

SiBIC: a Tool for Generating a Network of Biclusters Captured by Maximal Frequent Itemset Mining

**Kei-ichiro Takahashi · David A. duVerle ·
Sohiya Yotsukura · Ichigaku Takigawa ·
Hiroshi Mamitsuka**

Received: date / Accepted: date

Abstract Biclustering extracts coexpressed genes under certain experimental conditions, providing more precise insight into the genetic behaviors than one-dimensional clustering. For understanding the biological features of genes in a single bicluster, visualizations such as heatmaps or parallel coordinate plots and tools for enrichment analysis are widely used. However, simultaneously handling many biclusters still remains a challenge. Thus, we developed a web service named SiBIC, which, using maximal frequent itemset mining, exhaustively discovers significant biclusters, which turn into networks of overlapping biclusters, where nodes are gene sets and edges show their overlaps in the detected biclusters. SiBIC provides a graphical user interface for manipulating a gene set network, where users can find target gene sets based on the enriched network. This chapter provides a user guide/instruction of SiBIC with background of having developed this software. SiBIC is available at <http://utrecht.kuicr.kyoto-u.ac.jp:8080/sibic/faces/index.jsp>.

Keywords Gene expression · Biclustering · Frequent itemset mining · Gene set network · Gene enrichment analysis

K. Takahashi
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan.
E-mail: keiichiro@kuicr.kyoto-u.ac.jp

D. A. duVerle
Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, University of Tokyo, Kashiwa, Japan.

S. Yotsukura
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan.

I. Takigawa
Division of Computer Science and Information Technology, Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan.

H. Mamitsuka
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan, and Department of Computer Science, Aalto University, Finland.

1 Introduction

Gene expression matrix (‘genes’ \times ‘experimental conditions’) can be clustered by either of the two sides [1,2], while expression patterns can usually be grouped with only part of rows or columns (neither the entire rows nor columns). This leads to the idea of biclusters, which consist of subgroups of rows and subgroups of columns.

In general, from an expression matrix, biclustering algorithms produce many biclusters [3,4], causing a serious issue of visualizing them. To solve this issue, biclusters are visualized in many ways, such as heatmaps and parallel coordinate plots [5–8]. However they have limitation on scalability, particularly for many overlapping biclusters.

Another type of visualization is graph, which is more promising on handling many biclusters. Furby [9] displays overlapping biclusters as a graph, where a node corresponds to a heatmap (a bicluster itself) and edges correspond to rows and columns shared by two heatmaps. BicOverlapper [10] visualizes overlapping biclusters by a graph, in which each node represents a gene or a condition and edges are grouped by one or more biclusters. We developed SiBIC [11] by defining a weighted graph called as a *gene set network* on overlapping biclusters, where each node is a gene set derived from overlapping biclusters and each edge corresponds to the difference between two nodes. A gene set network removes duplications of genes, which share experimental conditions. This makes gene set networks more compact than Furby. Similarly, a gene set network is more compact (and scalable) than a network by BicOverlapper, because each node of gene set networks is a gene set, while each node of BicOverlapper is a single gene or a single condition. SiBIC also provides a GUI application for visualizing and manipulating gene set networks, to allow enrichment analysis in a more flexible manner than using only one bicluster.

In SiBIC, a bicluster is defined as genes that are coexpressed under each experimental condition. Fig. 1b shows an example of such biclusters, where values are similar in each column. This bicluster reveals genes which express similarly under certain experimental conditions. To exhaustively find this type of biclusters from a given expression dataset, SiBIC employs frequent itemset mining (FIM) [12]. SiBIC regards every set of genes sharing similar expression values as one item, and frequent itemsets enumerated by FIM as biclusters. SiBIC generates biclusters from FIM and then a gene set network from biclusters [11]. This book chapter provides a comprehensive user instruction of SiBIC.

2 Materials

The input of SiBIC is gene expression data, which is a matrix, in which rows are genes and columns are experimental conditions. We describe the format of input files in Section 3.2.1. Note that SiBIC exhaustively enumerates all possible significant biclusters and so is not necessarily designed for dealing with large-scale expression datasets. We recommend that a subset of interest should be extracted from the original gene expression dataset.

3 Methods

3.1 Overview of SiBIC

Our method consists of roughly four steps: 1) enumerating all possible biclusters as frequent itemsets and assigning p -values to them, 2) merging the overlapping biclusters, removing their redundancy, 3) generating gene set networks from merged biclusters and 4) analyzing gene functions by using the generated gene set networks. Fig. 2(1) to (4) show a schematic flow of the above 1) to 4), respectively.

3.1.1 Enumerating Biclusters

Our approach produces multiple, overlapping biclusters by frequent itemset mining (FIM). SiBIC first aggregates genes with similar values into items per experimental condition, and then FIM (MAFIA [13]) is run on the database of all items. Fig. 1a shows an explanatory example of a frequent itemset, which can be seen as a bicluster, as shown in Fig. 1b.

SiBIC computes empirical p -values to rank generated biclusters in terms of how significantly row vectors in biclusters are correlated. For each bicluster (with N genes and M experimental conditions), 500,000 matrices of the same size are randomly generated out of the input gene expression matrix. To generate an empirical distribution for a bicluster, SiBIC computes the following test statistic T over each random matrix:

$$T = \frac{1}{N} \sum_{i=1}^N \text{corr}(g_i, \bar{g}), \text{ where } \bar{g} = \frac{1}{N} \sum_{i=1}^N g_i \quad (1)$$

where g_i is a M -dimensional row vector and $\text{corr}(\cdot, \cdot)$ is Pearson correlation coefficient. SiBIC then calculates the T score of the bicluster to give its empirical p -value on the distribution.

