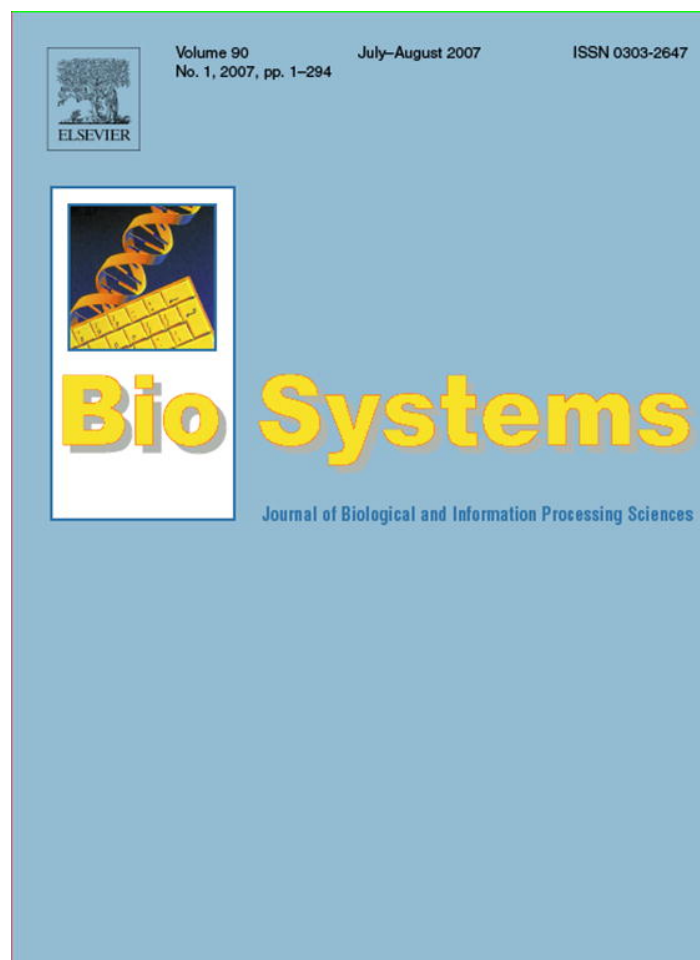


Provided for non-commercial research and educational use only.  
Not for reproduction or distribution or commercial use.



This article was originally published in a journal published by Elsevier, and the attached copy is provided by Elsevier for the author's benefit and for the benefit of the author's institution, for non-commercial research and educational use including without limitation use in instruction at your institution, sending it to specific colleagues that you know, and providing a copy to your institution's administrator.

All other uses, reproduction and distribution, including without limitation commercial reprints, selling or licensing copies or access, or posting on open internet sites, your personal or institution's website or repository, are prohibited. For exceptions, permission may be sought for such use through Elsevier's permissions site at:

<http://www.elsevier.com/locate/permissionusematerial>



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

 ScienceDirect

BioSystems 90 (2007) 197–210



[www.elsevier.com/locate/biosystems](http://www.elsevier.com/locate/biosystems)

# H-CORE: Enabling genome-scale Bayesian analysis of biological systems without prior knowledge

Sungwon Jung<sup>a</sup>, Kwang H. Lee<sup>b</sup>, Doheon Lee<sup>c\*</sup>

<sup>a</sup> Department of Electrical Engineering and Computer Science, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

<sup>b</sup> Department of BioSystems, AITrc, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

<sup>c</sup> Department of BioSystems, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

Received 6 March 2006; received in revised form 12 August 2006; accepted 16 August 2006

## Abstract

The Bayesian network is a popular tool for describing relationships between data entities by representing probabilistic (in)dependencies with a directed acyclic graph (DAG) structure. Relationships have been inferred between biological entities using the Bayesian network model with high-throughput data from biological systems in diverse fields. However, the scalability of those approaches is seriously restricted because of the huge search space for finding an optimal DAG structure in the process of Bayesian network learning. For this reason, most previous approaches limit the number of target entities or use additional knowledge to restrict the search space. In this paper, we use the *hierarchical clustering and order restriction* (H-CORE) method for the learning of large Bayesian networks by clustering entities and restricting edge directions between those clusters, with the aim of overcoming the scalability problem and thus making it possible to perform genome-scale Bayesian network analysis without additional biological knowledge. We use simulations to show that H-CORE is much faster than the widely used *sparse candidate* method, whilst being of comparable quality. We have also applied H-CORE to retrieving gene-to-gene relationships in a biological system (The ‘Rosetta compendium’). By evaluating learned information through literature mining, we demonstrate that H-CORE enables the genome-scale Bayesian analysis of biological systems without any prior knowledge.

© 2006 Elsevier Ireland Ltd. All rights reserved.

**Keywords:** Bayesian analysis; Heuristic search; Large system; Gene relationships

## 1. Introduction

Many approaches have been used to infer system architectures from observed biological phenomena. The system architectures have often been described with network-shaped models by incorporating biological entities as nodes and relations between those entities as edges (de Jong, 2002). Modelling these relationships requires information about the activities of the entities involved,

for which many types of technology are applied. For example, microarray technology can show the degree of gene activities in biological systems by representing the expression levels of genes with color and intensity information.

The Bayesian network model (Neapolitan, 2004) is widely used to describe relationships between biological entities because it has a solid mathematical basis and is able to describe complex stochastic processes. A Bayesian network  $B$  is described as  $(G, P)$ , where  $G = \langle V, E \rangle$  is a directed acyclic graph (DAG) with a set of nodes  $V$  and a set of edges  $E$ .  $G$  represents probabilistic (in)dependencies between nodes and  $P$  is the joint probability distribution of the random variables that

\* Corresponding author. Tel.: +82 428694316; fax: +82 428698680.

E-mail addresses: [swjung@biosoft.kaist.ac.kr](mailto:swjung@biosoft.kaist.ac.kr) (Sungwon Jung); [khlee@biosoft.kaist.ac.kr](mailto:khlee@biosoft.kaist.ac.kr) (Kwang H. Lee); [dhlee@biosoft.kaist.ac.kr](mailto:dhlee@biosoft.kaist.ac.kr) (Doheon Lee).

correspond to the nodes in  $G$ . When applying Bayesian networks to biological entities, each entity corresponds to a single random variable (one node in  $G$ ) and each data element of an entity is used as one sample instance from the probability distribution in the biological system. Thus inferring relationships between biological entities with observed data using the Bayesian network model involves the Bayesian network learning from a given set of observed activity data.

The learning required by Bayesian networks is a challenging problem because of the large associated search space (Chickering, 1996), which has resulted in Bayesian networks involving only tens of nodes in many applications (Neapolitan, 2004), in contrast to many biological applications requiring thousands of entities. To apply the Bayesian network model to large problems such as gene-expression analysis, Friedman et al. (1999) proposed a method to restrict the local network structure so as to reduce the DAG search space. Even though Friedman et al. (1999) was successful in scaling up the Bayesian network learning, it remains limited to problems involving hundreds of entities (Friedman et al., 2000). Other approaches (e.g. Hwang et al., 2002) that restrict local network structures also consider only hundreds of nodes. One of the main reasons for this limited scalability is the large number of candidate Bayesian network structures, which makes it too costly to compute the optimal (or near-optimal) structure. There can be other reasons of the limited scalability like the domain sizes of random variables but they become significant especially when the number of candidate network structures is large.

Learning large networks enables us to find novel knowledge between entities which could not be considered in the study of small selected entities. In this perspective, recent approaches used to scale up Bayesian network learning to the level of thousands of nodes for biological applications include that of Lee and Lee (2005), who used biological annotation information to build modules of biological entities and performed the Bayesian network learning in each module independently. Peña et al. (2005) proposed a method to grow Bayesian networks from given seed nodes to local networks with a given maximum radius. Even though these methods have been applied successfully to thousands of nodes for biological applications, they have weaknesses. For example, the method proposed by Lee and Lee (2005) needs biological annotation information a priori, and that proposed by Peña et al. (2005) has difficulty in coping with a large number of nodes when only a small amount of observed data is available. This will be mentioned further in Section 2.2.

