Bioinformatics, 33(16), 2017, 2523–2531
doi: 10.1093/bioinformatics/btx199
Advance Access Publication Date: 4 April 2017
Original Paper

Systems biology

Gracob: a novel graph-based constant-column biclustering method for mining growth phenotype data

Majed Alzahrani[1], Hiroyuki Kuwahara[1], Wei Wang[2] and Xin Gao[1,*]

[1]King Abdullah University of Science and Technology (KAUST), Computational Bioscience Research Center (CBRC), Computer, Electrical and Mathematical Sciences and Engineering (CEMCE) Division, Thuwal, 23955-6900, Saudi Arabia and [2]Department of Computer Science, University of California, Los Angeles, CA 90095, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on December 2, 2016; revised on February 17, 2017; editorial decision on March 31, 2017; accepted on April 3, 2017

Abstract

Motivation: Growth phenotype profiling of genome-wide gene-deletion strains over stress conditions can offer a clear picture that the essentiality of genes depends on environmental conditions. Systematically identifying groups of genes from such high-throughput data that share similar patterns of conditional essentiality and dispensability under various environmental conditions can elucidate how genetic interactions of the growth phenotype are regulated in response to the environment.

Results: We first demonstrate that detecting such ‘co-fit’ gene groups can be cast as a less well-studied problem in biclustering, i.e. constant-column biclustering. Despite significant advances in biclustering techniques, very few were designed for mining in growth phenotype data. Here, we propose Gracob, a novel, efficient graph-based method that casts and solves the constant-column biclustering problem as a maximal clique finding problem in a multipartite graph. We compared Gracob with a large collection of widely used biclustering methods that cover different types of algorithms designed to detect different types of biclusters. Gracob showed superior performance on finding co-fit genes over all the existing methods on both a variety of synthetic data sets with a wide range of settings, and three real growth phenotype datasets for *E. coli*, proteobacteria and yeast.

Availability and Implementation: Our program is freely available for download at http://sfb.kaust.edu.sa/Pages/Software.aspx.

Contact: xin.gao@kaust.edu.sa

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Under a standard lab condition, a vast majority of genes have little to no effect on the normal growth of microorganisms (Korona, 2011). These so-called ‘dispensable’ genes account for over 90% in *E.coli* and *B.subtilis* (Baba *et al.*, 2006; Kobayashi *et al.*, 2003), while over 80% in yeast (Giaever *et al.*, 2002; Kim *et al.*, 2010). A molecular-network level understanding of the cause of this gene dispensability has important implications in evolution and systems biology (Bochner, 2009).

One theory to explain this phenomenon is mutational robustness, which argues that these genes are dispensable because the genetic architecture has evolved to compensate for gene mutations either by duplicate genes or by backup pathways (Gu *et al.*, 2003; Wagner, 2000). Another theory is environment-dependent genetic interaction, which argues that these seemingly dispensable genes are actually essential in other environments as the activation of genetic interactions depends on environmental conditions (Papp *et al.*, 2004). Whereas both theories could explain dispensable genes, the

© The Author 2017. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

latter was shown to provide explanations for a majority of dispensable genes in yeast (Hillenmeyer *et al.*, 2008). To advance our knowledge of environment-dependent genetic interactions, one key question to address is how to find *co-fit genes*, which are defined to be a group of genes that share similar patterns of conditional essentiality and dispensability across various environmental conditions (Deutschbauer *et al.*, 2014; Hillenmeyer *et al.*, 2010). The illustration in Figure 1(a) shows how similar phenotype patterns could help reveal the underlying organization of the genetic interactions.

The recent development in genome-wide growth-phenotype (i.e. fitness) profiling methods enabled the measurement of fitness scores of a large number of gene-deletion strains over many stress conditions (Bochner, 2009; Giaever *et al.*, 2002; Hillenmeyer *et al.*, 2008; Nichols *et al.*, 2011). Importantly, such growth phenotype data can be used to assess the effects of a loss-of-function mutation of each gene on fitness and detect which genes are essential and dispensable under different stress conditions. That is, for a given environmental condition, conditionally essential genes are defined to be those whose loss-of-function mutations have very low fitness values, while conditionally dispensable genes are defined to be those whose loss-of-function mutations have very high fitness values. Thus, we can use such growth phenotype data to systematically identify sets of co-fit genes, allowing us to probe how the genetic interactions are organized and how environmental conditions can change the genetic interactions. Such environment-dependent genetic interactions have been commonly analyzed using flux balance analysis (e.g. Harrison *et al.*, 2007; Papp *et al.*, 2004; Segrè *et al.*, 2005). While flux balance analysis is a powerful method that can predict how metabolic activities may change given various environmental and genetic perturbations, its accuracy depends on prior knowledge about the structure of a given metabolic system and metabolic flux boundaries. Here, we propose an alternative, data-driven approach that can be used for analysis of environment-dependent genetic interactions. In this approach, by representing a growth phenotype dataset by a two-dimensional matrix, whose rows are the gene-deletion strains and columns are the stress conditions, we transform a problem of finding sets of co-fit genes into a constant-column biclustering problem (as illustrated in Fig. 1(b)).

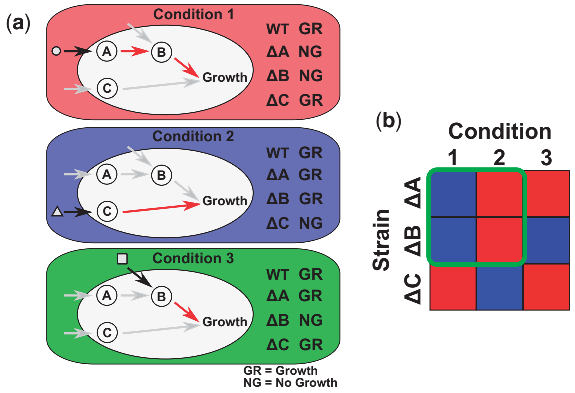

Fig. 1. A minimalist example to illustrate environment-dependent genetic interactions. (**a**): Conditionally essential and dispensable genes. The circle, triangle and square symbols illustrate environmental inputs to the cell, for example input metabolites and ligands. Red, gray and black arrows denote active paths in wild type, inactive paths and active paths in each condition, respectively. The wild type grows normally under each condition, while the deletion of each gene has different effects on fitness under different conditions. ΔX denotes the strain of deleting gene X ($X \in \{A, B, C\}$). 'GR' and 'NG' stand for normal growth and no growth, respectively. (**b**): The corresponding growth phenotype data. Blue and red denote low and high fitness, respectively. The constant-column bicluster in the green box captures co-fit genes, *A* and *B*, which cannot be captured by any other constant biclusters

We argue that in growth phenotype data, finding constant-column biclusters results in detecting more meaningful biclusters, i.e. co-fit genes. There is a fundamental difference in the nature of growth phenotype data and gene expression data, the latter of which was the target for almost all existing biclustering methods. In gene expression data, each row (i.e. a gene) has a reference value, which is the expression level of this gene under the normal condition. Thus, the reference values for different rows are different from each other. Although data normalization or transpose can be done to transform the problem of mining gene expression data into the constant-column biclustering problem, mining other types of biclusters, e.g. constant biclusters or coherent biclusters, is more prevalent in mining gene expression data. In contrast, in growth phenotype data, all rows (i.e. strains) have a same reference value, which is the growth of the wild type (without any knock-out) under the normal condition. Thus, detecting constant-column biclusters in such data can identify co-fit genes because such a bicluster implies the deletion of this group of genes has similar effects on fitness (i.e. similar values in the same column imply similar changes to the reference value) under a subset of stress conditions (as illustrated in Fig. 1).