3.1.2 Merging Biclusters

Maximal FIM enumerates all possible biclusters of the largest frequent itemsets, while they can be redundant in the sense that they can be still heavily overlapped with each other. SiBIC merges biclusters that have exactly the same experimental conditions, as long as the significance as computed in Eq. (1) is kept.

3.1.3 Gene Set Networks

To visualize the biclusters, we use *gene set networks*, each being a weighted graph, where a node corresponds to a coexpressed gene set and an edge indicates the difference of experimental conditions between two nodes. SiBIC generates gene set networks from overlapping biclusters as follows: 1) All genes in overlapping biclusters are first divided into disjoint gene sets so that respective sets are shared by the same biclusters. 2) SiBIC then treats each set as a node, and an edge connects two nodes if both nodes are taken from the same bicluster, where the weight of an edge

is the number of biclusters containing the genes of the two nodes of the edge. Fig. 3 shows an explanatory example of building a gene set network (b) from overlapping biclusters (a). From Fig. 3(b), we can easily see that each node represents a newly redefined bicluster with an expression pattern.

A gene set network has the following three properties:

- 1) Reversibility: a gene set network exhaustively keeps overlapping bicluster information, by which the original biclusters can be reproduced from a gene set network.
- 2) Compactness: each node is a gene set (a bicluster), by which a gene set network is more compact than usual gene networks.
- 3) Interpretability: an edge corresponds to the difference between two nodes of this edge, by which nodes can be interpreted as coexpressed gene units.

3.2 Web Service

In this section, we describe the usage of SiBIC (*see Note 1*), which has two steps: 1) selecting an expression dataset, 2) inputting parameters. In the first step, an expression dataset is uploaded or a SOFT file is selected from the GEO repository [14], which will be described in Section 3.2.1. In the second step, Section 3.2.2 describes parameters details. Section 3.2.3 and 3.2.4 explain how to analyze and interpret biclusters and gene set networks.

3.2.1 Expression Dataset

The top page of SiBIC provides the following two interfaces to send an expression dataset to the server:

File uploading interface (Fig. 4a):

A local plain text file with the extension .txt or .text can be uploaded by clicking the 'Choose File' button to pick up a local file and then clicking the 'submit' button. The file must be a tab-delimited file, where the maximum file size is 15MB and the file format must have sample identifiers in the first line and a gene identifier followed by expression values in the following lines. Lines starting with a symbol '#' or '!' are ignored. Note that no identifier can start with a symbol '#' or '!'. Missing expression values can be entered as null or na.

GEO dataset search interface (Fig. 4b):

A query in the text field can be typed and the 'search' button can be clicked to list the matched SOFT files from the GEO database [14]. Then the 'select' button can be clicked to obtain the corresponding SOFT file. Query syntax should follow ESearch format [15], where a space should be replaced by a plus sign '+' if required in a query term (*see Note 2*).

3.2.2 Parameters

Submitting the input expression dataset directs to the parameter setting page. Fig. 5 shows a screenshot of the page, where MIN_ROW, MIN_COL and MAX_DIFF

are the main three parameters, which define biclusters. MIN_ROW and MIN_COL specify the minimum size of biclusters, and MAX_DIFF gives the maximum range of values in each column. It is not easy for users to decide MAX_DIFF, because a proper range depends upon experimental conditions. Thus, instead of MAX_DIFF, SiBIC has BIN as a parameter to easily compute MAX_DIFF. Below are all parameters to be specified in the parameter input interface of SiBIC.

- 1) MIN_ROW: specifies the minimum number of genes in biclusters. Computation time becomes heavier as MIN_ROW is smaller, because the number of frequent itemsets (i.e. biclusters) is larger. An integer of 5 or larger is the possible input. The default value is 10.
- 2) BIN: defines MAX_DIFF, which is $(MAX-MIN)/BIN$, where MAX and MIN are the maximum and minimum expression values per experimental condition, respectively. SiBIC handles the combination of i) a gene set with values within MAX_DIFF and ii) an experimental condition as an item for maximal FIM. Note that assuming that some Gaussian distribution over expression values, the number of genes in one item is larger as the MAX_DIFF is wider, particularly including the mean expression value. To remove outliers, SiBIC internally modifies the distribution's tails, by replacing all expression values falling outside of $(\hat{\mu} - 3\hat{\sigma}, \hat{\mu} + 3\hat{\sigma})$ with $\mu \pm 3\hat{\sigma}$, by which $MAX = \hat{\mu} + 3\hat{\sigma}$, $MIN = \hat{\mu} - 3\hat{\sigma}$, and $MAX_DIFF = 6\hat{\sigma} / BIN$. Note that smaller BIN allows larger biclusters with less similarity in expression values and more experimental conditions. The default value is 7.
- 3) MIN_COL: specifies the minimum number of experimental conditions in biclusters. After running maximal FIM and before computing p -values, biclusters with a smaller number of experimental conditions than MIN_COL are filtered out. The default value is 3.
- 4) SD_COEFF: is used to remove expression values (for each column) with only little changes and biologically insignificant. Genes with expression values within $SD_COEFF \times SD$ (SD : standard deviation) are removed. Alternatively, a user can set $SD_COEFF=0.0$ and specify a range of interest by PERCENTILE.
- 5) ABS: is a boolean parameter to treat expression values as absolute values. The default value is 'false' (unchecked in the check button).
- 6) TEST: is a parameter to choose a method for computing p -values, out of three choices:
 - 'genes': p -values are computed in terms of how much genes in a bicluster are correlated each other by Eq. (1).
 - 'conds': p -values are computed in terms of how much experimental conditions in a bicluster are correlated each other.
 - 'both': both genes and conditions are used.
 When TEST is set, the cutoff value for p -values can be also specified. The default setting is 'genes' with the cut-off value of 0.01.
- 7) GENES: is a parameter to specify genes of interest. The input is identifiers (separated by ',') of genes, which should be included in biclusters. If this parameter is used, biclusters without the specified genes are ignored, making the computation far faster.