The aim of this study was to apply Bayesian network analysis to thousands of biological entities without requiring any knowledge in addition to the observed data (although the availability of such additional knowledge could be used to refine the results further). To achieve this goal, we applied a fast learning method to large Bayesian networks by clustering nodes hierarchically and restricting edge directions between those clusters.

This paper is organized as follows. In Section 2, we describe conventional Bayesian network learning and previous approaches for scaling it upward. Section 3 presents our method for scaling up Bayesian network learning to thousands of entities, and shows the result of benchmark evaluations to illustrate its benefits. Section 4 presents experimental results obtained when applying our method to biological data, with the results and further issues being discussed in Section 5.

## 2. Bayesian network learning

### 2.1. Conventional Bayesian network learning

The learning for a Bayesian network  $B = \langle G, P \rangle$  using given observed data  $D$  involves two steps: learning the graph structure  $G$  and learning the probability distribution  $P$ . There are two approaches to structure learning (which is the more important step): scoring-based and constraint-based. The scoring-based approach involves finding a  $G$  such that  $\text{Score}(G; D)$  is maximal for a given scoring measure  $\text{Score}$ . One of the widely used scoring measures is the Bayesian score (Heckerman et al., 1995), an important characteristic of which is its *decomposability*. If a scoring measure  $\text{Score}$  satisfies decomposability,  $\text{Score}(G; D)$  can be evaluated by summing the local score for each node  $g_i$  with its set of parents  $P_G(g_i)$ :

$$\text{Score}(G; D) = \sum_i \text{Score}(g_i | P_G(g_i); D) \quad (1)$$

This decomposability gives an important benefit to the learning process. A local search procedure involving changing the status of a single edge connection allows the gain in the score to be evaluated efficiently, since the score change involves only the two nodes connected by the edge. This characteristic is useful especially when using approximate search procedures such as greedy hill climbing.

In the constraint-based approach, conditional independency test based algorithms are used to determine the presence of  $\text{Edge}(g_i, g_j)$  (an edge from  $g_i$  to  $g_j$ ) by applying statistical analyses such as the  $\chi^2$ -test between the two variables. However, we focus on the scoring-based approach here because the models of near optimal

structures can be averaged to produce a final result when there are few observed data available—the constraint-based approach lacks this ability (Neapolitan, 2004).

Scoring-based Bayesian structure learning involves finding the optimal  $G$  for which the  $\text{Score}(G;D)$  is maximal from among all possible graph structures. However, it is infeasible to find an optimal solution even for small problems (e.g. tens of nodes in a network) because the number of possible graph structures increases super-exponentially with the number of nodes (Robinson, 1973). When using the Bayesian scoring measure (Heckerman et al., 1995), it is known that Bayesian structure learning is an NP-complete problem (Chickering, 1996). For this reason, a popular way for a Bayesian structure to find a near-optimal graph structure is using heuristic approaches such as greedy hill climbing (Neapolitan, 2004).

## 2.2. Large-scale Bayesian network learning

Since the conventional approximate search methods cannot cope with problems with more than hundreds of nodes, several heuristic approaches have been proposed to restrict the DAG search space. Friedman et al. (1999) proposed the *sparse candidate* (SC) method, which determines  $CP(g_i)$  – the set of candidate parents – of size  $k$  ( $<n$ , where  $n$  is the total number of nodes) for each node  $g_i$  before the DAG search. The DAG search in the SC method involves finding a  $G$  with a high score such that  $P_G(g_i) \subseteq CP(g_i), \forall g_i \in U$  (where  $U$  is the set of all  $n$  nodes), is satisfied for the  $G$ . Hwang et al. (2002) proposed a method to restrict the local neighbor structure around each node using a Markov blanket in a Bayesian network, which is the set of neighbor nodes composed of parents, children and children's parents of a node. Brown et al. (2004) proposed the *max-min hill climbing* method, which restricts the candidate parents and children at each node. All of these methods use heuristics to restrict the local structure around each node. Even though restricting the local structure is successful in the learning for Bayesian networks with hundreds of nodes, it is difficult to apply such methods to problems involving thousands of nodes such as gene-expression analysis, because a combinatorial search for an optimal (or near-optimal) DAG structure is still required for all  $n$  nodes.

For the Bayesian analysis of thousands of biological entities, Lee and Lee (2005) proposed a method that groups entities into several overlapping clusters and performs Bayesian network learning in each cluster independently. All the local Bayesian networks are then merged into a single global Bayesian network by overlapping the identical nodes in different clusters.

However, this method requires functional annotation information additional to the observed data to group biological entities into functionally related groups. Because no result has been given without those additional information, it is unclear whether the approach of Lee and Lee (2005) can be applied to the general Bayesian network learning or not. Peña et al. (2005) proposed a constraint-based Bayesian network learning method that grows network models from given seed nodes. However, this method experiences difficulty when only a small amount of data is available, because it is difficult to apply conventional model averaging techniques to constraint-based approaches and thus it does not have a proper way to handle large problems with limited data. The method of Peña et al. (2005) is therefore considered to be more appropriate for analyzing the local area around specific seed biological entities of interest than for analyzing the entire network area especially when only a small amount of data is available.

## 3. H-CORE (hierarchical clustering and order restriction) method

### 3.1. Method outline

Our aim was to apply Bayesian network analysis to biological systems containing thousands of entities using observed data but not prior knowledge. In the Bayesian structure learning perspective, we denote this problem to find an optimal DAG structure  $G_U$  in the target Bayesian network with  $U$ , which is a set of  $n$  entities that correspond to the nodes in  $G_U$ . Our approach involves restricting the DAG search space much more than the previous approaches (Brown et al., 2004; Friedman et al., 1999; Hwang et al., 2002) that have determined candidate local structures for each node before conducting the DAG search process. To achieve this goal, we restrict the search space by restricting the global structure and thus letting the combinatorial search of considering cycles which occur for at most  $c_{\max}(<n)$  entities. We cluster entities hierarchically with cluster size of at most  $c_{\max}(<n)$  and restrict the edge directions between entities in different clusters into one direction to disallow cycles between clusters. The search space for DAG structures is restricted significantly with this approach because the consideration of cycles in the DAG search process is needed only in each cluster of size at most  $c_{\max}$ . This is much more restricted search space compared to that of DAG search considering cycles for entire  $n$  nodes.

Our approach of Bayesian network learning takes following two steps with this structure restriction method:

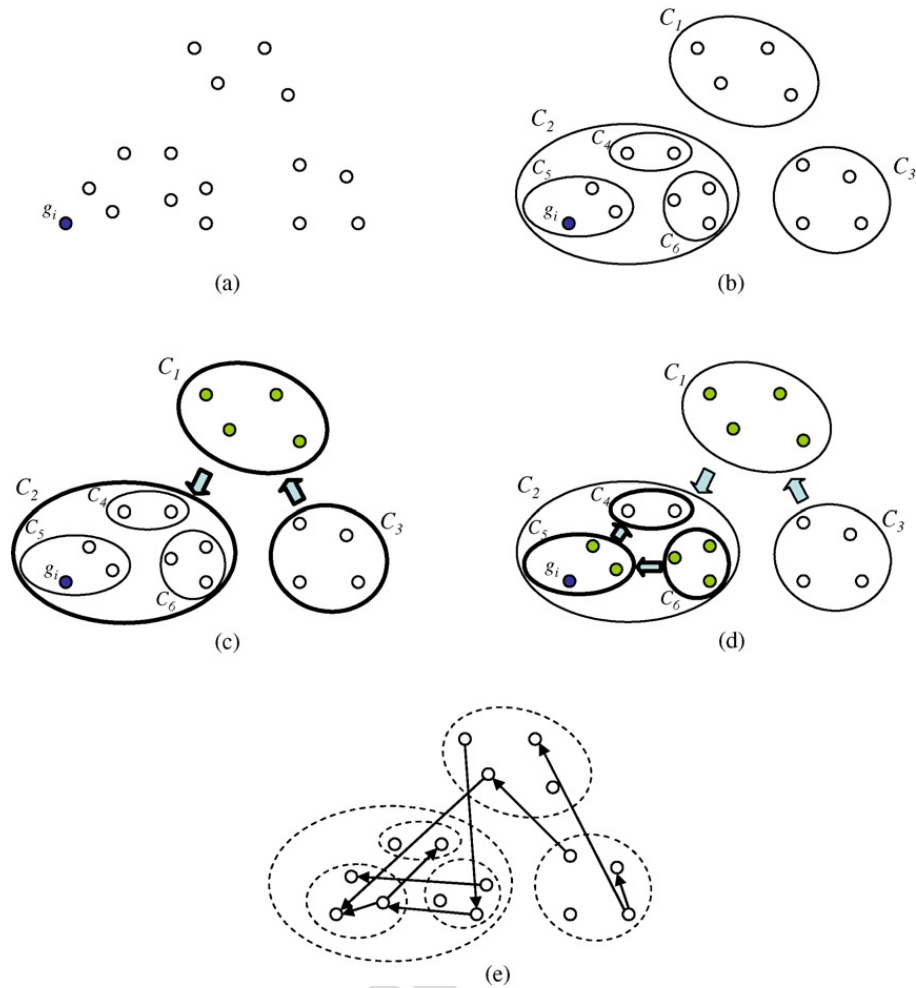


Fig. 1. The outline of the proposed method with an example of  $U$ , which involves 16 entities. (a) Given 16 entities including  $g_i$  for learning a Bayesian network. (b) The result of applying H-Cluster with  $c_{\max} = 4$ . The root cluster is  $C_0 = \{C_1, C_2, C_3\}$ , but not shown here. (c) The result of applying C-DAGSearch ( $C_0, D$ ). For  $g_i$ , the elements of  $CP(g_i)$  are represented with green color. (d) The result of applying C-DAGSearch ( $C_2, D$ ). For  $g_i$ , the  $CP(g_i)$  is updated. (e) A Bayesian network  $G_U$  that satisfies the restriction on candidate parents. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

## 1. Structure restriction step:

- (1) Hierarchically cluster entities with cluster size of at most  $c_{\max} (< n)$ .
- (2) Restrict edge directions between entities in different clusters.

## 2. Structure search step:

- (1) Find a DAG that best fits the given data while satisfying the restriction on edge directions determined in the previous step.

In the structure restriction step, we first build the hierarchy of clusters for entities such that each cluster has a predetermined maximum size  $c_{\max}$  (Algorithm 1. H-Cluster; see the example in Fig. 1(a) and (b)). In H-Cluster, clusters  $C_i$  and  $C_j$  with maximum Proximity are merged into one cluster, where the Proximity between two clusters is obtained from the mutual information

$I_D(g_k; g_l)^1$  measured from  $D$  between two entities  $g_k$  ( $\in C_i$ ) and  $g_l$  ( $\in C_j$ ), such that each entity is included in the corresponding clusters  $C_i$  and  $C_j$ . Note that the notation ' $g \in C$ ' is used for the case that 'an entity  $g$  is included in a cluster  $C$ ' while the notation ' $C_{\text{low}} \in C$ ' will be used for the case that 'a cluster  $C_{\text{low}}$  is a child of a cluster  $C$  in the hierarchy'. From this perspective, we use the notation  $|C|_{\text{ent}}$  to indicate 'the number of entities in  $C$ ' while  $|C|$  indicates 'the number of children clusters of  $C$ ' in the cluster hierarchy. See Fig. 1 as an example, where  $C_2 = \{C_4, C_5, C_6\}$  is shown. Then all of follow-

<sup>1</sup> We use conventional mutual information evaluation method from discretized data. By the way, it is known that mutual information is biased towards higher values when data is strictly discretized and there is an improved method for mutual information for continuous data (Daub et al., 2004).



ing descriptions are true:  $g_i \in C_5$ ,  $g_i \in C_2$ ,  $C_5 \in C_2$ ,  $|C_2| = 3$ ,  $|C_5|_{\text{ent}} = 3$  and  $|C_2|_{\text{ent}} = 8$ . An edge which has nodes in different clusters can be lost in the structure search step because edge direction will be restricted between clusters. Because we should minimize the number of edges between clusters, which can be lost in the structure search step, we define Proximity as the averaged mutual information between two clusters, which represents the expected denseness of edges between clusters, by considering mutual information as an expectation value of an edge:

---

**Algorithm 1.** H-Cluster( $U, D, c_{\max}$ )
 

---

```

1: C := ∅
2:
3:
4: for all  $g_i$  such that  $g_i \in U$  do
5:    $C_i := \{g_i\}$ 
6:   C := C ∪ { $C_i$ }
7: end for
8: repeat
9:    $\{C_l, C_m\} := \text{argmax}_{C_i, C_j \in C} \text{Proximity}(C_i, C_j; D)$ 
10:
11:  if  $|C_l| + |C_m| > c_{\max}$  then
12:     $C_{\text{new}} := \{C_l, C_m\}$ 
13:  else
14:     $C_{\text{new}} := C_l \cup C_m$ 
15:  end if
16:
17:  C := (C \ { $C_l, C_m$ }) ∪ { $C_{\text{new}}$ }
18: until  $|C| = 1$ 
19:
20: return C

```

---

$$\text{Proximity}(C_i, C_j; D) = \frac{\sum_{k=1}^{|C_i|_{\text{ent}}} \sum_{l=1}^{|C_j|_{\text{ent}}} I_D(g_k; g_l)}{|C_i|_{\text{ent}} \times |C_j|_{\text{ent}}},$$

where  $g_k \in C_i$ ,  $g_l \in C_j$  (2)

By defining Proximity in this way, we can reduce the number of possible missing edges between clusters when the edge direction is restricted to a single direction between clusters in the next step. Second, edge directions between any two clusters on each hierarchy level are restricted to a single direction (see examples in Fig. 1(c) and (d)) with Algorithm 2 (C-DAGSearch), by using a selected entity named as a representative gateway node  $GN_i$  for each cluster  $C_i$  (Fig. 2). Algorithm 3 (RestrictCP) then determines  $CP(g_i)$ , which is a set of candidate parents of  $g_i$ , for each  $g_i$  ( $\in U$ ) recursively in the cluster hierarchy according to the restricted edge direction between clusters (Fig. 1(c) and (d)). This use of sets of candidate parents lets the restriction on edge directions to be applied to the structure search step. In the structure search step, the DAG search process is conducted for all the nodes, which correspond to  $n$

entities, to find  $G_U$  of high score that satisfies the restriction  $P_{G_U}(g_i) \subseteq CP(g_i)$ ,  $\forall g_i \in U$ . This whole method is outlined as H-CORE in Algorithm 4, which uses H-Cluster (Algorithm 1), C-DAGSearch (Algorithm 2) and RestrictCP (Algorithm 3) for the structure restriction step and lastly does the structure search step.

---

**Algorithm 2.** C-DAGSearch( $C, D$ )
 

---

```

1: GN := ∅
2:
3: for all  $C_i$  such that  $C_i \in C$  do
4:    $GN_i := \text{argmax}_{g_j \in C_i} \text{Proximity}(\{g_j\}, C \setminus \{C_i\}; D)$ 
5:   GN := GN ∪ { $GN_i$ }
6: end for
7:
8: Find  $G_{\text{GN}}$  maximizing  $\text{Score}(G_{\text{GN}}|D)$ 
9:  $G_C := (C, \emptyset)$ 
10:
11: for all Edge( $GN_i, GN_j$ ) in  $G_{\text{GN}}$  do
12:   Add Edge( $C_i, C_j$ ) to  $G_C$ 
13: end for
14:
15: return  $G_C$ 

```

---



---

**Algorithm 3.** RestrictCP( $C, D, c_{\max}$ )
 

---

```

1:  $G_C := \text{C-DAGSearch}(C, D)$ 
2:
3: for all  $C_i$  such that  $C_i \in C$  do
4:   for all  $C_j$  such that  $C_j \in P_{G_C}(C_i)$  do
5:     for all  $g_k$  such that  $g_k \in C_i$  do
6:        $CP(g_k) := CP(g_k) \cup \{g_l | g_l \in C_j\}$ 
7:     end for
8:   end for
9: end for
10:
11: for all  $C_i$  such that  $C_i \in C$  do
12:   if  $|C_i|_{\text{ent}} > c_{\max}$  then
13:     RestrictCP( $C_i, D, c_{\max}$ )
14:   else
15:     for all  $g_j$  such that  $g_j \in C_i$ 
16:        $CP(g_j) := CP(g_j) \cup (C_i \setminus \{g_j\})$ 
17:     end for
18:   end if
19: end for