This motivated us to develop a novel biclustering method, Gracob, that is designed to identify constant-column biclusters in growth phenotype datasets. To our knowledge, this is the first work that develops and applies biclustering methods to mining co-fit genes in growth phenotype data. The identification of co-fit genes by such a method can be useful for gaining new insights into the functional organization of genes, which has been commonly analyzed using the pairwise correlation coefficient across all the conditions considered in an experimental setup. This is because a co-fit gene measure can detect a significant local fitness similarity under a subset of conditions, while such strong signals can be diluted in the overall correlation coefficient measure owing to the rest of the conditions.

We compared Gracob with 13 representative widely used methods that cover a wide spectrum of algorithms and types of biclusters they can detect. When evaluated on a variety of synthetic datasets, Gracob showed nearly perfect performance with respect to different noise levels and overlapping degrees. We then applied Gracob to three real growth phenotype datasets for *E. coli*, proteobacteria and yeast. Gracob was able to identify maximal constant-column biclusters while the existing methods failed to do so. Functional enrichment analysis through KEGG pathways and GO terms demonstrated that Gracob is on average more than twice as precise as the other methods.

2 Related work

2.1 Previous biclustering methods

The biclustering problem was first proposed by Cheng and Church (2000) to analyze gene expression data. Since then, extensive efforts have been made in both computer science and statistics to develop different types of biclustering methods (e.g. Bergmann *et al.* 2003; Ben-Dor *et al.*, 2003; Bozdağ *et al.*, 2009; Cho *et al.*, 2004; Gu and Liu, 2008; Gusenleitner *et al.*, 2012; Henriques and Madeira, 2014, 2015; Hochreiter *et al.*, 2010; Huttenhower *et al.*, 2009; Kluger *et al.* 2003; Lazzeroni and Owen, 2002; Liu and Wang, 2003; Li *et al.*, 2009; Murali and Kasif, 2003; Pandey *et al.*, 2009; Prelić *et al.*, 2006; Serin and Vingron, 2011; Sheng *et al.*, 2003; Tanay *et al.*, 2002, 2004; Turner *et al.*, 2005; Yang *et al.*, 2002, 2003; Wang *et al.*, 2002).

Existing methods mainly deal with three types of biclusters (Madeira and Oliveira, 2004), i.e. constant biclusters within which the variation is low, constant-column (or constant-row) biclusters within which the column-wise (or the row-wise) variation is low, and coherent biclusters in which the data generally follow an additive or a multiplicative model. As shown in Figure 1(b), the problem of finding co-fit genes is equivalent to finding constant-column biclusters in a growth phenotype data matrix. That is, we are interested in finding a group of genes that, under multiple conditions, have similar fitness to each other. However, despite the success of the existing biclustering methods to analyze gene expression data, to the best of our knowledge, there are no other studies that developed and applied biclustering methods to mining growth phenotype data.

Here, we review 13 biclustering methods that are widely used in various comparative studies (Eren *et al.*, 2013; Henriques *et al.*, 2015; Prelić *et al.*, 2006), which will be later compared with our method on both synthetic datasets and real growth phenotype datasets. These methods are CC (Cheng and Church, 2000), Plaid (Lazzeroni and Owen, 2002; Turner *et al.*, 2005), FLOC (Yang *et al.*, 2003), ISA (Bergmann *et al.*, 2003), xMOTIFs (Murali and Kasif, 2003), Spectral (Kluger *et al.*, 2003), SAMBA (Tanay *et al.*, 2004), Bimax (Prelić *et al.*, 2006), BBC (Gu and Liu, 2008), QUBIC (Li *et al.*, 2009), CPB (Bozdağ *et al.*, 2009), iBBiG (Gusenleitner *et al.*, 2012) and BicPAM (Henriques and Madeira, 2014). Since most of the existing methods used different definitions of biclusters and were reported to be general as they are not restricted to certain types of data, it is difficult to clearly categorize them.

Here we first group these biclustering methods according to the general types of biclusters such methods used for evaluation in their papers or in comparative studies. A typical class of the existing methods work with 'constant' biclusters. Here constant is often defined to be the same value after discretizing the input data matrix into 0's and 1's (e.g. Bimax and iBBiG). Another major class of the existing methods have their own definitions of the biclusters they are looking for, which do not directly correspond to constant-, constant-column-, or coherent-biclusters. For example, CC uses the mean squared residue to define a bicluster, which basically measures the variance of the individual data points in the biclusters with respect to the mean of the corresponding rows, the corresponding columns and the entire bicluster. Plaid models the data matrix as a sum of layers and minimizes the fitting error through optimization. Similarly, BBC uses the plaid model of biclusters which defines a bicluster as a combination of the main effect, the gene effect, the condition effect and the noise. FLOC extends the CC algorithm by using a probabilistic model to account for missing values in data. ISA requests that the mean value of each row must be higher than a threshold, and so does each column. CPB defines the biclusters in a similar way, i.e. the Pearson correlation coefficient between columns and rows must be higher than a threshold. Spectral tries to detect checkerboard structures. Therefore, this class of methods can theoretically detect different types of biclusters. A number of methods were developed to (preferably) detect constant-column (or equivalently constant-row) biclusters. SAMBA discretizes the data into different bins and finds biclusters with each column belonging to the same bin. Similarly, xMOTIFs attempts to find biclusters within each of which genes have the same state under different samples. The method picks up randomly sampled subsets over the conditions and chooses the corresponding subsets of genes that satisfy this requirement. However, when the number of conditions is large, the chance of picking the proper subsets of conditions becomes very low. QUBIC thresholds the extreme values (both positive and negative) and detects constant-column and constant-row biclusters on the discretized values only. Recently, BicPAM was proposed to detect both additive and multiplicative coherent biclusters.

In terms of the techniques such methods use, they can be classified into iterative methods (i.e. CC, ISA, Bimax, CPB, Plaid, FLOC and iBBiG), matrix decomposition-based methods (i.e. ISA and Spectral), graph-based methods (i.e. SAMBA and QUBIC), sampling-based methods (i.e. xMOTIFs and BBC) and pattern mining-based methods (i.e. BicPAM). The iterative methods either gradually grow biclusters from small seeds, or delete columns or rows that cannot be a part of the biclusters from the original matrix. The decomposition-based methods mainly use different variants of singular value decomposition to reduce the dimensionality in order to better detect biclusters. The graph-based methods model the problem in a bipartite graph, and look for cliques or densely connected subgraphs. The sampling-based methods try to control the way of sampling to increase the probability of finding large biclusters. The pattern mining-based methods rely on frequent itemset mining or association rules to identify biclusters.

