is not overly restrictive.

Briefly, a topological sort ordering is a partial linear ordering of a network's vertices or nodes such that all directed edges go from left to right [24]. Searching over topological sort orderings has been successfully applied to Bayesian networks [25] and is applicable for reconstructing networks from gene sets if the original network was a directed acyclic graph. Once the true topological sort ordering is known, reconstructing the network becomes straightforward since a topological sort ordering contains the ordering information that has been previously lost. Thus, by employing an additional assumption, the problem of reconstructing a network from unordered gene sets may now be casted as finding a topological sort ordering of the original network.

The major parameters for GSCA are the number of generations or iterations J, the number of independent search agents/ beliefs B, and the number of elite beliefs to retain T. Both J and B play a role in the algorithm's complexity whereas T helps to determine the number of random topological sort orderings to be introduced into the population each generation. It should also be noted that T plays the role of the size of the situational knowledge preserved at the end of each generation in GSCA where a smaller value of T will lead to greater exploration as $B-T$ new topological sort orderings are introduced. However, a smaller value of T may also lead to lack of exploitation of fit topological sort orderings. A balance between exploration and exploitation is sought by fixing T to be $B/2$.

III. The Belief and Population Spaces

GSCA proceeds by dividing the unordered gene sets U into a set of unique pairs R. R is bounded by $O(N(N-1)/2)$ where N is the number of unique nodes or genes in U. Via the use of R, one is able to define a neighborhood for a topological sort ordering by randomly swapping a pair of nodes in a topological sort ordering. For example, if the unordered gene sets are $\{(1,2,3,4),(2,3,4,5)\}$, then R consists of the pairs $\{(1,2),(1,3),(1,4),(2,3),(2,4),(2,5),(3,4),(3,5),(4,5)\}$. If the topological sort ordering is $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5)$ and the pair from R is $(1,2)$, the topological sort ordering is then changed to $(2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 5)$. Furthermore, by limiting the pair swaps to R, one may avoid swapping a pair of genes that are not present together within at least a single gene set. Using the example above, $(1,5)$ will be considered an invalid swap since 1 and 5 are not present together in at least one gene set.

GSCA then proceeds to initialize the belief space. The belief space is represented by B topological sort orderings, which are then transferred into B search agents whose neighborhoods are explored. In the absence of prior knowledge, the belief space is randomly initialized to B different topological sort orderings. If prior knowledge were available from pathway databases such as KEGG, any cycles or strongly connected components are first removed from the pathway. In particular, the heuristic from Query Structure Enrichment Analysis (QSEA) [26] is used. After removing any applicable cycles, the transitive closure of the prior knowledge is calculated and stored. A topological sort ordering based on prior knowledge is then constructed by iteratively selecting one of the roots of the prior knowledge's transitive closure with uniform probability. Upon selecting a root, the root and all of its edges are removed. The root is then added to the end of a topological sort ordering. The process of repeatedly selecting a root with uniform probability and removing all applicable edges is repeated until all edges are removed. By using this procedure, a topological sort ordering that obeys prior knowledge is extracted and retrieved. A simple example illustrating this procedure is illustrated in Figure 3.

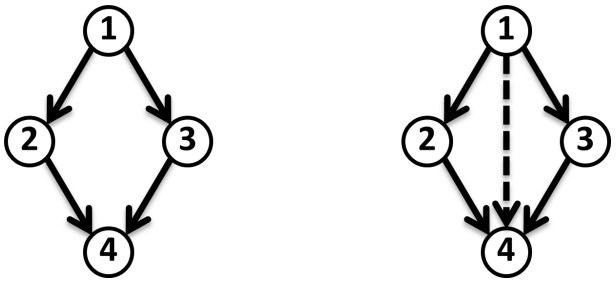

Fig. 3. An example graph (left) and its transitive closure (right). In the beginning, the only root in the transitive closure is 1. 1 is added to the start of the topological sort ordering. After removing 1 and all of its edges, the vertices 2 and 3 are roots. With uniform probability, one of them is selected to be added to the topological sort ordering. Suppose 3 was added to yield the partially constructed topological sort ordering $(1 \rightarrow 3)$. Then, 3 and all of its edges are removed. At the next step, 2 is the root, so it is added to yield $(1 \rightarrow 3 \rightarrow 2)$. After removing 2 and all of its edges, only 4 remains. After adding 4, the topological sort ordering is $(1 \rightarrow 3 \rightarrow 2 \rightarrow 4)$. The algorithm terminates as no vertices remain in the graph. It should be noted that using the aforementioned procedure, another valid topological sort ordering, $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4)$, can also be generated.