```

---



---

**Algorithm 4.** H-CORE ( $U, D, c_{\max}$ )
 

---

```

1:  $CP(g_i) := \emptyset, \forall g_i \in U$ 
2: C := H-Cluster( $U, D, c_{\max}$ ) /* As a result, C = { $C_0$ }, where  $C_0$  is a root cluster.*/
3: RestrictCP( $C_0, D, c_{\max}$ )
4: Find  $G_U$  maximizing  $\text{Score}(G_U|D)$  that satisfies  $P_{G_U}(g_i) \subseteq CP(g_i) \forall g_i$ ,
5: return  $G_U$ 

```

---

H-Cluster is a type of hierarchical agglomerative clustering algorithm that builds a  $c_{\max}$ -nary tree dendrogram, in contrast to conventional hierarchical clustering, which builds a binary tree dendrogram. Limiting the maximum size of each cluster to  $c_{\max}$  results in C-DAGSearch per-

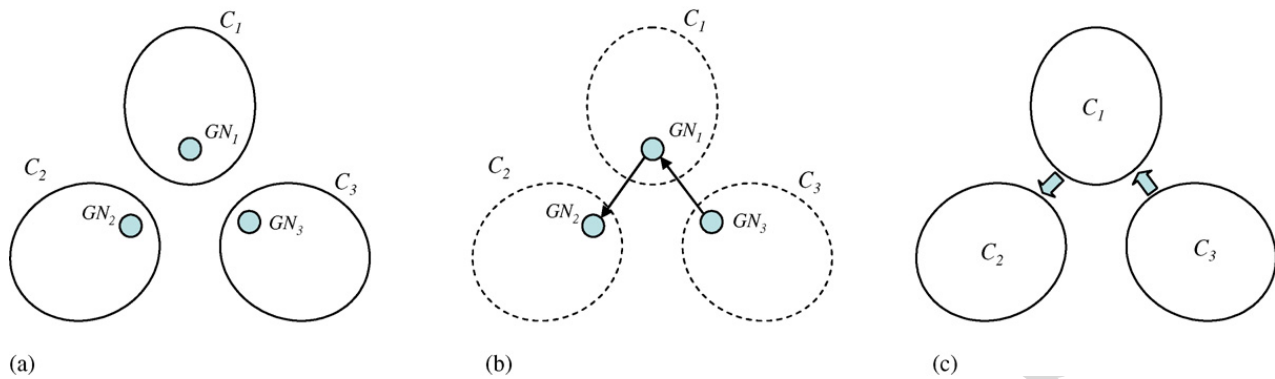


Fig. 2. The outline of C-DAGSearch. (a) Determining a representative gateway node for each cluster. (b) Searching a DAG between gateway nodes. (c) Projection of the DAG structure between gateway nodes to the DAG structure between clusters.

forming DAG searches between clusters for no more than  $c_{\max}$  clusters. In RestrictCP,  $CP(g_i)$  in each cluster is determined according to the structure of  $G_C$  learned from C-DAGSearch. The structure of  $G_C$  in C-DAGSearch is estimated using two assumptions. First, we assume that the global graph is composed of local subgraphs, since it can be considered to be a ‘graph of subgraphs’. From this assumption, we make clusters and find a graph structure between them. Second, we assume that each subgraph has a *boundary region* that includes gateway nodes connected with other subgraphs, where connections between subgraphs are only via these gateway nodes. From these assumptions, we estimate the graph structure between clusters by taking one representative gateway node  $GN_i$  for each cluster  $C_i$  and the learning for a Bayesian network using those gateway nodes (Fig. 2). We determine an entity  $g_i (\in C_i)$  that has the maximal Proximity value with the outside of cluster  $C_i$  as a representative gateway node for  $C_i$ . This is derived from the intuition that such a gateway node may be the closest one to the outside of the cluster. After learning the structure  $G_{GN}$  between these gateway nodes, this is projected to the structure of  $G_C$  between clusters by matching each  $\text{Edge}(GN_i, GN_j)$  in  $G_{GN}$  to the corresponding  $\text{Edge}(C_i, C_j)$  in  $G_C$ . By recursively applying C-DAGSearch to all of the clusters in the hierarchy, RestrictCP restricts the global structure by determining graph structures between clusters, and the restriction on global structures is applied to the DAG search process of finding  $G_U$  by determining  $CP(g_i)$  for each  $g_i \in U$ . The last process of H-CORE involves finding a target Bayesian network structure  $G_U$  for the set of all entities  $U$ . A  $G_U$  maximizing the given scoring measure is searched for while preserving the restriction on the candidate parents and thus satisfying the restriction on global structures.

H-CORE restricts the global structure between clusters as a DAG-shaped one recursively in the cluster hierarchy, which significantly reduces the number of

candidate DAG structures and makes the DAG search process extremely fast. Further, the DAG search process for finding a  $G_U$  can be implemented by applying DAG searches independently to each cluster with the corresponding sets of candidate parents if we use decomposable scoring measures such as the Bayesian score. The restriction on edge directions gives an order between clusters and optimizing the score in a cluster is independent with that in other clusters by the decomposability of the scoring measure—the total score is the sum of the score for each node and the score for each node is determined only with the node and its parents as shown in Eq. (1). Thus optimizing the score for a Bayesian network can be done with several independent optimizations in each cluster with corresponding candidate parents of entities, which is determined in the structure restriction step.

### 3.2. Evaluation with benchmark Bayesian networks

We evaluated our method using the four benchmark Bayesian networks listed in Table 1 (Abramson et al., 1996; Andreassen et al., 1991; Beinlich et al., 1989; Heckerman et al., 1992). From these benchmark networks, 5000 data instances were sampled for each variable as an observed training data set. For comparison, our H-CORE and previous SC methods were applied to the data instances to learn the target networks. Greedy hill climbing was used to find for high-scoring DAG

Table 1  
Benchmark Bayesian networks

Network	Number of nodes	Number of edges	Mean indegree	Maximum indegree
ALARM	37	46	1.24	4
HAILFINDER	56	66	1.18	4
PATHFINDER	109	195	1.79	5
DIABETES	413	602	1.46	2

structures, with a graph with no edges used as the initial graph. In each step of greedy search, a candidate DAG structure is selected for next search step, which shows the highest gain of score by applying single edge modification operation to the previous candidate. An edge modification operation is one of adding, removing and reversing an edge. The search stops if no gain of score is achieved. The scoring measure used in the search procedure was a Bayesian score with equivalent sample sizes of 3, 6 and 10. The equivalent sample size is a parameter which is related to the problem condition rather than learning algorithms and it is used to present our prior belief on the probability distribution. Larger values of equivalent sample size mean that our prior belief on the probability distribution depends on larger amount of previously observed samples and thus it reduces the effect of the training samples on the learning procedure. If we have little knowledge on the real probability distribution, larger values of equivalent sample size can bias the learning to the wrong direction (Yang and Chang, 2002). For this reason, larger values of equivalent sample size are not proper in our experiment because we do not assume any prior knowledge on the probability distribution. We consider the equivalent sample sizes which are used in our experiment are small values and proper for the experimental conditions. Further, the equivalent sample size value of 10 have been also used in other studies of Bayesian network learning including that of Brown et al. (2004). The values of  $k$  for the SC method were 5 and 10, and  $c_{\max}$  values of 15 and 30 were used in the H-CORE method.

The experiment was repeated five times with different training data, and the evaluation was done for the structural error and the number of candidate DAGs explored until convergence. The structural error of a learned network represents the number of different edge connections between two nodes compared to the original network, including missing, wrongly added and reversed

edges, and hence represents the quality of the result. The reversed edges may be considered as less serious errors than missing and wrongly added edges. However, a reversed edge breaks the causal relationship in the original network and can prohibit other true edges from being added because it can create cycles with the erroneous direction. Because considering such causal relationships is more rigorous way of evaluation than discarding reversed edges from errors, we use this structural error as a quality evaluation measure for the simulation experiment and it has been also used widely in evaluation of Bayesian network learning. The number of candidate DAGs explored during the search process is used to represent the asymptotic learning time because every search step for a candidate DAG is done by applying single edge modification operation. Further, we can easily expect the difference between the sizes of search spaces considered by two methods with the number of explored candidate structures. We list the averaged results of the five experiments in this section.