2.2 Co-fitness measurement with constant-column biclustering

Co-fit genes are traditionally defined using the pairwise correlation coefficient of two genes across all the stress conditions, and hierarchical clustering is used to group co-fit genes together (Deutschbauer *et al.*, 2011; Hillenmeyer *et al.*, 2008; Nichols *et al.*, 2011). However, as mentioned in Section 1, the use of correlation coefficient to measure similarity could miss strong signals detected in a subset of conditions owing to 'correlation dilution' through the rest of the conditions. To further elucidate this, let us take the genes LSM2 and LSM3 of *Saccharomyces cerevisiae* (Hillenmeyer *et al.*, 2008) as an example. These two genes have a low correlation value, $r = 0.15$, although they share many common functions and high sequence similarity. Both genes are part of one complex that binds to the 3′ end of U6 snRNA, and are responsible for its regulation and stability (Pannone *et al.*, 2001). LSM2 and LSM3 are required for pre-mRNA splicing and their mutations inhibit mRNA decapping (Tharun *et al.*, 2000). Another study showed that LSM2 and LSM3 form many interactions with each other (Mayes *et al.*, 1999). The semantic similarity between their cellular component GO terms is 0.95 as calculated using Wang *et al.* (2007). Thus, these two genes are in the same functional organization by definition. However, the correlation coefficient measure cannot capture this. Our method, on the other hand, predicted them as co-fit genes since they were in the same constant-column bicluster based on similar fitness values representing conditional essentiality or dispensability. Specifically, our method detected similar, extreme fitness values between the LSM2- and LSM3-deletion strains for 51 out of 726 different stress conditions in the yeast phenotype profiling data, showing statistically significant association (P-value $= 3.0 \times 10^{-6}$), and these deletion strains have very high correlation ($r = 0.99$) over these 51 conditions.

Therefore, the co-fitness can only be detected by local measures as they capture the similarity over a subset of conditions. Furthermore, by using biclustering methods to find co-fit genes, it is possible to explicitly identify which subset of genes shares similar patterns of conditional essentiality and dispensability under which subset of stress conditions. By definition of co-fitness, a bicluster of co-fit genes should have similar values in each column of this bicluster, but values across different columns can be very different, which is the same definition as constant-column biclusters.

3 Materials and methods

3.1 Gracob

The proposed method, GRAph-based Constant-cOlumn Biclustering (Gracob), is a deterministic graph-based method that is designed to find maximal constant-column biclusters in any given data matrix (Fig. 2), where a maximal bicluster means that it is not possible to extend the bicluster by either rows or columns while keeping the same level of specified similarity. Although most interesting variants of the biclustering problems are well known to be NP-Complete (Tanay *et al.*, 2002), the proposed method takes advantage of the sparsity of biclusters. That is, compared to the size of the input data matrix, the number of biclusters in the matrix is small. For the sake of simplicity, here, we define that each row represents a gene-deletion strain and each column represents a condition.

The main idea of Gracob is that once the users define how ‘constant’ they want the biclusters to be column-wise in the preprocessed data (Fig. 2(a)), Gracob looks at the subsets of strains that maximally satisfy this ‘constant’ requirement inside each independently sorted column. Each of such subsets is defined to be a *block*, which is a multi-row one-column vector in the corresponding sorted column. Consequently, any column in any potential bicluster is contained by at least one of these blocks (Fig. 2(b) and (c)). We then build a multipartite graph in which each node is a block and an edge is created between two blocks from two different conditions if they share a sufficient number of strains (Fig. 2(d)). This number is defined to be the minimum number of strains in a desired bicluster. For instance, if this number is set to be 1, then every single strain constitutes a constant-column bicluster by definition; however, such biclusters are most likely not of interest to the users. If there is a bicluster of n conditions, there must exist in the graph a clique of m ($m \geq n$) nodes that contain these n blocks (Fig. 2(e)). The problem then becomes finding maximal cliques in this multipartite graph. We propose an efficient method to solve this problem. The idea is to divide the problem into smaller ones, and make use of the characteristics of the data and the requirements of biclusters to search for solutions in a reasonable amount of time (Fig. 2(f)). Finally, biclusters are identified inside these cliques (Fig. 2(g)).

Here, we start by formulating the problem and then briefly describing the main steps. The technical details can be found in Supplementary Materials. Gracob consists of three main phases: (i) the pre-processing phase, (ii) the graph creation phase and (iii) the maximal clique finding phase.

3.1.1 Problem formulation

Let G be a set of n mutant strains, each of which is a single gene knock-out mutation, and C be a set of m environmental stress conditions. We denote a_{ij} as the elements of the growth phenotype data matrix $A_{(n\times m)}$ where a_{ij} is a real value that represents the growth of the ith mutant under the jth stress condition where $i \leq n$ and $j \leq m$.

To define a constant-column bicluster, the user has to specify three parameters. The first one is the range threshold, δ, to define how ‘constant’ each column is in desired biclusters. For example, if δ is set to be 0, then the user is looking for biclusters within which each column contains data with exactly the same value. The second one is the row threshold, r, to define the minimum number of strains (or genes) that each bicluster must have. If r is set to be 1, each row becomes a trivial constant-column bicluster because each column for the same row has 0 variance. The third one is the column threshold, c, to define the minimum number of conditions each desired bicluster must contain. If c is set to be 1, the biclusters will be a part of a single column, but is not an interesting one.

Once the requirements are provided by the user, let $I \subseteq G$ and $J \subseteq C$, we say that I and J specify a desired constant-column bicluster if the following conditions are satisfied:

$$|f(a_{i_1 j}) - f(a_{i_2 j})| \leq \delta, \tag{1}$$

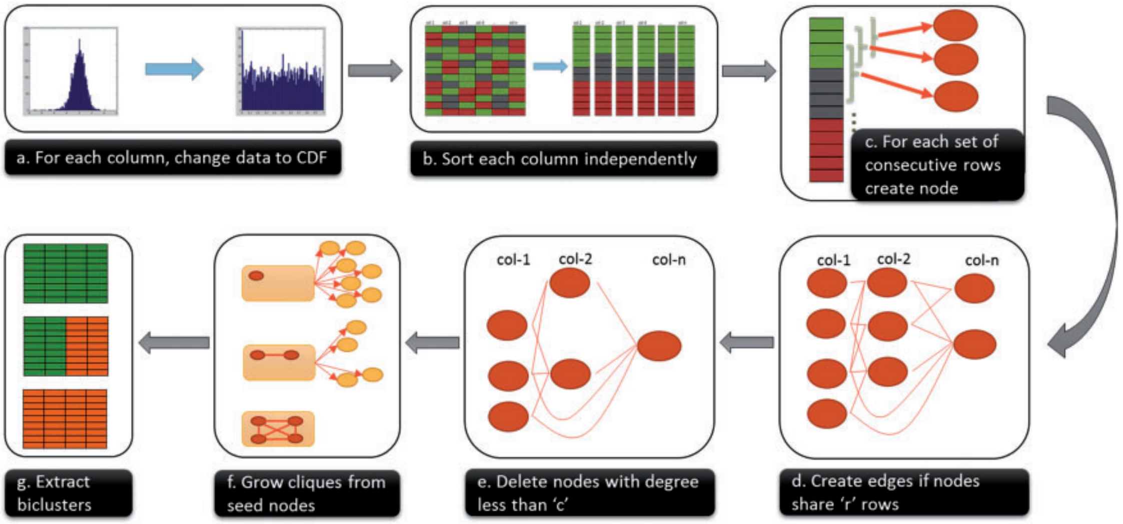
a. For each column, change data to CDF
b. Sort each column independently
c. For each set of consecutive rows create node
d. Create edges if nodes share ‘r’ rows
e. Delete nodes with degree less than ‘c’
f. Grow cliques from seed nodes
g. Extract biclusters
col-1
col-2
col-n

Fig. 2. Workflow of Gracob. (**a**): The data in each column are transformed using a cumulative distribution function, independently. (**b**): Data values in each column are sorted independently from other columns while keeping track of the original row indexes. (**c**): Nodes are created for each consecutive row subset such that the range of their values is at most δ (user defined value for how ‘constant’ each column of desired biclusters should be). A row subset can overlap with other row subsets but cannot be contained by others. (**d**): An edge is created between any pair of nodes if the nodes are from different columns and share at least r (user defined threshold for the smallest number of strains in desired biclusters) rows (i.e. strains). (**e**): Nodes with degree less than c (user defined threshold for the smallest number of conditions in desired biclusters) are deleted from the graph. (**f**): Each node is used to grow a clique with its connected nodes (orange circles) while thresholds, r and c, are repeatedly checked to detect future failures as early as possible. (**g**): Row and column index information from each clique is used to extract biclusters from the original data matrix

$$|I| \geq r, \tag{2}$$

$$|J| \geq c, \tag{3}$$

where $i_1, i_2 \in I$ and $j \in J$. $|.|$ denotes the cardinality of a set. $f(.)$ is a transformation function that we will specify later. Eq. (1) ensures that the values within each column of the bicluster are similar, whereas Eq. (2) and (3) make sure only non-trivial biclusters are reported. The objective is to find all I and J that satisfy these conditions, and there is no I' and J' such that $I \subseteq I'$ and $J \subseteq J'$ that satisfies these conditions, i.e. only maximal constant-column biclusters are returned.

3.1.2 Preprocessing phase

There are two major steps in this phase, i.e. transforming the data in each condition based on a cumulative distribution and creating blocks (nodes). The input growth phenotype data are often assumed to follow a standard normal distribution where the data have been z-score normalized inside each column (Nichols *et al.*, 2011). As most of the outlier data points are distributed along a long range of values, they are considered to show similar phenotypes, i.e. growth is extremely sensitive (negative outliers) or stable (positive outliers) with respect to environment conditions. Thus, there is a need to transform the data into another space which preserves the similarity of these values. We chose to apply CDF (cumulative distribution function) transformation to each column, independently, in the input matrix. Consequently, data points in the tail of each side will be assigned very close values, which satisfies our needs. The right panel of Figure 2(a) shows the distribution of the values for a column after the CDF transformation.

The second step is to create blocks that are the nodes for the multipartite graph. The idea is to sort (Fig. 2(b)) and then linearly scan each column to get all the blocks in which the range of values is at most δ. These blocks are used as the (unit) nodes for the following phases (Fig. 2(c)). Detailed steps can be found in Section S1.2.

3.1.3 Graph creation phase

In this phase we create edges between the blocks (unit nodes). The edges are not weighted but rather labeled by the shared subsets of strains. There is no edge created between nodes from the same condition, and the cardinality of the shared subset of an edge must be at least r. The complexity of such a process is $O(S^2)$ where S is the total number of nodes. With genome-wide growth phenotype data, S can be in the order of millions and $O(S^2)$ runtime becomes infeasible. Therefore, we propose a divide-and-conquer algorithm by repeatedly using the user defined thresholds c and r to reduce the search space, and thus reduce the practical runtime. The main idea is that we first merge all the blocks inside each column into a super-node and create edges among all these super-nodes. Then we try to divide these super-nodes into non-overlapping child nodes, each of which is a subset of blocks and inherits the edges from its parent node, unless the cardinality (i.e. number of genes) of the edge is below r, which means this edge will never be a part of a meaningful bicluster. If such a non-overlapping split is not feasible, then we split in the middle. Meanwhile, we delete all the nodes that have degree below c, which means the blocks in this nodes will never be a part of bicluster with at least c stress conditions. We recursively do this splitting until each node is a block. The detailed steps can be found in Section S1.3.

Note that although the divide-and-conquer idea has been used in biclustering methods, such as in Bimax (Prelić *et al.*, 2006), our divide-and-conquer algorithm is very different from the ones developed in literature.

3.1.4 Maximal clique finding phase

The objective of this phase is to find and return all maximal cliques, from which biclusters can be easily extracted. Unfortunately, existing general-purpose maximal clique finding algorithms do not scale up well on our problem. We thus propose a tailored algorithm that starts from each remaining unit node from the previous phase, and sequentially grows cliques seeded from this node by gradually adding connected nodes to the existing cliques. The main idea is to use the minimum row and column thresholds, r and c, to detect future failures as early as possible and to eliminate those cliques that have no hope to grow to the required size. The details of our algorithm can be found in Section S1.4.

It is noteworthy that Gracob is an exhaustive algorithm that finds all maximal biclusters in the given growth phenotype dataset, under the given thresholds, δ, r and c. Neither the divide-and-conquer algorithm used in the graph creation phase nor the early detection of failures trick used in the maximal clique finding phase affects the optimality of the search.

3.2 Validating Gracob on synthetic data

Following Prelić *et al.* (2006); Li *et al.* (2009); Eren *et al.* (2013), we validated Gracob by a variety of synthetic datasets, where different types of implanted biclusters, different levels of noise, and different degrees of bicluster overlaps were simulated (Section S2.1). Since the ground-truth biclusters are known for the synthetic datasets, we used recall, precision and F1-score to measure the performance (Section S2.2).

Our results (Supplementary Fig. S1) show that among the 14 compared methods, ISA, QUBIC, and Gracob are all able to detect both constant biclusters and constant-column biclusters well. These three methods can also tolerate noise. However, when the overlapping degree of the implanted biclusters is high, Gracob is the only method that can almost perfectly identify all the implanted biclusters (Section S2.3).