At this point, GSCA enters its population space. In the population space, each search agent/ belief B_i or topological sort ordering has its neighborhood explored by applying a unique pair from R one at a time and swapping the corresponding nodes in B_i. If a pair swap from R leads to neighboring belief that contradicts with prior knowledge, the neighboring belief is discard. To achieve this goal, the transitive closure of the prior knowledge matrix is calculated. For a neighbor of a B_i, it is first reversed. Then all weak orders in the reversed belief are checked against the transitive closure of the prior knowledge matrix. If any weak order from the reversed belief is found to exist in the transitive closure of the prior knowledge, it is determined that random belief goes against prior knowledge and is thus discarded.

For each belief B_i and its applicable neighbors, the fitness is calculated by sorting the unordered gene sets U according to each topological sort order. A transition matrix

$$M = [c_{xy}]_{N \times N} \tag{1}$$

is first reconstructed from the ordered gene sets where c_{xy} is the count of node x appearing directly before node y across all ordered gene sets. The matrix M is very similar to the transition probability matrix Π used by the GSGS and GSSA algorithms except that its rows are not normalized to sums of 1. The rationale behind this action is to preserve magnitude information found in the counts, which is otherwise lost if M is transformed into a transition probability matrix.

After reconstructing M, a heuristic based on the Chu-Liu [27] and Edmonds' algorithms [28] is used. Briefly, the Chu-Liu and Edmonds' algorithms allow one to find the maximum weighted arborescence of a directed graph. An arborescence is a graph where for a root vertex x and its descendant y, there is exactly one path from x to y. As such, an arborescence may take the form of a directed rooted tree where all edges point away from a root x. Based on the implementation used by GSCA, it also possible to generate a forest of directed trees. It should be noted that the concept of arborescences for directed graphs is analogous to the concept of spanning trees for undirected graphs. Since a reconstructed M corresponds to a directed acyclic graph, there is no need to check for cycles. The fitness score FS is calculated as

$$FS = \frac{\Sigma_{n=1}^{N}(max(M_{n\cdot}) + max(M_{\cdot n}))}{|M_E|} \quad (2)$$

where $M_{n\cdot}$ corresponds to the n^{th} row of M, $M_{\cdot n}$ corresponds to the n^{th} column of M, and $|M_E|$ corresponds to the number of edges or nonzero elements in M. As such, Equation 2 can be interpreted as calculating the sum of the arborescences of M and its transpose while dividing by the number of edges in M to favor more sparse networks.

The searching in the population space thus influences the belief space where a belief B_i or its neighbor with highest fitness score FS is promoted to the belief space B to influence the next generation. At this stage, both history and domain knowledge, if available, may be used to guide the choice of random topological sort orderings. For the history knowledge component, a tabu list is used to keep track of all beliefs or topological sort orderings last seen within a window of fixed size. The use of the tabu list thus helps to avoid visiting recently explored beliefs and in turn yields a more efficient search. Domain knowledge may be available through the use of the KEGG pathways, for example. Thus, using both history knowledge in the form of a tabu list and domain knowledge in the form of prior knowledge may better guide the search for the underlying network. The belief space B is then exited after introducing $B-T$ random topological sort orderings to avoid being trapped in local peaks.

GSCA concludes after $J-1$ generations or iterations have been reached. Since GSCA begins with the belief space, the steps for the population space are undertaken one more time. After entering the population space for the last time, the output graph G may be reconstructed using a number of ways. For the purposes of this paper, the most fit belief B_i or topological sort ordering is used to order the unordered gene sets U. After ordering U, one can simply trace the linear paths in U to add edges to reconstruct the output graph G.

TABLE I
DREAM3 AND DREAM4 NETWORK STATISTICS

Network	$\|V\|$[a]	$\|E\|$[b]	Diameter[c]	Max[d]	$\|U\|$[e]	% Used[f]
E. coli 1	27	33	4	5	125	100%
E. coli 2	30	35	3	4	34	100%
E. coli 3	48	53	4	5	141	100%
E. coli 4	42	47	3	5	114	100%
Insilico[g]1	82	107	5	7	150	41.10%
Insilico 2	93	178	6	7	150	28.90%
Insilico 3	98	173	10	17	150	3.56%
Insilico 4	97	176	9	14	150	3.02%
Insilico 5	93	171	9	11	150	5.33%

[a] the number of vertices in the network
[b] the number of edges in the network
[c] the network diameter
[d] the length of the longest gene set in the network
[e] the number of gene sets available for the network
[f] the percentage of the gene sets used for the network
[g] Feedback arcs sets were removed for all *Insilico* networks.