Table 2 lists the summarized result of quality and the learning speed of two methods by showing the structural error and the explored number of candidate DAG structures when we use the Bayesian scoring measure with the equivalent sample size 10. This result indicates that the H-CORE method is of comparable quality to the SC method, while it exhibits a significantly reduced learning time. Even though a simple greedy search was used for the DAG search process in both methods, the learning time increases significantly for the SC method for more than hundreds of nodes. Applying the SC method with  $k = 10$  to DIABETES did not produce a reasonable result, due to the heavy computational cost.

For more detailed description for the quality of the results, we show the specificity and the sensitivity of predicting edges by both methods in Tables 3–5. The specificity and sensitivity of predicting Edge( $g_i, g_j$ ) is

Table 2  
Summarized result of evaluation with the structural error and the number of explored candidate DAGs

Network	Structural error				Number of explored candidate DAGs			
	H-CORE		SC		H-CORE		SC	
	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$
ALARM	46	45	42	<b>33</b>	<b>10K</b>	18K	28K	58K
HAILFINDER	48	<b>43</b>	53	55	<b>10K</b>	17K	48K	99K
PATHFINDER	231	231	<b>192</b>	212	<b>28K</b>	51K	260K	566K
DIABETES	<b>590</b>	597	613	N/A	<b>80K</b>	158K	2916K	N/A

Bayesian scoring measure with equivalent sample size 10 was used. Best cases are denoted as boldface. N/A indicates no result due to heavy computational cost.



Table 3  
Specificity and sensitivity of H-CORE and SC using the Bayesian scoring measure with an equivalent sample size 3

Network	Specificity				Sensitivity			
	H-CORE		SC		H-CORE		SC	
	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$
ALARM	0.981	0.979	0.977	<b>0.985</b>	0.478	0.413	0.370	<b>0.587</b>
HAILFINDER	0.987	0.986	<b>0.992</b>	<b>0.992</b>	0.394	0.379	0.621	<b>0.636</b>
PATHFINDER	0.989	0.989	0.989	<b>0.990</b>	0.328	0.369	0.364	<b>0.379</b>
DIABETES	<b>0.997</b>	<b>0.997</b>	<b>0.997</b>	N/A	0.256	<b>0.282</b>	0.276	N/A

Table 4  
Specificity and sensitivity of H-CORE and SC using the Bayesian scoring measure with an equivalent sample size 6

Network	Specificity				Sensitivity			
	H-CORE		SC		H-CORE		SC	
	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$
ALARM	0.983	0.982	0.977	<b>0.984</b>	0.522	0.500	0.370	<b>0.565</b>
HAILFINDER	0.989	0.987	<b>0.991</b>	0.990	0.500	0.424	<b>0.591</b>	0.561
PATHFINDER	0.989	0.989	<b>0.990</b>	0.989	0.338	0.364	<b>0.379</b>	0.354
DIABETES	<b>0.997</b>	<b>0.997</b>	<b>0.997</b>	N/A	0.229	<b>0.261</b>	0.238	N/A

evaluated as follows:

$$\text{specificity} = \frac{\text{no. of true negative Edge}(g_i, g_j)}{\text{no. of false positive Edge}(g_i, g_j) + \text{no. of true negative Edge}(g_i, g_j)} \quad (3)$$

$$\text{sensitivity} = \frac{\text{no. of true positive Edge}(g_i, g_j)}{\text{no. of true positive Edge}(g_i, g_j) + \text{no. of false negative Edge}(g_i, g_j)} \quad (4)$$

The specificity shows the ratio of correct rejection for non-existing edges and the sensitivity shows the ratio of correct hit for existing edges. Tables 3–5 show that both methods show similar specificity for benchmarks. For sensitivity, both methods also shows comparable results. When we consider the best cases for each benchmark, SC achieves most of the best cases for ALARM, HAILFINDER and PATHFINDER where

small numbers of nodes (37, 56 and 109) are involved. However, H-CORE achieves best cases for the large benchmark of DIABETES involving 413 nodes. Further, SC cannot learn acceptable results in reasonable time for such large problems.

From these results, we can say that H-CORE is appropriate for large problems where SC cannot be applied. If we can get results in reasonable time by applying the SC method to some problem, SC can be a proper approach to solve the problem. However, the H-CORE method gives acceptable results in reasonable time for very large problems where the SC method cannot be applied.

#### 4. Application to large biological systems

##### 4.1. Experimental environment

To test the performance of our method in a real biological application, we used the ‘Rosetta compendium’ gene-expression profile (Hughes et al., 2000) as the target data. This is a set of gene-expression profiles

Table 5  
Specificity and sensitivity of H-CORE and SC using the Bayesian scoring measure with an equivalent sample size 10

Network	Specificity				Sensitivity			
	H-CORE		SC		H-CORE		SC	
	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$	$c_{\max} = 15$	$c_{\max} = 30$	$k = 5$	$k = 10$
ALARM	0.975	0.975	0.977	<b>0.984</b>	0.304	0.304	0.347	<b>0.543</b>
HAILFINDER	0.989	<b>0.990</b>	<b>0.990</b>	<b>0.990</b>	0.515	<b>0.545</b>	<b>0.545</b>	<b>0.545</b>
PATHFINDER	0.988	0.989	<b>0.990</b>	<b>0.990</b>	0.262	0.318	<b>0.400</b>	0.389
DIABETES	<b>0.998</b>	<b>0.998</b>	<b>0.998</b>	N/A	<b>0.312</b>	0.311	0.302	N/A

corresponding to 300 diverse mutations and chemical treatments for the entire *Saccharomyces cerevisiae* genome. For the Bayesian analysis, we selected 6148 genes by discarding those for which there were too many missing values in the gene-expression data; thus the target biological system included 6148 entities. A few missing values were imputed using Norm (Schafer, 1997). The normalized log-ratio values of gene expression were discretized into three levels while maximizing the mutual information in the gene-expression profiles (Hartemink, 2001). Even though we had 300 expression profiles, which represented a larger number of experiments compared to the data from other expression studies, they were still insufficient for learning the Bayesian networks of 6148 genes. We used a bootstrap approach to overcome this lack of information, which has been used widely in analyses of large systems (Friedman et al., 2000; Hartemink et al., 2002; Lee and Lee, 2005; Neapolitan, 2004). In this experiment, we performed 100-fold bootstrapping by applying the H-CORE method 100 times with  $c_{\max} = 80$ , so as to obtain 100 different results for confidence analysis.

The results of the H-CORE method can be significantly affected by the clustering result of H-Cluster. Because H-Cluster is deterministic and thus gives a similar result for each run, we used *randomized mutual information* rather than the original mutual information in evaluating the Proximity values so as to add small variation in the clustering result. H-Cluster uses the Proximity measure that in turn uses mutual information  $I_D(g_i; g_j)$  as the proximity value between two genes. Before the experiment, we evaluated the mutual information value between every gene pair  $(g_i; g_j)$ . Then, in each experiment, we used a random mutual information value  $I_r(g_i; g_j)$  for a gene pair  $(g_i; g_j)$  from a normal distribution with mean value  $I_D(g_i; g_j)$  and standard deviation value  $I_D(g_i; g_j)$ . By determining the standard deviation in this way, a larger variance will be given to the randomized mutual information if the original mutual information was higher one, and thus most of the randomized variation will occur between gene pairs of higher mutual information values. This randomized variation around the original mutual information value for each gene pair resulted in a variation in the clustering result of H-Cluster.