- 8) **MERGER**: is a boolean parameter to skip the merging process. The default value is 'true' (checked in the check button).

Another possible input is an email address to receive a notification e-mail immediately after the result is obtained. After inputting all parameters and clicking 'confirm', 'run' can be clicked if there are no problems on the input parameter values; otherwise click 'back' to go back to the parameter input interface to input parameter values again. By clicking 'run' in the confirmation page, SiBIC moves to the running status page as shown in Fig. 6, where the status of computation (*see Note 3*) can be shown. In the running status page, the 'cancel' button to cancel the current job can be clicked. The URL to the page showing results, provided on the running status page, is recommended to be saved as a bookmark. Note that this URL is the only way to access the result, if the email address has not been given in the parameter setting page,

Practically several different parameter sets are recommended to be tried to find a good balance to obtain favorable biclusters without spending much computation time. Computation time directly depends on the number of biclusters to be generated, which further depends on two factors: 1) the size of items (the number of genes in each item) and 2) the number of items (the size of an expression matrix). We mention a couple of points regarding them below.

- 1) **Size of items**: Simply smaller BIN (the larger size of items) can generate a larger number of biclusters, taking longer computation time. Also with a larger number of genes in one item (which result in being less similar each other), candidate biclusters can be less statistically significant, despite of longer computation time (since the number of experimental conditions of biclusters can be larger). Thus several values of MIN_ROW and BIN should be tried, where a larger value of MIN_ROW results in less computation time, and a larger value of BIN can also reduce the computation time by making the size of items smaller and expression values more similar. The progress of the running job can be checked through the running status page (Fig. 6). If the progress is very slow, the job can be canceled to try another parameter set (*see Note 4*).
- 2) **Number of items**: SiBIC produces $O(MN)$ items for M genes and N experimental conditions. The search space of FIM is larger due to a larger number of items, meaning that a larger expression dataset is heavier in computation. A simple way of making an expression dataset smaller is to remove genes by making SD_COEFF larger (or PERCENTILE). Also another way is to specify particular genes of interest by using GENES, by which SiBIC focuses on only biclusters with those genes. Furthermore, if the expression dataset has replicates of experimental conditions or genes, they should be removed (by taking the average over the replicated, etc.) to make the input file smaller (*see Note 5*).

3.2.3 Biclusters

The result page shown in Fig. 7a can be accessed through the bookmark link or the link in the notification email mentioned in Section 3.2.2. The result page has two parts: 'Bicluster Information' and 'Gene Set Networks for Overlapping Biclusters'.

Below we explain ‘Bicluster Information’ and ‘Gene Set Networks for Overlapping Biclusters’ will be explained in the next section.

‘Bicluster Information’ shows a table of biclusters with their sizes, heights, widths and p -values, where biclusters can be shown by clicking a triangle icon right beside the column titles. Also, the size of biclusters can be adjusted by the ‘Top’ dropbox or by searching biclusters with genes specified in the query box right above the table. An individual page for the bicluster shown in Fig. 7b can be accessed by clicking each ID in the table. The individual page contains a heatmap, a line chart and a numerical table of the bicluster. Moreover, SiBIC provides an interface for enrichment analysis by using DAVID (Database for Annotation, Visualization and Integrated Discovery) [16] on the bottom of the page. Enrichment analysis can be done for the bicluster, through this interface, as follows:

1. Select a proper ‘Query Type’ out of ID_REFF, NAME_ID, ACC, and GENE_ID each of which corresponds to a column of the bicluster table above.
2. Select a proper ‘Gene Type’ defined in DAVID such as ENTREZ.GENE_ID, which must be a whole background containing ‘Query Type’.
3. Select a ‘Tool’ such as ‘Functional Annotation Chart’ for annotation analysis.
4. Select ‘Annotations’ such as GOTERM_BP_FAT and KEGG_PATHWAY, and then click the DAVID logo. If ‘Query Type’ and ‘Gene Type’ are a correct combination, then SiBIC opens a DAVID page to show the result.

3.2.4 Gene Set Network Viewer

‘Gene Set Networks for Overlapping Biclusters’ shows a table of gene set networks. In this table, the ‘GNS file’ column allows to download .gns files to run with the GUI application to be described below. The ‘overlap’ column shows the number of overlapping biclusters from which a gene set network is made, the ‘genes’ column shows the number of genes in the network, the ‘vertex’ column shows the number of nodes, and the ‘edge’ column shows the number of edges.

Fig. 8 shows the GUI application for displaying gene set networks which enables to conduct enrichment analysis in a more flexible manner than using one bicluster only. For example, genes in the node with the maximum degree and its neighboring nodes or genes in the most significant bicluster can be selected. The application is available from the ‘Get gene set network viewer’ button on the result page (right under the table of gene set networks) in Fig. 7a. The GUI is developed using Java 8 Swing and JUNG (Java Universal Network/Graph Framework) library [17]. The application can be launched by the following command

```
java -jar sibic_gsn_app.jar
```

or by right-clicking the jar file to open a dialog box, which will show a menu item for launching the application. The input of the application is a GNS (Gene Set Network) file, which is a binary file containing information about a gene set network. GNS files can be downloaded by clicking ‘download’ in the ‘GNS file’ column of the gene set networks’ table in the result page. JRE (Java Runtime Environment) 1.8 or later must be installed on the local machine before launching the application.

As shown in Fig. 8, the GUI has left and right panes:

Left pane: has a drawing controller on the top and a network viewer on the bottom.

In the top, the network 'LAYOUT' can be selected out of four types: 'Circle', 'KK', 'FR' and 'ISOM'. The default layout is 'Circle', which positions nodes equally spaced on a regular circle, while 'KK', 'FR' and 'ISOM' use the Kamada-Kawai algorithm [18], the Fruchterman-Reingold force-directed algorithm [19], and a layout algorithm based on Meyer's self-organizing graph methods [20], respectively. The diameter of nodes indicates the number of genes.