IV. HEURISTIC FITNESS FUNCTION JUSTIFICATION

We now justify the choice of Equation 2. To test the performance of Equation 2, four *E. coli* networks and five *Insilico* networks were extracted from GeneNetWeaver [32] corresponding to gold standard networks from DREAM3 and DREAM4 [29], [30], [31]. Furthermore, it should be noted that the heuristic for QSEA [26] was used to preprocess and remove feedback arc sets for the *Insilico* networks. After exhaustively generating all simple paths of the DREAM3 and DREAM4 gold standard networks, all gene sets of length 2 (pairs) were removed. The networks were then reconstructed from the pruned gene sets. All gene sets for the *E. coli* networks were used whereas 150 gene sets for the *Insilico* networks were randomly sampled. Some summary statistics of the reconstructed networks are displayed in Table I.

In Figure 4, $1,000,000$ random topological sort orderings were generated (with replacement), and the gene sets were ordered according to a random topological sort ordering and scored. The two score functions used include GSCA's Equation 2 as well as the log of the maximum likelihood function used by both GSSA and GSCA. For GSCA's Equation 2, only the *E. coli* 2 network had 0.2959% of random topological sort orderings dominating the true topological sort ordering whereas for all other networks, none of the scores of the random topological sort orderings dominated the scores of the true topological sort orderings. For the maximum likelihood function, on the other hand, the number of random topological sort orderings dominating the score of the true topological sort orderings were 0.7631%, 1.8777%, 0.3938%, 2.1189%, and 0.0001% for the *E. coli* 1, *E. coli* 2, *E. coli* 3, *E. coli* 4, and *Insilico* 2 networks, respectively. As such, it may be inferred that when ample or sparse gene sets are available, Equation 2 performs similarly to the maximum likelihood function.

V. SIMULATED DATA ANALYSIS

In Figure 5, the performances of GSGS and GSCA were compared. GSSA was not used since knowing the end terminals of gene sets in conjunction with GSCA's DAG assumption may be sufficient to reconstruct the underlying