We used the 100 learned results to evaluate the confidence value of the relationship for each gene pair  $(g_i; g_j)$ :

$$\text{Confidence}(g_i; g_j) = \frac{\text{no. of results with an edge between } (g_i; g_j)}{\text{no. of experiments}} \quad (5)$$

For evaluation of the result, a popular approach is to compare the result with a known true answer on gene relationships (e.g. a true genetic network) or to find supporting true evidences from other knowledge sources (e.g. literatures). However, such an approach is limited to the case of small genetic networks because there is no known true genetic network of genome-scale and it is infeasible to find out supporting true evidences for genome-scale results by human efforts. For such genome-scale studies, one possible approach to present the correctness of a method is showing that it retrieves more meaningful results from real data than from artificial random data (e.g. Friedman et al., 2000; Keedwell et al., 2005). Even though the comparison with random cases is somehow useful in showing correctness of a method, we take more direct approach to show correctness, which is showing that our method gives consistent results with previous studies including true knowledge (this approach is also partly taken by other studies including that of Antonov et al. (2006) and Petti and Church (2005)).

To show that our method works reasonably, we performed a full text search of the SGD (Balakrishnan et al., 2006) literature database, which includes 31,000 literatures on *S. cerevisiae*, and show the correlation between the confidence of our result and the literature hit ratio. For each gene pair  $(g_i; g_j)$  in the result, we mined the database to find literatures containing sentences involving the co-occurrence of the two gene names or their synonyms. Literatures which include sentences with the co-occurrence of two genes represent evidence of a close biological relationship between them, or at least that they were studied together for a specific biological reason. In fact, there can be literatures which represent computationally predicted gene relationships. However, our result is independent of such previous computational predictions because our computational method is novel one for genome-scale Bayesian network analysis without prior knowledge.

In addition to the quality evaluation, we constructed a small set  $U_s$  of 594 genes whose standard deviations of the normalized log-ratio expression values were higher than 0.17, in order to compare our whole-genome results with those for selected genes only. It can be assumed that most of the gene pairs with high confidence can be found with a the small set of genes that have significant variations in their expression values, thus reducing the need to analyze the entire genome-scale. The results described below demonstrate that a significant number of gene pairs with a high confidence cannot be found with such selected genes. For the comparison of both methods in the case of the small set of genes, we also

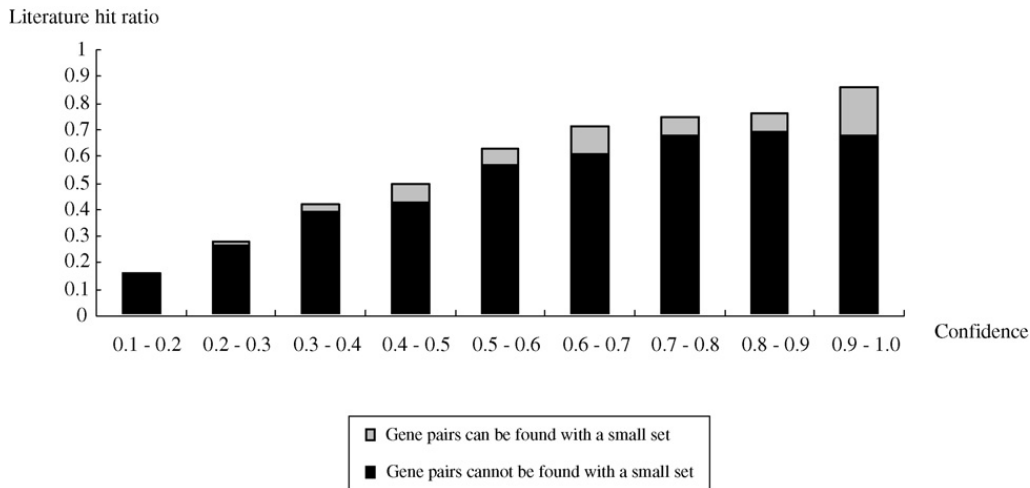


Fig. 3. Literature hit ratio for the number of gene pairs for which there are literatures. The black areas are the portions of gene pairs that cannot be found with  $U_s$ , the small set of selected genes.

applied H-CORE (with  $c_{\max} = 24$ ) and SC (with  $k = 4$ ) to  $U_s$  with 100-fold bootstrapping for each and show the compared result of confidence analysis in the following section. For SC, we could not use  $k$  larger than 4 because of heavy computational cost.

#### 4.2. Results

Fig. 3 shows the literature hit ratio for the number of gene pairs for which literatures were found from the SGD database for each confidence interval and represents that our method at least gives consistent result with the previous studies. The figure indicates that gene pairs of higher confidence values show higher hit ratio values, demonstrating that the results from the proposed genome-scale Bayesian analysis method are trustworthy, because higher literature hit ratio values represent a higher expectation of the presence of real biological relationships. Thus when we apply Bayesian analysis to large biological systems for less studies species, a result with a higher confidence will be useful in finding a novel biological knowledge because we found that the higher confidence indicates a greater expectation that there will be corresponding biological knowledge.

To show the benefit of a genome-scale Bayesian analysis relative to using a small set of selected genes, we selected the small set of genes  $U_s$  that showed a large expression variation. If either of the genes in a gene pair ( $g_i; g_j$ ) is not in  $U_s$ , that gene pair information cannot be found by investigating  $U_s$ . Fig. 3, in which the portions of gene pairs that cannot be found with  $U_s$  are shown as black areas, indicates that most of the gene pairs with corresponding literatures cannot be found with  $U_s$ . This implies that there is also considerable informa-

tion available on genes with lower expression variation. Thus we can apply our Bayesian analysis method to large biological systems to retrieve much more trustworthy information from the observed data than that can be obtained with small sets of selected biological entities.

Fig. 4 shows the compared result of confidence analysis of H-CORE and SC to the small set of data  $U_s$ . As shown in the graph, the confidence values of gene pairs from both methods are correlated each other with correlation value 0.76. This means that both methods generally give similar result when they are applied to the small set of selected genes. However, the plotted shape of

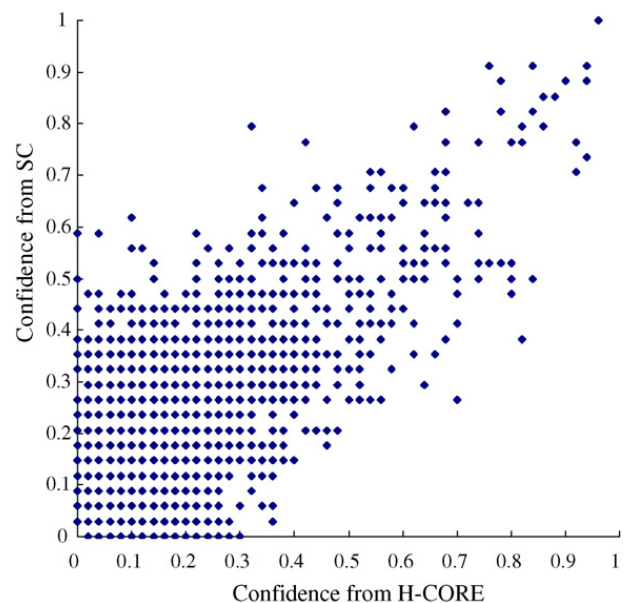


Fig. 4. Compared result of confidence analysis by applying H-CORE and SC to  $U_s$ . Each plot represents one gene pair. Gene pairs of confidence value 0 for both cases are omitted.