The 'MODE' of the network viewer can be further selected, where 'Picking' allows to pick and drag the nodes of interest, while 'Transforming' enables to drag the whole network. Both modes allow to zoom in or out the network view by scrolling with the mouse wheel. Subnetworks can be filtered out by adjusting 'VERTEX SIZE' or 'EDGE WEIGHT'.

By unchecking the 'CONNECTED' box above, the network viewer can show unconnected networks. Clicking a vertex under the 'Picking' mode updates the information in the right pane.

A helpful function of this side is that multiple nodes are clickable at the same time by dragging the mouse to make a rectangle so that it encompasses the multiple nodes, and clicking one of the selected nodes.

Right pane: has four tabs, 'SELECTED', 'BICS', 'GENES' and 'NODES', to display various information of the gene set network in the left pane.

'SELECTED' shows information on the selected nodes in the network, with 'SELECTED NODES INFO' in the top and 'NEIGHBORING NODES' in the bottom. 'SELECTED NODES INFO' further consists of four subtabs, 'GENE', 'COND', 'BIC' and 'HEATMAP'. 'GENE' (Fig. 9a) shows a table of genes in the selected nodes, where data can be copied to the clipboard by dragging target cells and clicking the cells. 'COND' (Fig. 9b) and 'BIC' (Fig. 9c) show a table of experimental conditions and a table of biclusters, respectively. 'HEATMAP' (Fig. 9d) shows a list of heatmaps for the respective nodes. These heatmaps are not those of merged biclusters but of Fig. 3b. Finally 'NEIGHBORING NODES' displays a list of neighboring nodes of the selected nodes.

'BICS', 'GENES' and 'NODES' provide the information on the entire network in the left pane. 'BICS' shows a table of all biclusters of the network. 'GENES' shows a table of all genes in all nodes in the network. 'NODES' shows a table of features of all nodes, such as the degree and weighted degree.

By using the left and right panes, genes of interest can be picked up to check, following the basic flow of manipulating a gene set network:

1. Click node(s) in the network viewer (the left pane in Fig. 8).
2. Find genes of interest in the information tables (the right pane in Fig. 8).
3. Copy the genes into your clipboard by right-click and run a third-party tool to perform functional analysis.

Here we describe two detailed scenarios for functional analysis:

- 1) Finding particular genes in the most significant bicluster in a gene set network
 1. Click any node in the network in the left pane, to activate the right pane.
 2. Click 'BICS' in the right pane and then click 'P(GENE)' (p -values) to sort the table.

3. Click a cell in the row of the most significant bicluster.
 4. Open a popup by right-click and select 'Find nodes'.
 5. Click the nodes highlighted in aqua blue in the network viewer.
 6. Click 'SELECTED' in the right pane to see the information such as gene names and heatmaps.
 7. Copy the target genes in 'GENE' to the clipboard by right-click for functional analysis.
- 2) Finding particular genes in the node with the maximum weighed degree and its neighboring nodes
1. Click any node in the network viewer.
 2. Click 'NODES' and then click 'W.DEG' (weighted degree) to sort the table.
 3. Click a cell in the row of the node with the maximum weighted degree.
 4. Open a popup by right-click and select 'Find nodes'.
 5. Click the nodes highlighted in aqua blue in the network viewer.
 6. Click 'SELECTED' in the right pane to see the information such as gene names and heatmaps.
 7. Copy the target genes from 'GENE' and 'NEIGHBORING NODES' to the clipboard by right-click for functional analysis.

4 Notes

1. SiBIC is available at <http://utrecht.kuicr.kyoto-u.ac.jp:8080/sibic/faces/index.jsp>. For a quick start, SiBIC has a tutorial page at <http://utrecht.kuicr.kyoto-u.ac.jp:8080/sibic/faces/howto.jsp>. We note that SiBIC cannot handle concurrent multiple jobs from a single user. The user must run only one job at a time.
2. In an ESearch query, users can specify a term with a 'field tag' such as [orgn] for 'organism' or [n_samples] for 'the number of samples'. Note that if no field tag is specified for a term, the term is directed to all fields. Below are some query examples.
 - `saccharomyces+cerevisiae[orgn]+AND+4:7[n_samples]`
 - `log+ratio[vtyp]+OR+log2+ratio[vtyp]+OR+log10+ratio[vtyp]`
 - `cancer[title]+AND+(homo+sapiens[orng]+OR+mus+musculus[orgn])`
3. SiBIC was a single-server platform in [11], Currently, the architecture of SiBIC is the combination of a web server with a high-end fast server. Thus the server response has been much improved. SiBIC generates a five times larger number of random matrices than [11] to compute p -values, by concurrent and distributed computing, which makes p -values more precise. However, MAFIA [13] does not support concurrent computing, by which the computation time of FIM has not been improved so drastically. Hence, the running status page should be checked to see if FIM takes an extraordinary large computation time.
4. In more detail, if FIM takes more than one minute, the number of bicluster candidates generated is already intractable for computing p -values. So the job should

be canceled and another parameter set to generate a smaller dataset should be tried.

5. For a GEO dataset, SiBIC can provide a ‘unified expression matrix’ (.uem) file at the bottom of the parameter input interface, which is made by simply taking the average over replicates and can be uploaded as the input in the top page of SiBIC.

Acknowledgements Part of this research has been supported by MEXT KAKENHI #16H02868 and #17H01783, ACCEL and PRESTO of JST and FiDiPro of Tekes.