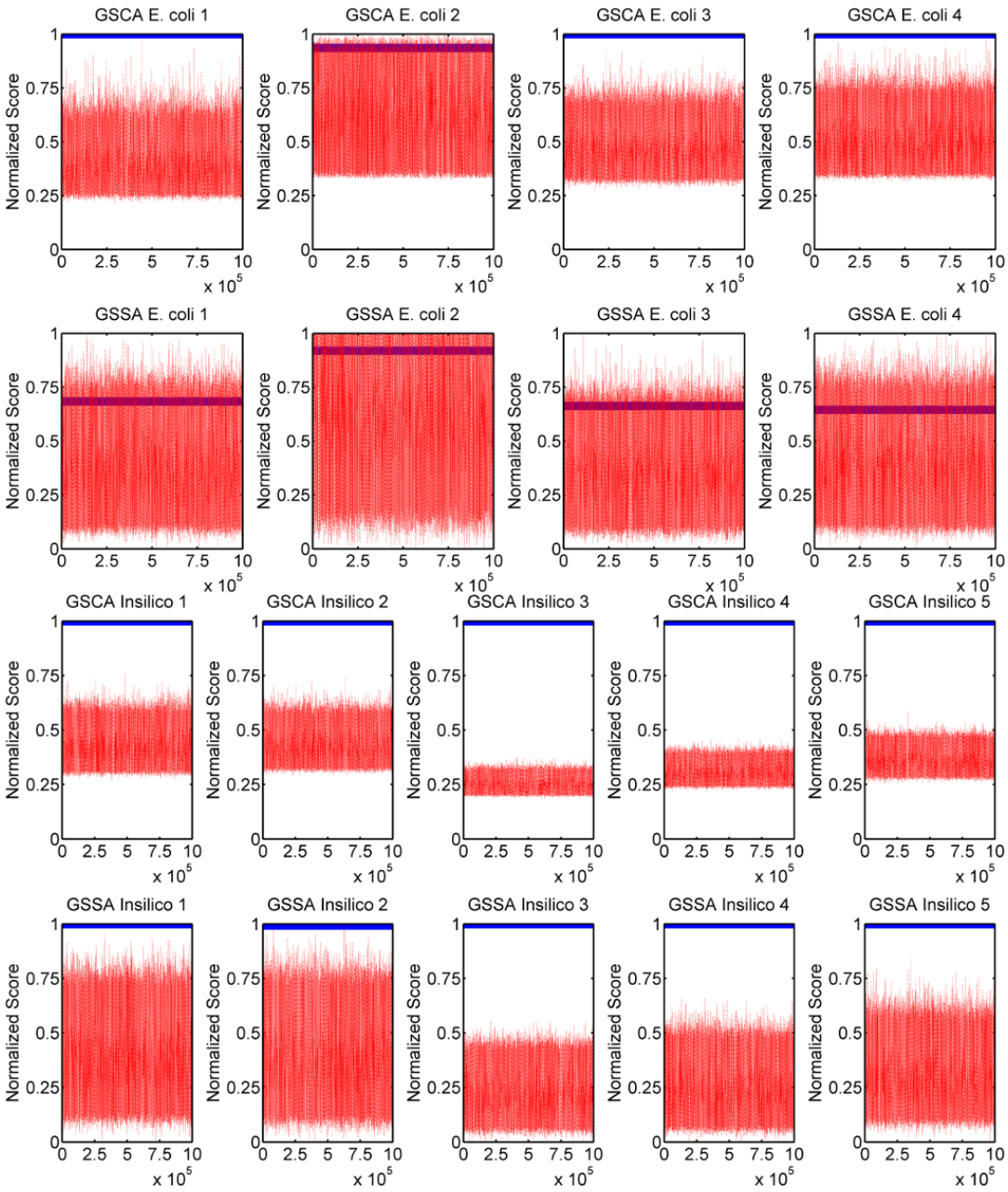

Fig. 4. GSCA's Equation 2 versus the log of the maximum likelihood function used by both GSSA and GSGS for four *E. coli* and five *Insilico* networks from the DREAM3 and DREAM4 initiatives [29], [30], [31]. $1,000,000$ random topological sort orderings were generated for all networks. After sorting the gene sets according to a given topological sort ordering, both GSCA's fitness score and the maximum likelihood score were calculated for the underlying network topologies. GSCA's scores were scaled to $(0, 1]$ by dividing by the maximum score for each network for each plot. The log of the maximum likelihood scores were scaled to $[0, 1]$ by shifting the scores by the maximal likelihood score to the right and by then dividing by the maximum score of the shifted scores for each network for each plot. The fitness of random topological sort orderings are represented as red dots whereas the fitness of the true topological sort ordering is represented by a solid blue line. Equation 2 performs similarly to the log of the maximum likelihood across all networks. Compared to the figure presented in [11], the figure presented here involves a total of nine networks versus the three networks presented earlier.

network in the presence of ample gene sets. The primary parameters for GSGS are the number of iterations for the burn-in stage and the number of samples to collect after the burn-in stage is completed. Briefly, the burn-in stage is part of the Gene Set Gibbs Sampler algorithm that discards the results of the initial iterations as the joint distribution of gene sets moves to what is hoped to be the true distribution. As for GSCA, the relevant parameters are the number of generations J, the number of beliefs/search agents B, and the number of elite beliefs to retain after each generation T.

For each network in Figure 5, GSGS and GSCA were run 10 times each on randomly ordered gene sets of sizes described in Table I. The parameters for the GSGS algorithm were $5,000$ iterations each for both the burn-in state and for sample collection for a total of $10,000$ iterations for each run. For the $5,000$ iterations of sample collection, networks were