Table 6

Gene pairs with a confidence higher than 0.9. Descriptions are from the SGD database and the cited literature

Gene pairs	Confidence	Related description	
YHR215W (PHO12)	YAR071W (PHO11)	1.00	Both have molecular function of acid phosphatase activity
YIL169C	YOL155C	0.96	N/A <sup>a</sup>
YHR136C (SPL2)	YPL019C (VTC3)	0.96	YHR136C is phosphate-regulated. <sup>b</sup> VTC3 is involved in vacuolar polyphosphate accumulation
YML123C (PHO84)	YJL012C (VTC4)	0.96	PHO84 is a phosphate transporter. VTC4 is involved in vacuolar polyphosphate accumulation and is a vacuolar transporter chaperone
YLR158C (ASP3-3)	YLR160C (ASP3-4)	0.96	Both are involved in asparagine catabolism
YFL062W (COS4)	YGR295C (COS6)	0.96	N/A
YJL012C (VTC4)	YER072W (VTC1)	0.95	Both are involved in vacuole fusion
YLR160C (ASP3-4)	YLR155C (ASP3-1)	0.95	Both are involved in asparagine catabolism
YLR155C (ASP3-1)	YLR157C (ASP3-2)	0.95	Both are involved in asparagine catabolism
YBR012W-B	YER138C	0.95	Both are involved in Ty element transposition
YDL248W (COS7)	YHL048W (COS8)	0.95	N/A
YML132W (COS3)	YIR043C	0.95	N/A
YHR136C (SPL2)	YML123C (PHO84)	0.93	YHR136C is phosphate-regulated. <sup>b</sup> PHO84 is a phosphate transporter
YJL012C (VTC4)	YGR233C (PHO81)	0.93	VTC4 is involved in vacuolar polyphosphate accumulation. PHO81 is involved in phosphate metabolism
YLR158C (ASP3-3)	YLR155C (ASP3-1)	0.93	Both are involved in asparagine catabolism
YER189W	YEL075C	0.93	N/A
YHR215W (PHO12)	YHR136C (SPL2)	0.91	PHO12 has molecular function of acid phosphatase activity. YHR136C is phosphate-regulated <sup>b</sup>
YBR012W-A	YJR028W	0.91	Both are involved in Ty element transposition
YLL066C	YER190W (YRF1-2)	0.91	Both have function of helicase
YFL062W (COS4)	YNL336W (COS1)	0.91	N/A
YGR295C (COS6)	YJR161C (COS5)	0.91	N/A
YHL048W (COS8)	YIR043C	0.91	N/A

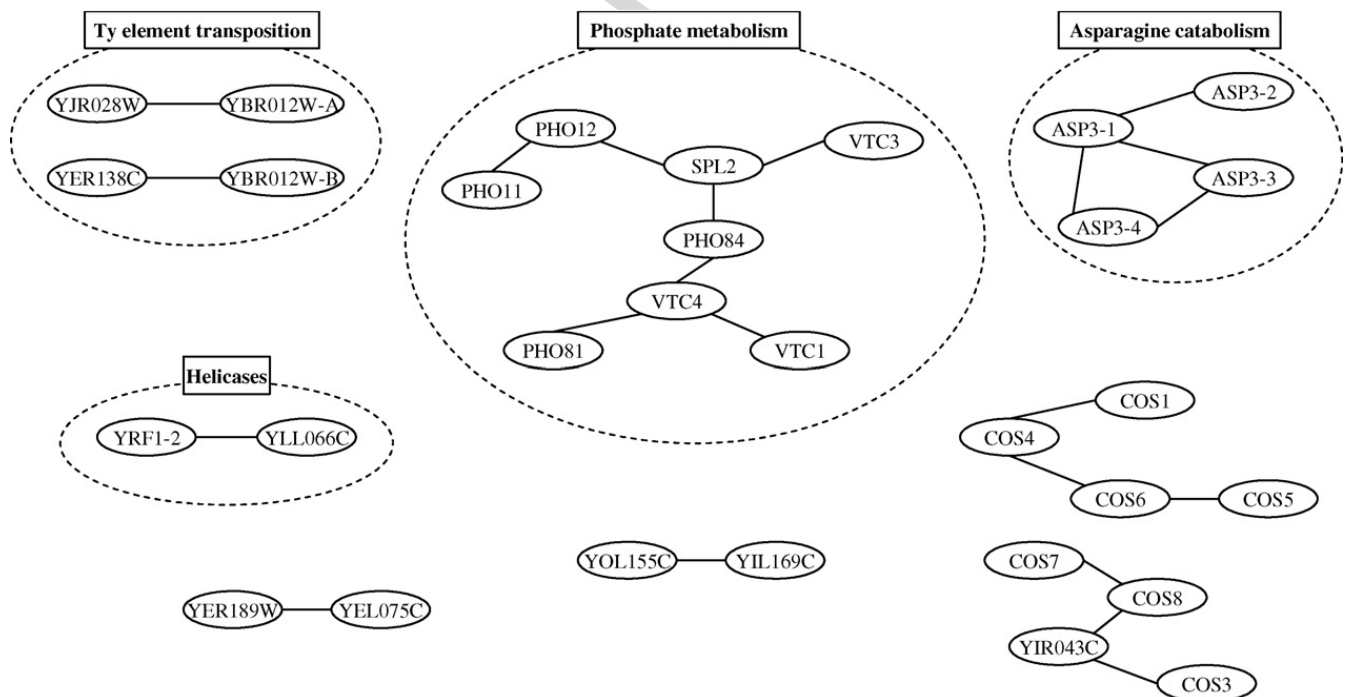
<sup>a</sup> N/A indicates no corresponding functional description found.<sup>b</sup> From Pinson et al. (2004).

Fig. 5. Network diagram of gene pairs with confidence values higher than 0.9. Possible common functional annotations are given in the boxes.



the graph is slightly biased upward. This represents that the SC method gives similar or slightly larger confidence values for gene pairs compared to the H-CORE method. From this result, we can find that the SC method may have slightly more chances of finding true knowledge compared to the H-CORE method but the difference is not significant. This result is also in agreement with that of the simulation study, where the SC method showed similar or slightly higher sensitivity compared to the H-CORE method for relatively small problems (Tables 3–5).

Table 6 lists the gene pairs for which our results indicated that the confidence values were higher than 0.9. We could find that 14 among the 22 listed gene pairs exhibit closely related functional annotations by searching the functional descriptions on each gene from the SGD database and the cited literature. The remaining gene pairs show no appropriate shared functional description because the functions of most of these genes are currently unknown, although we can predict that these pairs have close functional relationships. This assumption is supported by the network diagram drawn by connecting an undirected edge for each gene pair, as shown in Fig. 5.

Fig. 5 shows the network diagram drawn with gene pairs of confidence values higher than 0.9. There are nine connected components, with five of them showing common functional annotations. Most of the genes that are in the other four nonannotated connected components currently have no known functional annotations. YER189W and YEL075C are both hypothetical proteins with no known function. YOL155C is involved in cell wall organization and biogenesis, whereas YIL169C is a hypothetical protein that has no known function. Incidentally, the two connected components ‘COS1–COS4–

Table 7

Top 14 genes of the highest similarity with YIR043C by the BLASTN sequence query

Gene	<i>p</i> -Value
<b>COS8</b>	8.8e–123
<b>COS6</b>	7.8e–115
<b>COS1</b>	1.8e–101
<b>COS4</b>	3.3e–98
<b>COS5</b>	4.9e–98
COS2	3.2e–96
<b>COS3</b>	3.2e–96
<b>COS7</b>	3.0e–86
COS9	4.5e–75
COS10	6.9e–41
COS12	1.4e–38
YHL042W	9.8e–7
YCR102W-A	1.6e–08
PDS5	0.9992

Genes which were predicted from our result are denoted as boldface.

COS6–COS5’ and ‘COS3–YIR043C–COS8–COS7’ are mainly composed of COS genes, which suggests that these genes have close interrelationships. The inclusion of the hypothetical protein YIR043C in this group suggests that it also has close functional relationships with those of the COS genes. This is supported by the fact that YIR043C and the adjacent ORF, YIR044C, together may encode a non-functional member of the conserved, often subtelomerically encoded COS protein family (Harrison et al., 2002; Despons et al., 2006). Further, we could find that those COS genes show very high sequence similarity with YIR043C as shown in Table 7 through the BLASTN sequence query and those eight genes also show very similar expression pattern as shown in Fig. 6. Only the gene pairs with a very high confidence are shown in Fig. 5, and a deeper analysis of all the whole confidence

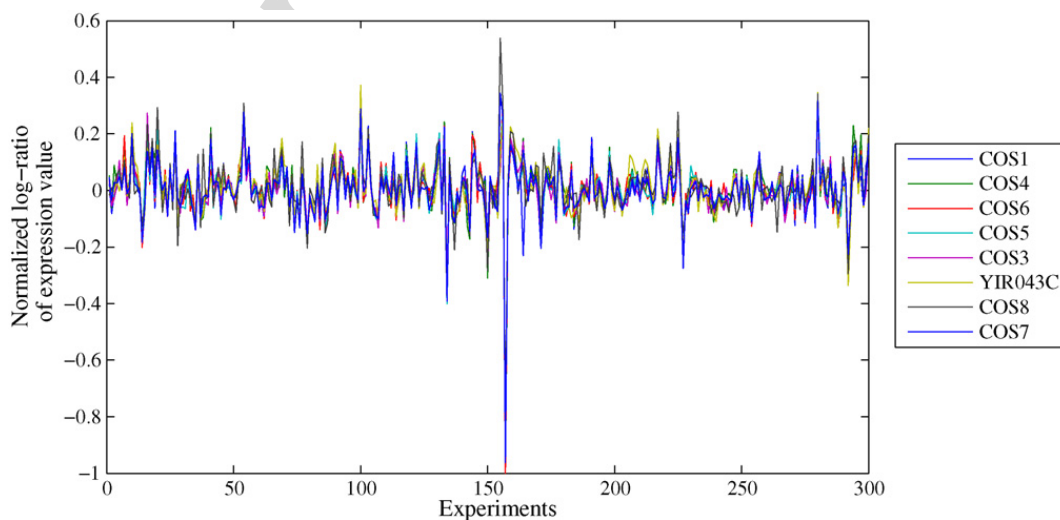


Fig. 6. Expression pattern of COS1, COS4, COS6, COS5, COS3, YIR043C, COS8 and COS7.



intervals for the network may yield further useful information on functional gene relationships.