References

1. Jiang D, Tang C, Zhang A (2004) Cluster analysis for gene expression data: a survey. *IEEE Trans Knowl Data Eng* 16(11):1370-1386
2. A Ben-Dor, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. *J Comput Biol* 6(3-4): 281-297
3. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. *IEEE Trans Comput Biol Bioinformatics* 1(1):24-45
4. Saber HB, Elloumi M (2015) DNA microarray data analysis: a new survey on biclustering. *Int J Comput Biol* 4(1):21-37
5. Barkow S, Bleuler S, Prelic A, Zimmermann P, Zitzler E (2006) BicAT: biclustering analysis toolbox. *Bioinformatics* 22:1282-1283
6. Cheng KO, Law NF, Siu WC, Lau TH (2007) BiVisu: software tool for bicluster detection and visualization. *Bioinformatics* 23 (17): 2342-2344
7. Grothaus GA, Mufti A, Murali TM (2006) Automatic layout and visualization of biclusters. *Algorithms Mol Biol* 1:15
8. Heinrich J, Seifert R, Burch M, Weiskopf D (2011) Bicluster viewer: a visualization tool for analyzing gene expression data. In: *Advances in Visual Computing*, Springer 641-652.
9. Streit M, Gratzl S, Gillhofer M, Mayr A, Mitterecker A, Hochreiter S (2014) Furby: fuzzy force-directed bicluster visualization. *BMC Bioinforma* 15(Suppl 6):4
10. Santamaria R, Theron R, Quintales L (2008) BicOverlapper: a tool for bicluster visualization. *Bioinformatics* 24(9):1212-1213
11. Takahashi K, Takigawa I, Mamitsuka H (2013) SiBIC: A web server for generating gene set networks based on biclusters obtained by maximal frequent itemset mining. *PLoS ONE* 8(12):e82890
12. Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. *Data Min Knowl Discov* 15: 55-86
13. Burdick D, Calimlim M, Flannick J, Gehrke J, Yiu T (2005) MAFIA: A maximal frequent itemset algorithm. *IEEE Trans on Knowl and Data Eng* 17:1490-1504
14. Edgar R, Domrachev M, Lash AE (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 30:207-210
15. Sayers E, Wheeler D (2004) Building customized data pipelines using the Entrez Programming Utilities (eUtils). NCBI
16. Huang DW, Sherman BT, Lempicki RA (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols* 4:44-57
17. Madadhain J, Fisher D, Smyth P, White S, Boey Y (2005) Analysis and visualization of network data using JUNG. *J Stat Soft* 10:1-35
18. Kamada T, Kawai S (1989) An algorithm for drawing general undirected graphs. *Inform Process Lett* 31:7-15
19. Fruchterman TMJ, Reingold EM (1991) Graph drawing by force-directed placement. *Softw-Pract Exp* 21:1129-1164
20. Meyer B (1998) Self-organizing graphs-a neural network perspective of graph layout. In *Graph Drawing Symposium*, August 1998.

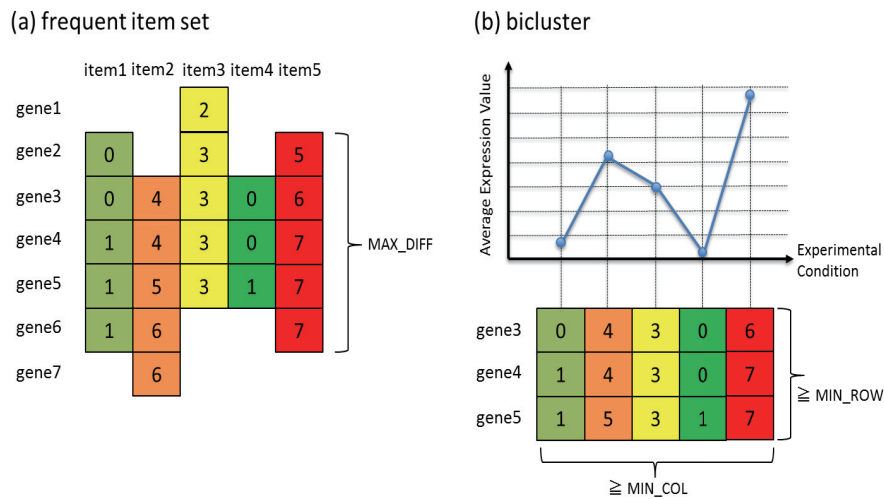


Fig. 1 (a) In SiBIC, genes having similar expression values within MAX_DIFF are dealt as one item. Note that items can have different numbers of genes. The figure assumes that each item is taken from different experimental conditions and the five items (item1 to item5) have the common genes (gene3 to gene5). The minimum number of common genes are specified by parameter MIN_ROW (in the bottom of the right figure (b)). SiBIC captures such items as a frequent itemset and enumerates all possible frequent itemsets from a given expression dataset. (b) The frequent itemset in (a) can be seen as a bicluster consisting of the common three genes and the five experimental conditions. This bicluster shows a coexpressed pattern in the top. MIN_ROW and MIN_COL are the parameters which specifies the size of biclusters: minimum number of genes and minimum number of conditions, respectively.