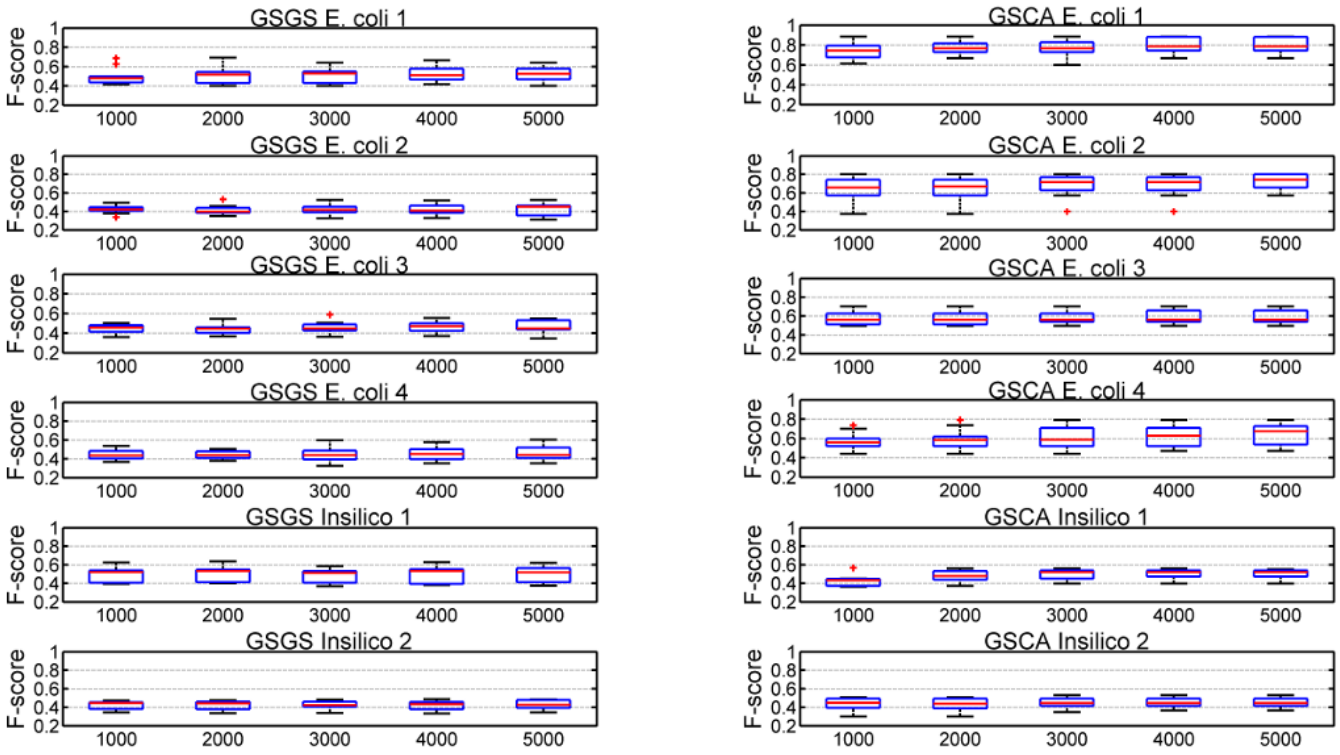

Fig. 5. A comparison of the performance of GSGS and GSCA. On the y-axis, the $F\text{-}score = 2 * Sensitivity * PPV/(Sensitivity + PPV)$ is measured. On the x-axis, snapshots of the performance of the GSGS and GSCA algorithm at varying number of iterations or generations is presented. Overall, GSCA outperforms (the *E. coli* networks) or performs similarly to the GSGS algorithm. It should be noted that a version of this figure was presented earlier in [11] consisting only of three networks.

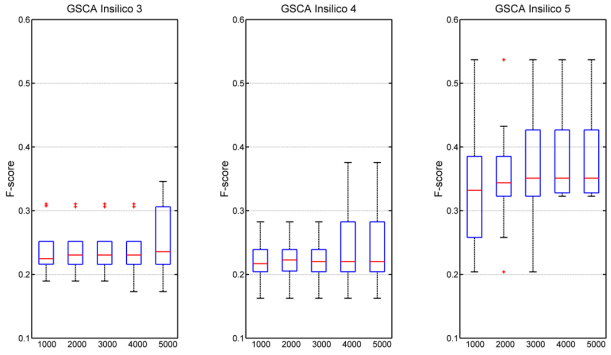

Fig. 6. Additional results for GSCA for three *Insilico* networks. GSGS was unable to run on a workstation with 4 GB of RAM due to lacking memory.

reconstructed after the collection of $1,000$, $2,000$, $3,000$, $4,000$, and $5,000$ samples. For GSCA, it was run for a total of $5,000$ generations or iterations for each run. The number of search agents/ beliefs B was set to 10, and the number of elite solutions T preserved after each generation was set to 5. The size of the tabu list was set to $100,000$ beliefs. Similar to GSGS, $5,000$ generations were run, and the $F\text{-}score = 2 * Sensitivity * PPV/(Sensitivity + PPV)$ for the most fit belief was calculated after $1,000$, $2,000$, $3,000$, $4,000$, and $5,000$ generations. *Sensitivity* is calculated as the number of true positives, i.e., the number of predicted edges agreeing with true edges, divided by the total number of true edges. *PPV* or the *Positive Predictive Value* is the number of true positives divided by the total number of predicted edges. In particular, additional iterations for sample collection do not lead to vastly improved results for the GSGS algorithm as illustrated in Figure 5. For the GSCA plots, a "learning curve" may be observed for the E. coli 1, 2, and 4 networks. As seen in Figure 5, GSCA outperforms GSGS across all four *E. coli* networks and performs similarly for two *Insilico* networks. In addition, for three *Insilico* networks, results were presented only for GSCA in Figure 6 as the memory requirements for GSGS were cost prohibitive for a workstation with 4 GB of RAM. Finally, the use of prior knowledge for the DREAM3 and DREAM4 networks may be seen in Figure 7.

VI. Real Data Analysis

For real data analysis, the well-studied compendium of 5,350 genes and 300 expression profiles corresponding to diverse mutations and chemical treatments in the budding yeast *S. cerevisiae* [33] was used. Using the MTBA toolbox [34], the Cheng and Church algorithm [35] was used on the non-log scaled fold change data matrix to produce three biclusters. In particular, the bicluster consisting of 4,826 genes and 274 samples was selected for further analysis. Prior knowledge corresponding to the largest weakly connected component of the KEGG Cell Cycle pathway was used. Genes present in the weakly connected component were dis-