We conducted the sensitivity analysis on Gracob with respect to the parameters r (minimum number of rows for biclusters) and c (minimum number of columns), and Gracob shows strong robustness to these parameters (Supplementary Fig. S1(9a)). We further tested the three best performing methods with respect to the increasing size of the input data matrix. In terms of F1-score, Gracob is very stable whereas ISA and QUBIC are less (Supplementary Fig. S1(9b)). In terms of the runtime, Gracob has a similar runtime to QUBIC, while both are faster than ISA (Supplementary Fig. S1(9c)).

3.3 Growth phenotype data

To comprehensively evaluate the performance of Gracob, we used three recently measured growth/fitness phenotype datasets.

The first one is the genome-wide growth phenotype dataset of *E. coli* (Nichols *et al.*, 2011). This dataset consists of fitness data for 3979 mutant strains, each of which was measured under 324 different stress conditions. Each fitness value in the data matrix represents the relative growth rate of a given gene-knockout strain under a given stress condition, which is normalized column-wise to follow the unit normal distribution (Nichols *et al.*, 2011). Figure 3(1) shows this growth phenotype dataset.

The second one is the DNA tag-based pooled fitness assay dataset for *Shewanella oneidensis* MR-1, a Gram-negative γ-proteobacterium (Deutschbauer *et al.*, 2011). The dataset contains the mutant fitness for 3355 nonessential genes under the 195 pool fitness experiments.

The third one is the growth response dataset for *Saccharomyces cerevisiae* (Hillenmeyer *et al.*, 2008). The dataset contains 5337 heterozygous gene deletion strains over 726 conditions.

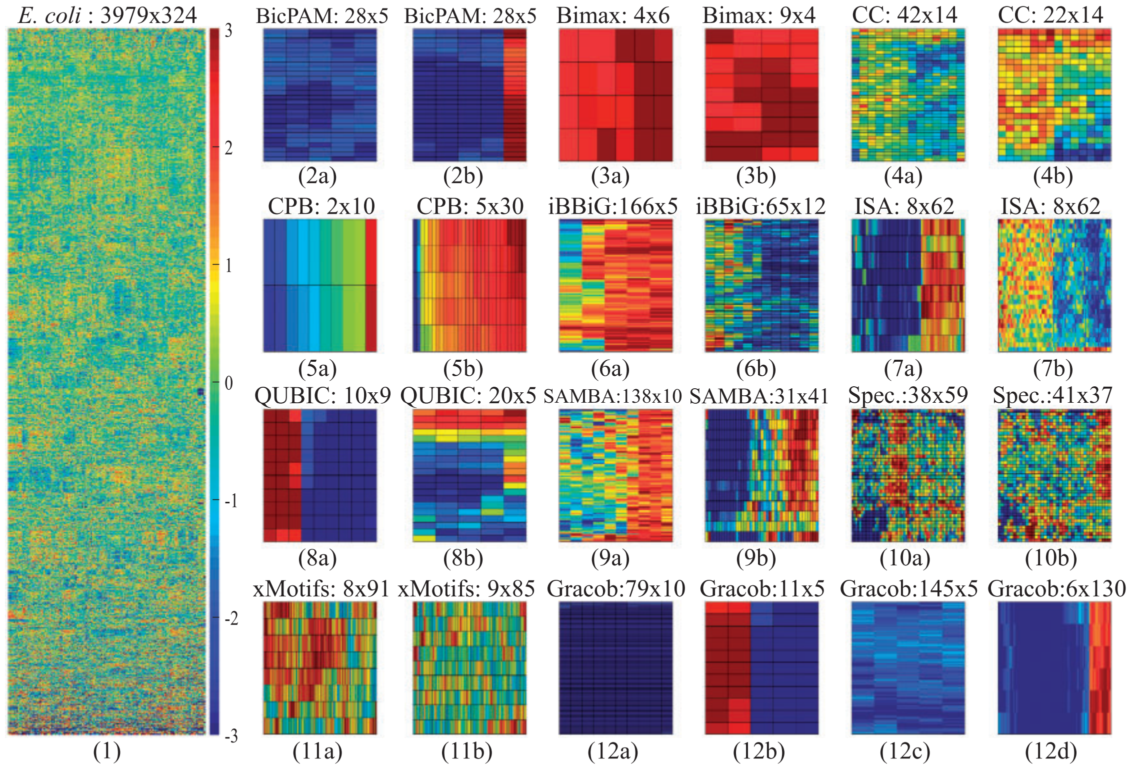

Fig. 3. Heatmap visualization of the *E. coli* growth phenotype data and the representative biclusters detected by the 11 methods. (1): The capped data matrix for the *E. coli* growth phenotype dataset with 3979 strains and 324 stress conditions. All the values bigger than 3.0 are capped as 3.0 and all the values smaller than -3.0 are capped as -3.0, for visualization purposes. (**2**)–(**12**): The representative biclusters detected by BicPAM, Bimax, CC, CPB, iBBiG, ISA, QUBIC, SAMBA, Spectral, xMOTIFs and Gracob, respectively. For each method, the predicted biclusters that have consistent patterns which appear many times in the results of this method are selected. For visualization purposes, rows and columns of each bicluster are organized by hierarchical clustering (Eisen *et al.*, 1998). That is, genes with similar values are clustered on the Y-axis and conditions with similar values are clustered on the X-axis

3.4 Performance measures

The real growth phenotype data do not have known ground-truth biclusters. Thus, to measure the performance of biclustering methods on the real data, we defined four performance measures. Since each biclustering method can discover a large number of biclusters in a given dataset, our measures consider the performance based on multiple biclusters. If the number of predicted biclusters is smaller than 100, we keep all of them. Otherwise, we keep the top 100 largest biclusters for evaluation. In order to reduce the bias caused by highly overlapping biclusters in evaluation, we sorted all the returned biclusters by their size in a descending order. We then applied a greedy approach to go down the list and keep only the biclusters that share less than 30% of the size of this bicluster with any previously selected bicluster, until we selected 100 biclusters.

The first measure is the average column-wise standard deviation. We first calculated the mean of the column-wise standard deviation for each bicluster, and then calculated the average of this value over all the predicted biclusters. The second measure is the average size of the predicted biclusters, where the size of a bicluster is measured by the number of rows times the number of columns. Thus, a method that simultaneously reports a small average standard deviation and a large average bicluster size is considered to be useful.

Furthermore, each bicluster is subject to two enrichment analyses, using pathway information from the KEGG database (Kanehisa and Goto, 2000) and gene ontology (GO) terms, respectively. For each of the predicted biclusters of a method, we found the set of genes that correspond to the strains of this bicluster, and searched for all the annotated pathways that contain at least one gene from this gene set. Then, we calculated the probability, i.e. *P*-value, of randomly finding these genes for each pathway with the hypergeometric calculation (Li *et al.*, 2009). The precision of a method is the ratio of biclusters which have at least one significant pathways (i.e. *P*-value smaller than a given threshold, e.g. 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4} or 10^{-3}) to the total number of selected biclusters for that method. The number of selected biclusters for any method is at most 100 as explained above. The same procedure was done for the GO term enrichment analysis, and the GO-level precision for different methods is reported as the fourth measure.