## 5. Concluding remarks

In this study, we applied the H-CORE method to the Bayesian analysis of large biological systems using observed data only. Conventional Bayesian analysis approaches are unable to cope with thousands of entities simultaneously. The previous methods for coping with such a large number of entities need additional information or may not be suitable for a practical environment in which there is only a very small amount of observed data available. Our method can overcome this scalability problem by restricting the global structure through hierarchical clustering and restricting edge directions between those clusters.

Through evaluation with benchmark networks, we have shown that the H-CORE method works well for the large problems that conventional methods cannot cope with. The application of our method to real biological data (the Rosetta compendium) has yielded promising results through confidence analyses of gene relationships. With full text mining of the SGD literature database, which includes 31,000 literatures on yeast, we have shown that results with a higher confidence from our experiment have higher literature hit ratios, which represent higher expectations of real biological knowledge. Thus we can also trust the results of the analysis of a large-scale biological system for relatively unknown target systems. Further, the majority of the results of our Bayesian analysis of a large biological system cannot be found by investigating the small set of selected entities with high activity variation. This implies not only that our method can provide much more additional information, which cannot be found through studies with small sets of selected entities, but also that applying such a Bayesian analysis to large biological systems is useful and that biological entities with a lower degree of activity still provide useful information. Our Bayesian analysis method for large biological systems presented in this paper is effective at retrieving such information. We also compared the results of confidence analysis for both methods by applying them to the small set of selected entities. Even though the SC method might have slightly more chances of finding true knowledge compared to the H-CORE method from the selected entities, we could see the difference is not significant and both results are correlated each other.

Future studies should develop a more efficient clustering method for use in the H-CORE method, and appropriate definitions of edge expectation and order

estimation will improve the correctness of the method. Using appropriate cluster validation measures based on the size and number of clusters will yield explicit guidelines for balancing between the quality of the result and the computational cost (running time of the algorithm). Further, we can apply our approach to dynamic Bayesian networks, where the difficulty associated with the large search space is more serious than for conventional Bayesian networks (e.g. Kim et al., 2004). For further analysis we can also compare networks of biological entities from observed data with the networks of all the known prior knowledge to systematically annotate the analyzed results, easily identify new findings and suggest possible hypotheses on such new findings related to the functions of genes.

## Acknowledgements

This work was supported by the Korean Systems Biology Research Grant (2005-00343) from the Ministry of Science and Technology, National Research Laboratory Grant (2005-01450) from the Ministry of Science and Technology. We would also like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics and the IBM SUR program for providing research and computing facilities.

## References

- Abramson, B., Brown, J., Edwards, W., Murphy, A., Winkler, R.L., 1996. Hailfinder: a Bayesian system for forecasting severe weather. *Int. J. Forecasting* 12, 57–71.
- Andreassen, S., Hovorka, R., Benn, J., Olesen, K.G., Carson, E.R., 1991. A model-based approach to insulin adjustment. *Proceedings of the Third Conference on Artificial Intelligence in Medicine*, pp. 239–248.
- Antonov, A.V., Tetko, I.V., Mewes, H.W., 2006. A systematic approach to infer biological relevance and biases of gene network structures. *Nucleic Acids Res.* 34, e6.
- Balakrishnan, R., et al., 2006. *Saccharomyces Genome Database*. <http://www.yeastgenome.org/>.
- Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F., 1989. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*.
- Brown, L.E., Tsamardinos, I., Aliferis, C.F., 2004. A Novel Algorithm for Scalable and Accurate Bayesian Network Learning. *MED-INFO*.
- Chickering, D.M., 1996. Learning Bayesian Networks is NP-Complete. *AI & STAT V*.
- Daub, C.O., Steuer, R., Selbig, J., Kloska, S., 2004. Estimating mutual information using B-spline functions—an improved similarity measure for analysing gene expression data. *BMC Bioinform.* 5, 118.
- de Jong, H., 2002. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* 9, 67–103.

- Despons, L., Wirth, B., Louis, V.L., Potier, S., Souciet, J.L., 2006. An evolutionary scenario for one of the largest yeast gene families. *Trends Genet.* 22, 10–15.
- Friedman, N., Nachman, I., Pe'er, D., 1999. Learning Bayesian network structure from massive datasets: the “Sparse Candidate” algorithm. *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 206–215.
- Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* 7, 601–620.
- Harrison, P., Kumar, A., Lan, N., Echols, N., Snyder, M., Gerstein, M., 2002. A small reservoir of disabled ORFs in the yeast genome and its implication for the dynamics of proteome evolution. *J. Mol. Biol.* 316, 409–419.
- Hartemink, A.J., 2001. Principled computational methods for the validation and discovery of genetic regulatory networks. Ph.D. Thesis. MIT.
- Hartemink, A.J., Gifford, D., Jaakkola, T., Young, R., 2002. Combining location and expression data for principled discovery of genetic regulatory networks. *Proceedings of the Pacific Symposium on Biocomputing 2002 (PSB02)*, pp. 437–449.
- Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learn.* 20, 197–243.
- Heckerman, D.E., Horvitz, E.J., Nathwani, B.N., 1992. Toward normative expert systems. Part I. The pathfinder project. *Meth. Inf. Med.* 31, 90–105.
- Hughes, T.R., et al., 2000. Functional discovery via a compendium of expression profiles. *Cell* 102, 109–126.
- Hwang, K., Lee, J., Chung, S., Zhang, B., 2002. Construction of large-scale Bayesian networks by local to global search. *PRICAI*, 375–384.
- Keedwell, E., Narayanan, A., 2005. Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 2, 231–242.
- Kim, S., Imoto, S., Miyano, S., 2004. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems* 75, 57–65.
- Lee, P., Lee, D., 2005. Modularized learning of genetic interaction networks from biological annotations and mRNA expression data. *Bioinformatics* 21, 2739–2747.
- Neapolitan, R.E., 2004. *Learning Bayesian Networks*. Pearson Prentice Hall, New Jersey.
- Peña, J.M., Björkegren, J., Tegnér, J., 2005. Growing Bayesian network models of gene networks from seed genes. *Bioinformatics* 21 (Suppl. 2), ii224–ii229.
- Petti, A.A., Church, G. M., 2005. A network of transcriptionally coordinated functional modules in *Saccharomyces cerevisiae*. *Genome Res.* 15, 1298–1306.
- Pinson, B., Merle, M., Franconi, J., Daignan-Fornier, B., 2004. Low affinity orthophosphate carriers regulate PHO gene expression independently of internal orthophosphate concentration in *Saccharomyces cerevisiae*. *J. Biol. Chem.* 279, 35273–35280.
- Robinson, R.W., 1973. *Counting Labeled Acyclic Digraphs*. *New Directions in the Theory of Graphs*. Academic Press, New York pp. 239–273.
- Schafer, J.L., 1997. *Analysis of Incomplete Multivariate Data*. Chapman and Hall.
- Yang, S., Chang, K.C., 2002. Comparison of score metrics for Bayesian network learning. *IEEE Trans. Syst. Man Cybernetics Part A: Syst. Hum.* 32, 419–428.