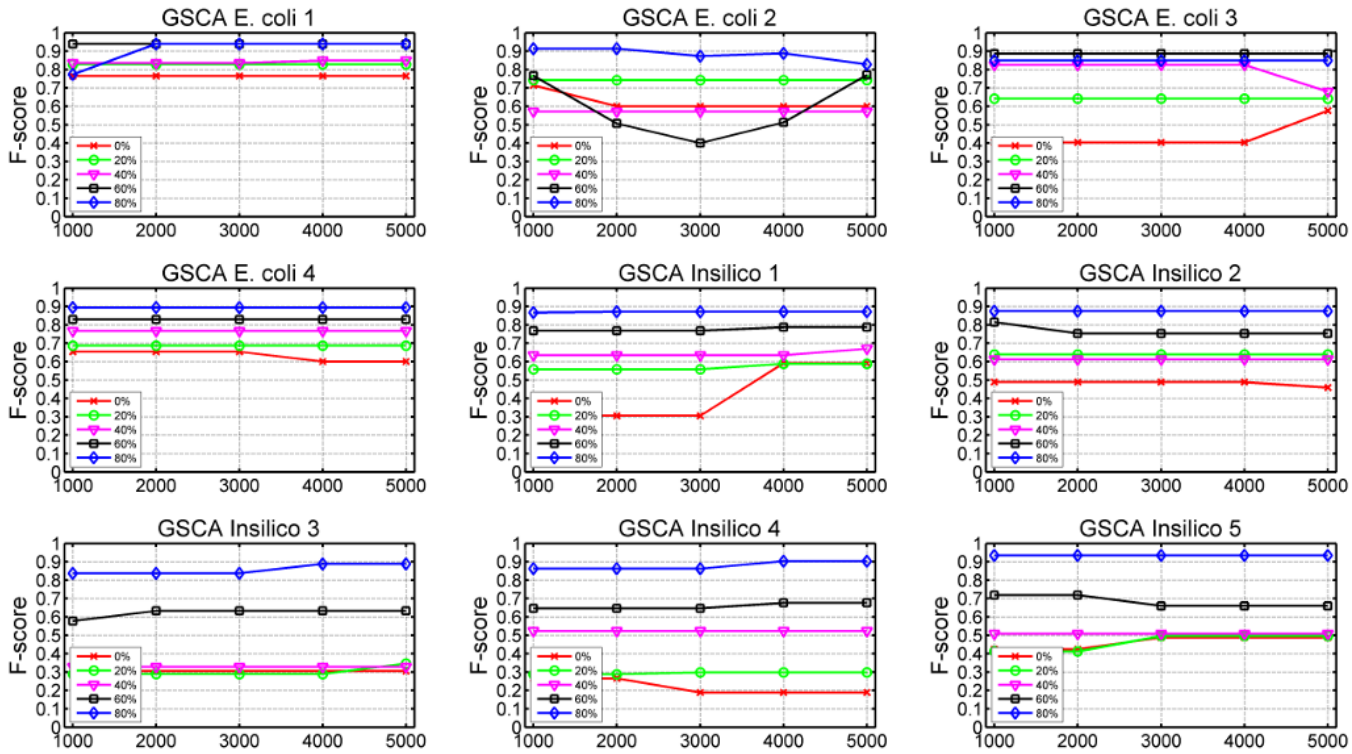

Fig. 7. The use of prior knowledge for the nine DREAM3 and DREAM4 networks. Prior knowledge was obtained by randomly sampling the specified percentage of edges from the networks presented in Table I. On the y-axis, the $F\text{-}score = 2 * Sensitivity * PPV/(Sensitivity + PPV)$ is measured. On the x-axis, snapshots of the performance of the GSCA algorithm at varying number of generations is presented. The lines in red represent no prior knowledge. The lines in green represent 20% prior knowledge. The lines in pink represent 40% prior knowledge. The lines in black represent 60% prior knowledge, and the lines in blue represent 80% prior knowledge. As can be seen overall, prior knowledge leads to an overall better performance for GSCA.