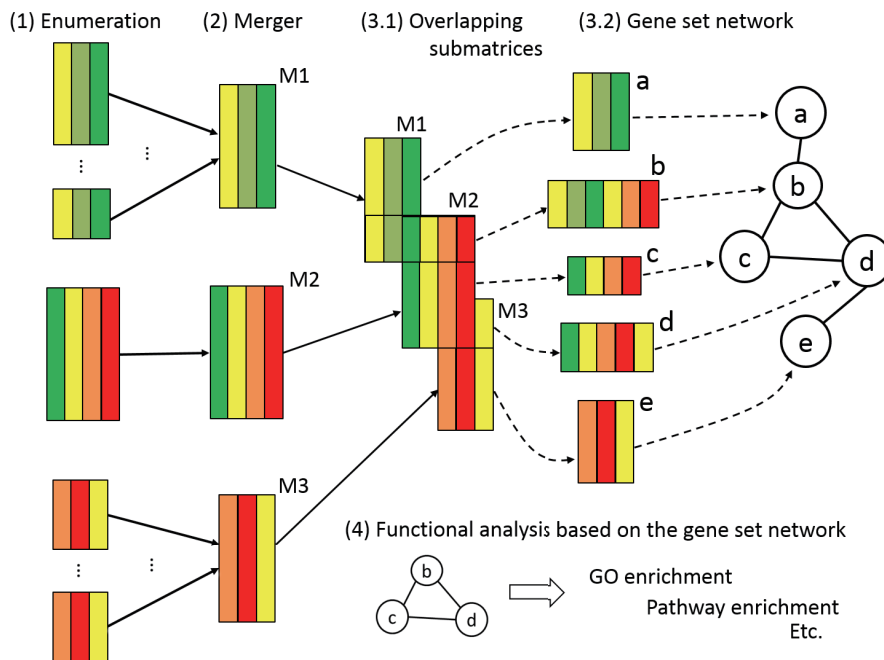


Fig. 2 A schematic flow of SiBIC: (1) Enumerating all biclusters by maximal frequent itemset mining, where each rectangle represents a maximal frequent itemset (i.e. a bicluster). Colors stand for similarities in expression values. (2) Merging overlapping biclusters with exactly the same conditions if they keep statistical significance. (3.1) and (3.2) Generating gene set networks from overlapping biclusters. Each node in the network indicates a newly redefined bicluster based on the overlapping submatrices and nodes from each bicluster form a complete subgraph. (4) Analyzing gene functions by using the obtained network.

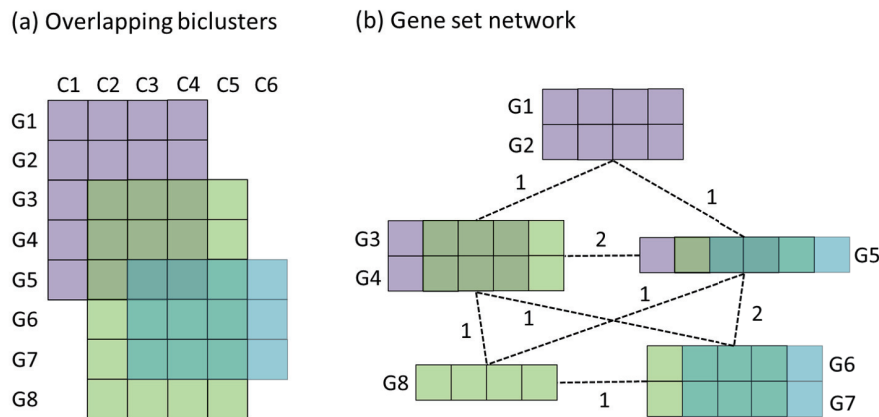


Fig. 3 Construction of a gene set network: (a) Three biclusters are overlapping with each other, for eight genes (G1 to G8) and six conditions (C1 to C6). (b) The three overlapping biclusters in (a) are converted into a graph with five nodes, according to the overlapping submatrices. Each node consists of genes shared by the same biclusters, resulting in a newly redefined bicluster with relevant conditions. If two nodes are from the same bicluster, they are connected by an edge weighted by the number of biclusters containing the genes of both nodes.

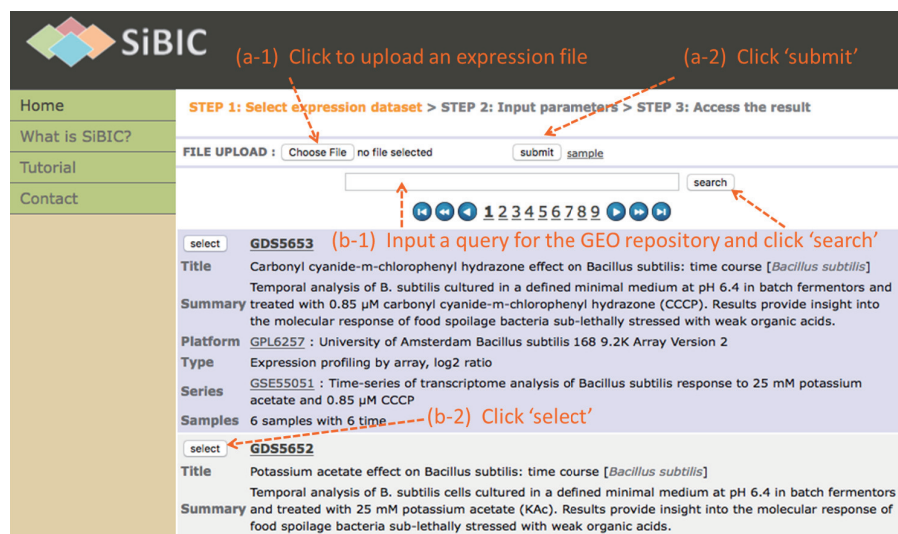


Fig. 4 The top page of SiBIC: A user can upload a local expression file or select a SOFT file from the GEO repository as the input of SiBIC.