4 Results and discussions

We compared Gracob with the 13 representative widely used biclustering methods introduced in the related work. For each experiment, the input data were transformed and preprocessed following the requirements of different methods. The parameter settings for the 13 methods were searched and optimized based on the recommended use from their papers.

4.1 Performance on growth phenotype datasets

Some representative biclusters predicted by 11 methods on the *E.coli* dataset are illustrated in Figure 3(2)-(12). BBC and FLOC failed to detect any bicluster on these large growth phenotype datasets in 3 hours, and Plaid only predicted less than three biclusters and thus is not included in the analysis for the sake of fair comparison. It is clear that the biclusters detected by Bimax are ‘purely’

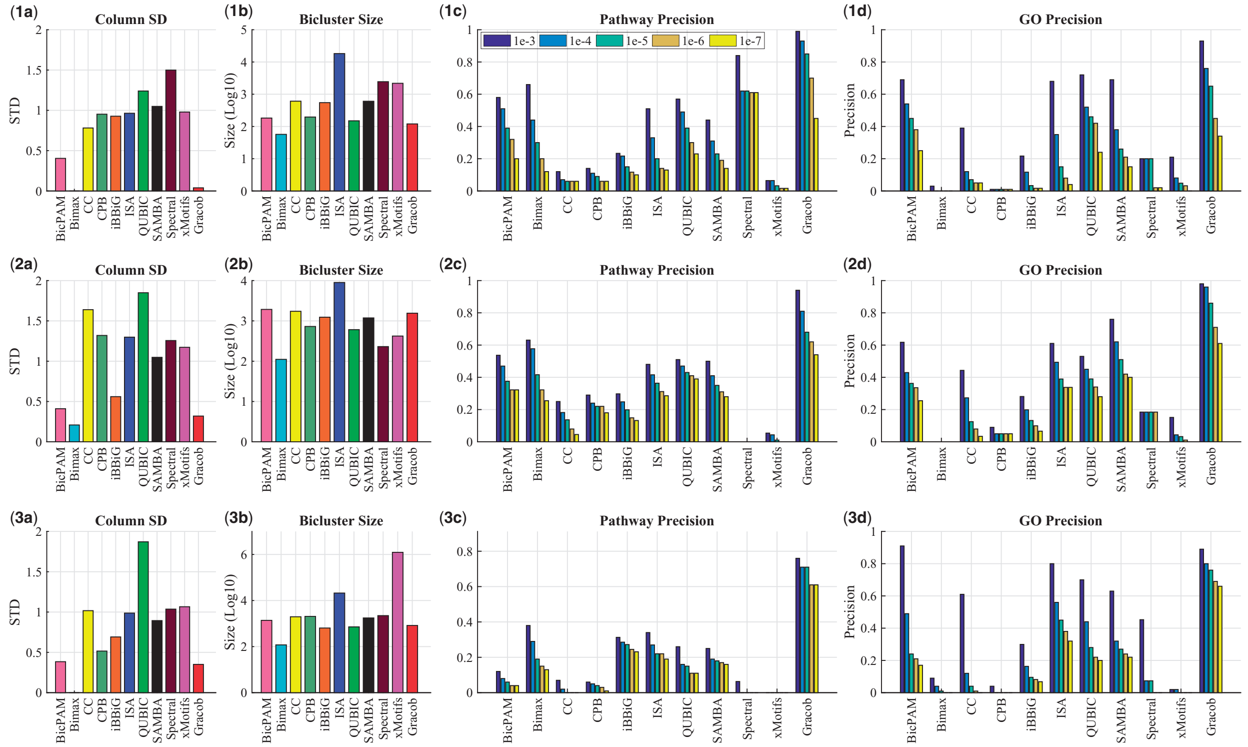

Fig. 4. Performance comparison of the 11 methods on the *E. coli*, proteobacteria and yeast growth phenotype datasets. (**1a**)–(**3a**): The average column-wise standard deviation on the three datasets, respectively. (**1b**)–(**3b**): The average size of the returned biclusters on the three datasets, respectively. (**1c**)–(**3c**): The KEGG pathway-level precision under five significance levels on the three datasets, respectively. (**1d**)–(**3d**): The GO term-level precision under five significance levels on the three datasets, respectively

constant, whereas the ones detected by CPB, iBBiG, ISA and SAMBA tend to have relatively constant columns, although they are still far less constant than the ones detected by BicPAM and Gracob. Among these four methods, CPB and iBBiG have relatively lower column-wise standard deviation, whereas ISA and SAMBA tend to detect bigger biclusters. It is worth noting that biclusters predicted by Bimax are not only smaller than those predicted by Gracob, but they also contain only large positive values. This is due to the required binary discretization step in Bimax. Among the biclusters returned by Gracob, about 62% consists of only conditionally essential genes (i.e. biclusters in the blue color), 20% consists of only conditionally dispensable genes (i.e. biclusters in the red color), and 18% consists of genes that are essential under certain conditions but dispensable under some other conditions (i.e. biclusters with mixed colors).

In terms of the average column-wise standard deviation, as expected, Bimax and Gracob have the lowest column-wise variance, followed by BicPAM (Fig. 4(1a)-(3a)). However, the average bicluster size of Gracob is one order of magnitude bigger than that of Bimax (Fig. 4(1b)-(3b)). Although ISA, Spectral and xMOTIFs can return large biclusters, they are very impure. Overall, Gracob has a remarkably strong ability to discover maximal constant-column biclusters. As shown in Figure 4(1c)-(3c), Gracob has the highest percentage of significantly enriched KEGG pathways among all the 11 methods, under almost all the different significance levels. The only exception is for the *E. coli* dataset, when the significance threshold is below 1E-7, the precision of Gracob is slightly lower than that of Spectral. The average precision of Gracob under the five significance thresholds (10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} and 10^{-7}) are 0.90, 0.82, 0.75, 0.64 and 0.53, respectively, whereas that of the second best method are 0.56 (Bimax), 0.44 (Bimax), 0.32 (QUBIC), 0.27 (QUBIC) and 0.24 (QUBIC), respectively. These results show that for this analysis Gracob is at least 61%, 86%, 134%, 137% and 121% more precise than any other biclustering method in terms of KEGG pathways under the five significance levels, respectively.

Similar conclusions can be drawn on the GO term-level precision. Gracob is clearly more precise than all the other methods under almost all the situations (Fig. 4(1d)-(3d)), except for the yeast data when the significance level is 10^{-3}, the GO-level precision of Gracob (0.89) is slightly lower than that of BicPAM (0.91). The average precision of Gracob over the three datasets under the five significance levels are 0.93, 0.84, 0.76, 0.62 and 0.54, which show that for this analysis Gracob is 26%, 71%, 105%, 88% and 108% more precise than the second best method, which are BicPAM (0.74), BicPAM (0.49), QUBIC (0.37), QUBIC (0.33) and SAMBA (0.26).