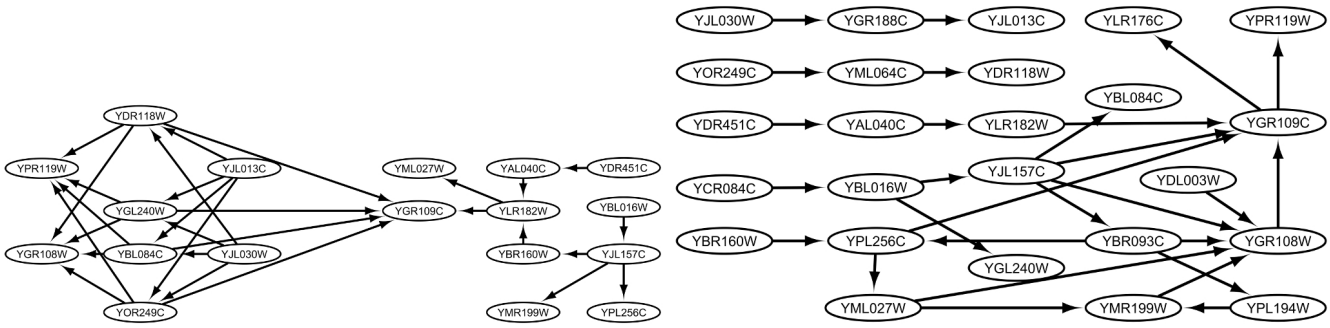

Fig. 8. **Left:** From KEGG the following *S. cerevisiae* Cell Cycle Pathway was used as prior knowledge. **Right:** The network reconstructed by GSCA using the prior knowledge and the 23 out of the 300 *S. cerevisiae* samples [33] consisting of 25 genes. GSCA preserves 17 weak order pairs extracted from the prior knowledge in its reconstructed network. Namely, it preserves the following: YDR451C to YAL040C, YAL040C to YGR109C, YBL016W to YGR109C, YBR160W to YGR109C, YDR451C to YGR109C, YJL157C to YGR109C, YLR182W to YGR109C, YBL016W to YJL157C, YAL040C to YLR182W, YDR451C to YLR182W, YBL016W to YML027W, YBR160W to YML027W, YJL157C to YML027W, YBL016W to YMR199W, YJL157C to YMR199W, YBL016W to YPL256C, and YJL157C to YPL256C.

cretized as 1 if the absolute value of their $\log_{10}$ fold change ratios were greater than or equal to $\log_{10}(2)$ and 0 otherwise. After converting the discretized data into gene sets, 23 gene sets with lengths ranging from 2 to 6 were extracted and in conjunction with the prior knowledge present in the KEGG Cell Cycle weakly connected component, GSCA was run for 50,000 iterations. As seen in Figure 8, GSCA preserves most of the weak order information found in the prior knowledge.

VII. Conclusion

In this paper we presented GSCA for reconstructing networks from unordered sets of genes. The primary focus of GSCA is to search the space of topological sort orderings that may represent the underlying network from which the gene sets may have originated. We presented a simulation study of the performance of the heuristic used as the fitness function algorithm for nine DREAM3 and DREAM4 networks. We also presented simulation studies for the performance of GSCA across nine simulated sets of gene sets for the afore-

mentioned networks from the DREAM initiatives. Finally, we presented a case study involving the use of 300 gene expression profiles. The network reconstructed using GSCA preserved most of the weak order information found in the KEGG network utilized as prior knowledge.

The approach presented here is useful since it robustly incorporates and exploits prior knowledge. Furthermore, each search agent/belief acts independently of one another in the search space allowing for a rather straightforward extension to threaded programming. The results produced by GSCA may also be thought of a set of weak orders. From this angle, the output of GSCA may then be used by other algorithms, such as the Bayesian based K2 algorithm, that rely upon a robust starting point to produce good results. As such, future hybrid algorithms may examine the data both from the aspects of gene sets (column view of the data) as well as the individual genes (row view of the data). Furthermore, future work may consist of examining in detail the relationships between the number of generations J, the number of beliefs B, and the number of elite beliefs T in hopes of finding an automated method of tuning the parameters based on the data set being used.

VIII. Acknowledgments

We thank Dr. Anuj Kumar and the reviewers for their feedback.

References

[1] C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler, and J. M. Stuart, "Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm," *Bioinformatics*, vol. 26, no. 12, pp. i237–45, 2010.

[2] C. Li, X. Li, Y. Miao, Q. Wang, W. Jiang, C. Xu, J. Li, J. Han, F. Zhang, B. Gong, and L. Xu, "Subpathwayminer: a software package for flexible identification of pathways," *Nucleic Acids Res*, vol. 37, no. 19, p. e131, 2009.

[3] T. Judeh, C. Johnson, A. Kumar, and D. Zhu, "Teak: topology enrichment analysis framework for detecting activated biological subpathways," *Nucleic Acids Res*, vol. 41, no. 3, pp. 1425–37, 2013.

[4] D. Croft, G. O'Kelly, G. Wu, R. Haw, M. Gillespie, L. Matthews, M. Caudy, P. Garapati, G. Gopinath, B. Jassal, S. Jupe, I. Kalatskaya, S. Mahajan, B. May, N. Ndegwa, E. Schmidt, V. Shamovsky, C. Yung, E. Birney, H. Hermjakob, P. D'Eustachio, and L. Stein, "Reactome: a database of reactions, pathways and biological processes," *Nucleic Acids Res*, vol. 39, no. Database issue, pp. D691–7, 2011.

[5] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe, "Kegg for integration and interpretation of large-scale molecular data sets," *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D109–14, 2012.

[6] L. Kaderali, E. Dazert, U. Zeuge, M. Frese, and R. Bartenschlager, "Reconstructing signaling pathways from rnai data using probabilistic boolean threshold networks," *Bioinformatics*, vol. 25, no. 17, pp. 2229–35, 2009.