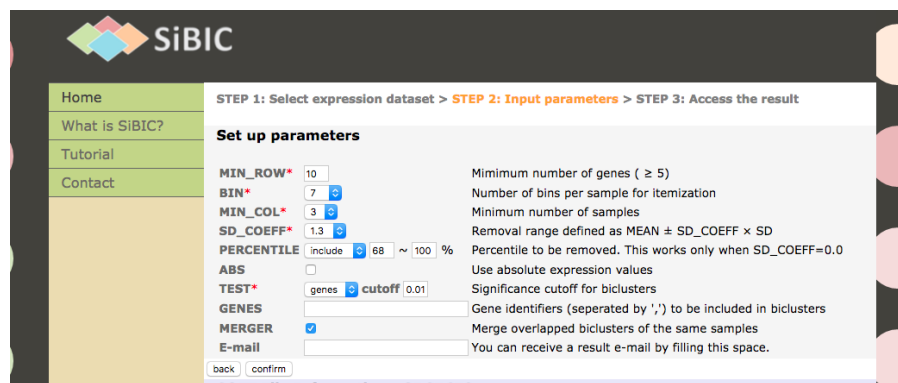
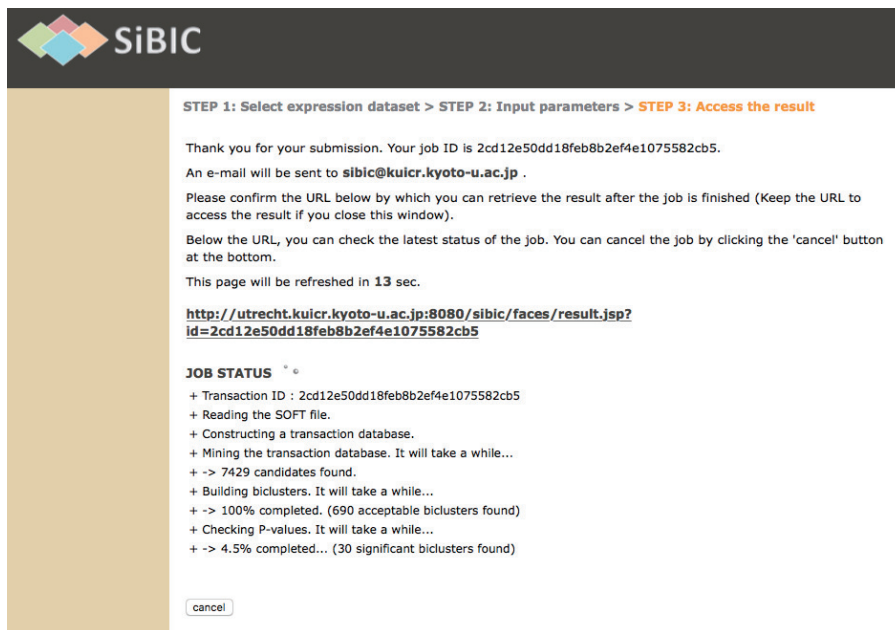


Fig. 5 The parameter setting page of SiBIC: After selecting a dataset, SiBIC leads the user to this page, where the user can specify a set of parameters such as MIN_ROW, BIN and MIN_COL.



STEP 1: Select expression dataset > STEP 2: Input parameters > STEP 3: Access the result

Thank you for your submission. Your job ID is 2cd12e50dd18feb8b2ef4e1075582cb5.

An e-mail will be sent to sibic@kuicr.kyoto-u.ac.jp.

Please confirm the URL below by which you can retrieve the result after the job is finished (Keep the URL to access the result if you close this window).

Below the URL, you can check the latest status of the job. You can cancel the job by clicking the 'cancel' button at the bottom.

This page will be refreshed in 13 sec.

<http://utrecht.kuicr.kyoto-u.ac.jp:8080/sibic/faces/result.jsp?id=2cd12e50dd18feb8b2ef4e1075582cb5>

JOB STATUS * ◦

- + Transaction ID : 2cd12e50dd18feb8b2ef4e1075582cb5
- + Reading the SOFT file.
- + Constructing a transaction database.
- + Mining the transaction database. It will take a while...
- + -> 7429 candidates found.
- + Building biclusters. It will take a while...
- + -> 100% completed. (690 acceptable biclusters found)
- + Checking P-values. It will take a while...
- + -> 4.5% completed... (30 significant biclusters found)

Fig. 6 The running status page of SiBIC: In this page, the user can check the running status and cancel the current job.

(a) Result page

Bicluster Information

Parameters
FILE: human
MIN_ROW: 10
BIN: 7
MIN_COL: 3
SD_COEFF: 2.0
TEST: genes 0.01
GENES: NOT SPECIFIED

Job ID: 3c2edf256b9b5d1ac3d109f9ebef583
Start: 10.5.2017 18:22:13
End: 10.5.2017 18:23:51
Runtime: 98 sec
Biclusters: 32 (MERGED)
Download: 600703681.DL.DX1

P-value Cutoff: 0.01 | Max Overlapping Rate: 100%
Top: 100 | Search

(A regular expression of identifiers can be entered)

ID	size	gene	cond	gene-P
UPID_1	60	10	6	< 2.0e-7
UPID_2	33	11	3	< 2.0e-7
UPID_3	30	10	3	< 2.0e-7
UPID_4	30	10	3	< 2.0e-7
UPID_5	40	10	4	< 2.0e-7
UPID_6	30	10	3	< 2.0e-7
UPID_7	36	12	3	< 2.0e-7
UPID_8	30	10	3	< 2.0e-7
UPID_9	33	11	3	< 2.0e-7
UPID_10	30	10	3	< 2.0e-7
UPID_11	30	10	3	< 2.0e-7
UPID_12	39	13	3	< 2.0e-7
UPID_13	30	10	3	< 2.0e-7
UPID_14	30	10	3	< 2.0e-7
UPID_15	30	10	3	< 2.0e-7
UPID_16	30	10	3	< 2.0e-7
UPID_17	30	10	3	< 2.0e-7
UPID_18	30	10	3	2.2519E-7
UPID_19	50	10	5	2.9064E-7
UPID_20	30	10	3	2.7023E-5

Gene Set Networks for Overlapping Biclusters

GSN file	Overlap	gene	vertex	edge
download	15	40	25	114
download	13	35	25	190
download	1	10	1	0
download	3	19	6	10

[Get the gene set network viewer](#)

(b) Bicluster page

Bicluster Charts

high color: yellow | low color: red | reverse

Bicluster Table

Simulation score (GENE): 0.0 | Simulation score (COND): Genes: 10 | Conditions: 6