We further analyzed the enrichment over the three branches of GO terms (Biological Process, Cellular Component and Molecular Function). Our results revealed that the highest percentage of enriched GO terms among the co-fit genes detected by Gracob biclusters belong to the Cellular Component (CC) branch in all the analyzed species (Supplementary Fig. S2). This is in agreement with the findings in Hillenmeyer *et al.* (2010) that co-fitness is a powerful tool to predict cellular functions.

We conducted parameter sensitivity analysis of Gracob over the *E.coli* dataset. Gracob is very stable with respect to the changes of parameters r and δ, while less so when c increases (Supplementary Section S4 and Figs S3 and S4).

4.2 Case study on *E. coli* growth phenotype data

We now focus on the largest bicluster (Fig. 3(12a)) that Gracob detected in the *E. coli* growth phenotype dataset. The bicluster groups 79 gene knock-out strains under 10 stress conditions (see Tables S1&S2 for details). The knock-out of any of these 79 genes leads to significantly reduced cell growth under these 10 conditions, although none of them is an essential gene. The 10 conditions consist of seven carbon-source conditions, one nitrogen-source condition and two ferrous sulfate-source conditions. These sources are known to be transported and metabolized by pathways that require amino acids, purines, pyrimidines and cofactors to be synthesized (Kim and Copley, 2007). Thus, deletions of genes involved in such pathways are expected to impact the cell growth under these conditions (Gottschalk, 1986).

Among the 79 genes, there are 74 enzyme coding genes (Section S5). We found that 72 of them are closely connected through KEGG pathways as can be seen in Supplementary Figure S5. In fact, 70 genes (88.6% of the genes in this bicluster) are involved in ‘Metabolic pathways’. This is statistically significant because only 15.2% of the total 3979 genes are known to be involved in metabolic pathways (see Tables S3 and Section S5 for details). The second and third most significant KEGG pathways in this bicluster are ‘Biosynthesis of secondary metabolites’ and ‘Biosynthesis of amino acids’, in which 44 and 41 of the genes are involved, respectively. This is interesting because secondary metabolites generally do not play a role in growth under the normal condition. However, it is discovered that they can be important in survival of organisms because they are involved in physiological functions like stress-response (Price-Whelan *et al.*, 2006).

Growth phenotype data can be used not only to analyze conditional essentiality and dispensability of genes for specific environmental settings (Hillenmeyer *et al.*, 2008; Nichols *et al.*, 2011), but also to facilitate computational analysis to gain new insights into the functional organization of genes (Deutschbauer *et al.*, 2011; Giaever *et al.*, 2002; Lee *et al.*, 2005; Nichols *et al.*, 2011). Since about one-third of the protein-coding genes are still uncharacterized (i.e. orphan genes) even in *E. coli* (Blattner *et al.*, 1997; Hu *et al.*, 2009)—one of the most well-known biological systems—such analysis is crucial to unraveling how the interplay of genetic and environmental factors orchestrates cellular-level phenotypes.

To illustrate this point, we examined the genes in the largest bicluster and analyzed the function of ycdY, which is the only orphan gene in this bicluster. This orphan gene is known to code for a chaperone protein that is suggested to be a redox enzyme maturation protein (REMP) (Turner *et al.*, 2004). No functional annotation is defined for ycdY. Surprisingly, ycdY deletion has strong effects on growth under these 10 conditions (P-value $= 3.33 \times 10^{-16}$). In order to predict its function we looked for the most significantly enriched GO terms in this bicluster. Seventy-one out of the 79 genes (89.9%) are annotated as ‘organonitrogen compound biosynthetic process’ whereas only 485 genes are annotated as this GO term among all the 3979 *E.coli* genes in this dataset, which gives a P-value of 9.57×10^{-55}. Other most significantly enriched GO terms are ‘cellular amino acid biosynthetic process’ (P-value $= 1.37 \times 10^{-49}$), ‘small molecule biosynthetic process’ (P-value $= 1.13 \times 10^{-48}$), ‘cellular amino acid metabolic process’ (P-value $= 2.18 \times 10^{-43}$) and ‘organonitrogen compound metabolic process’ (P-value $= 2.08 \times 10^{-42}$). Therefore, our analysis strongly suggests that the function of ycdY to be associated with these five GO terms.

Another case study on a bicluster containing 11 genes that are essential under three dyeing chemical conditions but are dispensable under a cold shock and an antibiotic, Spectinomycin, condition also demonstrates the value of Gracob (Section S6).

5 Conclusion

In this paper, we proposed a novel graph-based biclustering method, Gracob, that is developed to discover co-fit genes from large growth phenotype profiling datasets. To our knowledge, Gracob is the first biclustering method developed specifically to mine growth phenotype data. Experimental results from both a variety of synthetic datasets and three genome-scale growth phenotype datasets for *E.coli*, proteobacteria and yeast demonstrate the superior performance of Gracob over a great collection of the widely used biclustering methods to discover co-fit genes.

Funding

The research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/1976-04. NIH U01HG008488, NIH R01GM115833, NIH U54GM114833, NSF DBI-1565137, and NSF IIS-1313606.

Conflict of Interest: none declared.

References

Baba,T. *et al.* (2006) Construction of *Escherichia coli* K-12 in-frame, single-gene knockout mutants: the Keio collection. *Mol. Syst. Biol.*, **2**, 2006.0008.

Ben-Dor,A. *et al.* (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.*, **10**, 373–384.

Bergmann,S. *et al.* (2003) Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys. Rev. E*, **67**, 031902.

Blattner,F.R. *et al.* (1997) The complete genome sequence of *Escherichia coli* K-12. *Science*, **277**, 1453–1462.

Bochner,B.R. (2009) Global phenotypic characterization of bacteria. *FEMS Microbiol. Rev.*, **33**, 191–205.

Bozdağ,D. *et al.* (2009). A biclustering method to discover co-regulated genes using diverse gene expression datasets. In: *Proceedings of the 1st International Conference on Bioinformatics and Computational Biology*, BICoB ’09, pp. 151–163, Berlin, Heidelberg. Springer-Verlag.

Cheng,Y. and Church,G.M. (2000) Biclustering of expression data. *Intell. Syst. Mol. Biol. (ISMB)*, **8**, 93–103.

Cho,H. *et al.* (2004). Minimum sum-squared residue co-clustering of gene expression data. In: *SIAM International Conference on Data Mining, SDM*, vol. 3, p. 3. SIAM.

Deutschbauer,A. *et al.* (2011) Evidence-based annotation of gene function in *Shewanella oneidensis* MR-1 using genome-wide fitness profiling across 121 conditions. *PLoS Genet.*, **7**, e1002385.

Deutschbauer,A. *et al.* (2014) Towards an informative mutant phenotype for every bacterial gene. *J. Bacteriol.*, **196**, 3643–3655.