[7] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman, "Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data," *Nat Genet*, vol. 34, no. 2, pp. 166–76, 2003.

[8] P. Zoppoli, S. Morganella, and M. Ceccarelli, "Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach," *BMC Bioinformatics*, vol. 11, no. 1, p. 154, 2010.

[9] M. Bansal, G. Della Gatta, and D. di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics*, vol. 22, no. 7, pp. 815–22, 2006.

[10] S. Li, "Mechanisms of cellular signal transduction," *Int J Biol Sci*, vol. 1, no. 4, p. 152, 2005.

[11] T. Judeh, T. Jayyousi, L. Acharya, R. G. Reynolds, and D. Zhu, "Gene set cultural algorithm: A cultural algorithm approach to reconstruct networks from gene sets." ACM, 2013, Conference Paper, pp. 641–648.

[12] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proc Natl Acad Sci U S A*, vol. 102, no. 43, pp. 15 545–50, 2005.

[13] B. Efron and R. Tibshirani, "On testing the significance of sets of genes," *The Annals of Applied Statistics*, pp. 107–129, 2007.

[14] Y. Li, L. Liu, X. Bai, H. Cai, W. Ji, D. Guo, and Y. Zhu, "Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks," *BMC Bioinformatics*, vol. 11, no. 1, p. 520, 2010.

[15] J. Klema, M. Holec, F. Zelezny, and J. Tolar, *Comparative evaluation of set-level techniques in microarray classification*. Springer, 2011, pp. 274–285.

[16] M. Holec, F. elezn, J. Klma, and J. Tolar, *Integrating multiple-platform expression data through gene set features*. Springer, 2009, pp. 5–17.

[17] Y. Liu and H. Zhao, "A computational approach for ordering signal transduction pathway components from genomics and proteomics data," *BMC Bioinformatics*, vol. 5, no. 1, p. 158, 2004.

[18] S. Hashemikhabir, E. S. Ayaz, Y. Kavurucu, T. Can, and T. Kahveci, "Large-scale signaling network reconstruction," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 9, no. 6, pp. 1696–708, 2012.

[19] A. P. Engelbrecht, *Computational intelligence: an introduction*. Wiley. com, 2007.

[20] R. G. Reynolds, *An adaptive computer model of the evolution of agriculture for hunter-gatherers in the valley of Oaxaca, Mexico*, 1979.

[21] ——, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. World Scientific, 1994, Conference Proceedings, pp. 131–139.

[22] R. G. Reynolds and Y. A. Gawasmeh, "Evolving heterogeneous social fabrics for the solution of real valued optimization problems using cultural algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 2012, Conference Proceedings, pp. 1–8.

[23] F. Glover, E. Taillard, and D. de Werra, "A user's guide to tabu search," *Annals of Operations Research*, vol. 41, no. 1-4, pp. 3–28, 1993.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.

[25] M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning bayesian networks," *arXiv preprint arXiv:1207.1429*, 2012.

[26] T. Judeh, T. Nguyen, and D. Zhu, "Qsea for fuzzy subgraph querying of kegg pathways." ACM, 2012, Conference Paper, pp. 474–481.

[27] Y.-J. Chu and T.-H. Liu, "On the shortest arborescence of a directed graph," *Science Sinica*, vol. 14, no. 1396-1400, p. 270, 1965.

[28] J. Edmonds, "Optimum branchings," *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, vol. 71B, no. 4, p. 233, 1967.

[29] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating realistic in silico gene networks for performance assessment of reverse engineering methods," *J Comput Biol*, vol. 16, no. 2, pp. 229–39, 2009.

[30] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference," *Proc Natl Acad Sci U S A*, vol. 107, no. 14, pp. 6286–91, 2010.

[31] R. J. Prill, D. Marbach, J. Saez-Rodriguez, P. K. Sorger, L. G. Alexopoulos, X. Xue, N. D. Clarke, G. Altan-Bonnet, and G. Stolovitzky, "Towards a rigorous assessment of systems biology models: the dream3 challenges," *PLoS One*, vol. 5, no. 2, p. e9202, 2010.

[32] T. Schaffter, D. Marbach, and D. Floreano, "Genenetweaver: in silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–70, 2011.

[33] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend, "Functional discovery via a compendium of expression profiles," *Cell*, vol. 102, no. 1, pp. 109–26, 2000.

[34] N. K. V. Jayesh Kumar Gupta, Sumanik Singh, "Matlab Toolbox for Biclustering Analysis (MTBA)," http://home.iitk.ac.in/iil/mtba/.

[35] Y. Cheng and G. M. Church, "Biclustering of expression data," pp. 93–103, 2000.