Kit	ID	REF	NAME	ID	NAME	ID	NAME	ID	NAME	ID	NAME
[1]	IGH	GCL	4.096215	[1]	GCL	3.752715	[1]	GCL	4.420195	[2]	IGHG1
[1]	IGHA2	GCL	4.096215	[1]	GCL	3.752715	[1]	GCL	4.420195	[2]	IGHA2
[1]	IGHA1	GCL	2.1184	[1]	GCL	1.6962	[1]	GCL	2.062	[2]	IGHA1
[1]	IGHG1	GCL	4.096215	[1]	GCL	3.752715	[1]	GCL	4.420195	[2]	IGHG1
[1]	IGHG2	GCL	4.096215	[1]	GCL	3.752715	[1]	GCL	4.420195	[2]	IGHG2
[1]	IGHM	GCL	4.096215	[1]	GCL	3.752715	[1]	GCL	4.420195	[2]	IGHM
[0]	IGCC	IGCC	5.55887143	[0]	IGCC	7.781122857	[0]	IGCC	1.64357143	[1]	IGCC
[0]	LOC440871	LOC440871	5.55887143	[0]	LOC440871	7.781122857	[0]	LOC440871	7.32797420	[1]	LOC440871

Enrichment Analysis by DAVID

Query Type: GENE ID | Tool: Functional Annotation Chart | Gene Ontology: GOTERM_BP_ALL | Gene Ontology: GOTERM_BP_FAT

Fig. 7 The result pages: (a) The result page consists of a table of biclusters and a table of gene set networks. (b) The individual page of a bicluster shows a heatmap, a line chart and a numerical table. This page also has an interface of DAVID at the bottom.

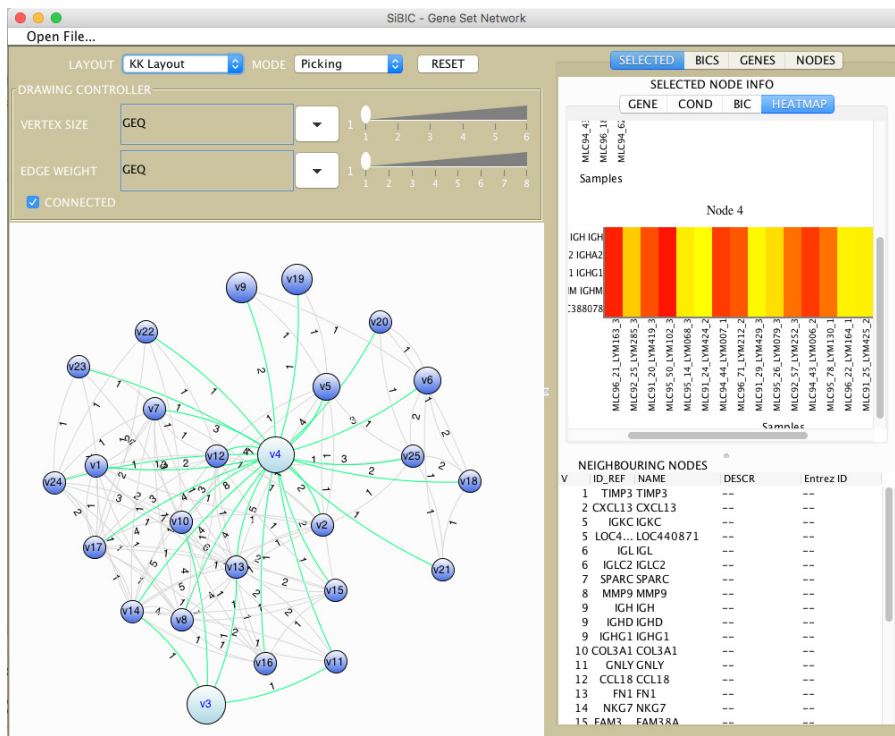


Fig. 8 Gene set network viewer has the left and right panes. The left pane shows a gene set network, and the right pane shows information on selected nodes in the network.

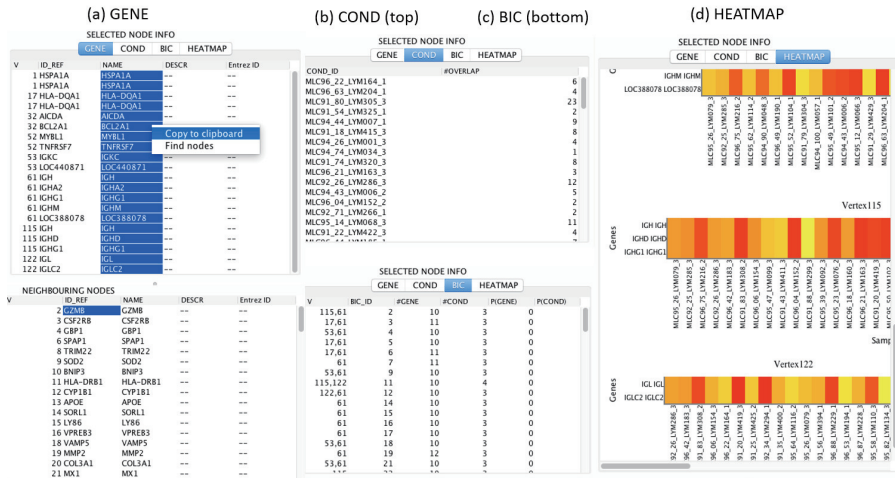


Fig. 9 By clicking 'SELECTED' in the right pane of Fig. 8, (a) GENE, (b) COND, (c) BIC and (d) HEATMAP are displayed in the pane. GENE and COND show the list of genes and conditions in the selected nodes, respectively. BIC shows the information on biclusters related to the selected nodes. HEATMAP shows a list of heatmaps of the selected nodes.