Eisen,M.B. *et al.* (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A.*, **95**, 14863–14868.

Eren,K. *et al.* (2013) A comparative analysis of biclustering algorithms for gene expression data. *Brief. Bioinform.*, **14**, 279–292.

Giaever,G. *et al.* (2002) Functional profiling of the *Saccharomyces cerevisiae* genome. *Nature*, **418**, 387–391.

Gottschalk,G. (1986) Biosynthesis of Escherichia coli Cells from glucose. In: Gottschalk, G. (ed.), *Bacteria Metabolism*. Springer-Verlag New York Inc., New York, pp. 38–95.

Gu,J. and Liu,J.S. (2008) Bayesian biclustering of gene expression data. *BMC Genomics*, **9**, S4.

Gu,Z. *et al.* (2003) Role of duplicate genes in genetic robustness against null mutations. *Nature*, **421**, 63–66.

Gusenleitner,D. *et al.* (2012) iBBiG: iterative binary bi-clustering of gene sets. *Bioinformatics*, **28**, 2484–2492.

Harrison,R. *et al.* (2007) Plasticity of genetic interactions in metabolic networks of yeast. *Proc. Natl. Acad. Sci. U. S. A.*, **104**, 2307–2312.

Henriques,R. and Madeira,S.C. (2014) Bicpam: pattern-based biclustering for biomedical data analysis. *Algorithms Mol. Biol.*, **9**, 27.

Henriques,R. and Madeira,S.C. (2015) Biclustering with flexible plaid models to unravel interactions between biological processes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **12**, 738–752.

Henriques,R. *et al.* (2015) A structured view on pattern mining-based biclustering. *Pattern Recogn.*, **48**, 3941–3958.

Hillenmeyer,M.E. *et al.* (2008) The chemical genomic portrait of yeast: uncovering a phenotype for all genes. *Science*, **320**, 362–365.

Hillenmeyer,M.E. *et al.* (2010) Systematic analysis of genome-wide fitness data in yeast reveals novel gene function and drug action. *Genome Biol.*, **11**, R30.

Hochreiter,S. *et al.* (2010) Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, **26**, 1520–1527.

Hu,P. *et al.* (2009) Global functional atlas of *Escherichia coli* encompassing previously uncharacterized proteins. *PLoS Biol.*, **7**, e96.

Huttenhower,C. *et al.* (2009) Detailing regulatory networks through large scale data integration. *Bioinformatics*, **25**, 3267–3274.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27–30.

Kim,D.-U. *et al.* (2010) Analysis of a genome-wide set of gene deletions in the fission yeast *Schizosaccharomyces pombe*. *Nat. Biotechnol.*, **28**, 617–623.

Kim,J. and Copley,S.D. (2007) Why metabolic enzymes are essential or nonessential for growth of *Escherichia coli* K12 on glucose. *Biochemistry*, **46**, 12501–12511.

Kluger,Y. *et al.* (2003) Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.*, **13**, 703–716.

Kobayashi,K. *et al.* (2003) Essential *Bacillus subtilis* genes. *Proc. Natl. Acad. Sci. U. S. A.*, **100**, 4678–4683.

Korona,R. (2011) Gene dispensability. *Curr. Opin. Biotechnol.*, **22**, 547–551.

Lazzeroni,L. and Owen,A. (2002) Plaid models for gene expression data. *Stat. Sin.*, **12**, 61–86.

Lee,W. *et al.* (2005) Genome-wide requirements for resistance to functionally distinct DNA-damaging agents. *PLoS Genet.*, **1**, e24.

Li,G. *et al.* (2009) QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Res.*, **37**, e101–e101.

Liu,J. and Wang,W. (2003). Op-cluster: Clustering by tendency in high dimensional space. In: *Third IEEE International Conference on Data Mining, 2003. ICDM 2003*, pp. 187–194. IEEE.

Madeira,S.C. and Oliveira,A.L. (2004) Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **1**, 24–45.

Mayes,A.E. *et al.* (1999) Characterization of sm-like proteins in yeast and their association with u6 snrna. *EMBO J*, **18**, 4321–4331.

Murali,T. and Kasif,S. (2003) Extracting conserved gene expression motifs from gene expression data. *Pac. Symp. Biocomput.*, **8**, 77–88.

Nichols,R.J. *et al.* (2011) Phenotypic landscape of a bacterial cell. *Cell*, **144**, 143–156.

Pandey,G. *et al.* (2009) An association analysis approach to biclustering. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pp. 677–686, New York, NY, USA. ACM.

Pannone,B.K. *et al.* (2001) Multiple functional interactions between components of the Lsm2-Lsm8 complex, U6 snRNA, and the yeast La protein. *Genetics*, **158**, 187–196.

Papp,B. *et al.* (2004) Metabolic network analysis of the causes and evolution of enzyme dispensability in yeast. *Nature*, **429**, 661–664.

Prelić,A. *et al.* (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.

Price-Whelan,A. *et al.* (2006) Rethinking 'secondary' metabolism: physiological roles for phenazine antibiotics. *Nat. Chem. Biol.*, **2**, 71–78.

Segrè,D. *et al.* (2005) Modular epistasis in yeast metabolism. *Nat. Genet.*, **37**, 77–83.

Serin,A. and Vingron,M. (2011) Debi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms Mol. Biol.*, **6**, 18.

Sheng,Q. *et al.* (2003) Biclustering microarray data by Gibbs sampling. *Bioinformatics*, **19**, ii196–ii205.

Tanay,A. *et al.* (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18**, S136–S144.

Tanay,A. *et al.* (2004) Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genome-wide data. *Proc. Natl. Acad. Sci. U. S. A.*, **101**, 2981–2986.

Tharun,S. *et al.* (2000) Yeast sm-like proteins function in mrna decapping and decay. *Nature*, **404**, 515–518.

Turner,H. *et al.* (2005) Improved biclustering of microarray data demonstrated through systematic performance tests. *Comput. Stat. Data Anal.*, **48**, 235– 254.

Turner,R.J. *et al.* (2004) Sequence analysis of bacterial redox enzyme maturation proteins (remps). *Can. J. Microbiol.*, **50**, 225–238.

Wagner,A. (2000) Robustness against mutations in genetic networks of yeast. *Nat. Genet.*, **24**, 355–361.

Wang,H. *et al.* (2002). Clustering by pattern similarity in large data sets. In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pp. 394–405. ACM.

Wang,J.Z. *et al.* (2007) A new method to measure the semantic similarity of go terms. *Bioinformatics*, **23**, 1274–1281.

Yang,J. *et al.* (2002). δ-clusters: Capturing subspace correlation in a large data set. In: *Proceedings 18th International Conference on Data Engineering, 2002*, pp. 517–528. IEEE.

Yang,J. *et al.* (2003). Enhanced biclustering on expression data. In: *Proceedings Third IEEE Symposium on Bioinformatics and Bioengineering, 2003*. pp. 321–327. IEEE.