

Title:

Defining transcription modules using large-scale gene expression data

Jan Ihmels, Sven Bergmann and Naama Barkai*

Departments of Molecular Genetics and Physics of Complex Systems
Weizman Institute of Science, 76100 Rehovot, Israel

Running title: Defining modules using large-scale expression data

ABSTRACT

Motivation: Large-scale gene expression data comprising a variety of cellular conditions holds the promise of a global view on the transcription program. While conventional clustering algorithms have been successfully applied to smaller datasets, the utility of many algorithms for the analysis of large-scale data is limited by their inability to capture combinatorial and condition-specific co-regulation. In addition, there is an increasing need to integrate the rapidly accumulating body of other high-throughput biological data with the expression analysis. In a previous work, we introduced the Signature Algorithm, which overcomes the problems of conventional clustering and allows for intuitive integration of additional biological data. However, the applicability of this approach to global analyses is constrained by the comprehensiveness of relevant external data and by its lacking capability of capturing hierarchical organization of the transcription network.

Methods: We present a novel method for the analysis of large-scale expression data, which assigns genes into context-dependent and potentially overlapping regulatory units. We introduce the notion of a *transcription module* as a self-consistent regulatory unit consisting of a set of co-regulated genes as well as the experimental conditions that induce their co-regulation. Self-consistency is defined by a rigorous mathematical criterion. We propose an efficient algorithm to identify such modules, which is based on the iterative application of the signature algorithm we proposed previously. A threshold parameter that determines the resolution of the modular decomposition is introduced.

Results: The method is applied systematically to over 1,000 expression profiles of the yeast *S. cerevisiae*, and the results are presented using two complementary visualization schemes we developed. The average biological coherence, as measured by the conservation of putative *cis*-regulatory motifs between four related yeast species, is higher for transcription modules than for clusters identified by other methods applied to the same dataset. Our method is related to the Singular Value Decomposition (SVD) and to the pair-wise average linkage clustering algorithm. It extends SVD by filtering out noise in the expression data and offering variable resolution to reveal hierarchical organization. It furthermore has the advantage over both methods of capturing overlapping modules in the presence of combinatorial regulation.

Contact: naama.barkai@weizmann.ac.il

Supplementary Information: <http://www.weizmann.ac.il/~barkai/modules>

* *To Whom Correspondence should be addressed:*

Naama Barkai
Department of Molecular Genetics
Weizmann Institute, 76100 Rehovot, Israel
Tel: +972-8-934-4429
Fax: +972-8-934-4108
Email: naama.barkai@weizmann.ac.il

INTRODUCTION

Microarray experiments monitor the expression of thousands of genes simultaneously. The availability of this high-throughput technology has led to the accumulation of large datasets comprising thousands of expression profiles for a variety of organisms ranging from bacteria to human (Gollub et al. 2003). While particular genome-wide expression measurements are typically performed to address specific biological issues, it is widely recognized that a wealth of additional information can be retrieved from a large and diverse data set describing the genomic response to a variety of different conditions (Lander 1999). For example, such data can be used to provide functional links for uncharacterized genes (Hughes et al. 2000b; Ihmels et al. 2002; Tavazoie et al. 1999; Wu et al. 2002), to predict novel *cis*-regulatory elements (Bussemaker et al. 2001; Hughes et al. 2000a) and to elucidate design principles of transcriptional regulation (Wang et al. 2002). Yet, a limiting factor for such applications is the lack of proper computational tools (Bittner et al. 1999).

Grouping together genes of similar expression pattern is a general starting point in the analyses of gene expression data. Typically, similarity between genes is measured by the correlation of their expression profiles, and clustering methods are used to obtain a global partitioning of the data into clusters of genes exhibiting a similar expression pattern. Commonly used clustering algorithms include *k*-means clustering (Tavazoie et al. 1999), self-organizing maps (SOM) (Tamayo et al. 1999) and hierarchical methods (Alon et al. 1999; Eisen et al. 1998). While these algorithms have led to remarkable results when applied to relatively small datasets, their utility for the analysis of large datasets appears to be limited by several drawbacks (Bittner et al. 1999; Cheng and Church 2000; Gasch and Eisen 2002; Getz et al. 2000; Hastie et al. 2000; Tanay et al. 2002). First, commonly used methods assign each gene to one cluster, while in fact genes may participate in several biological functions and should thus be included in multiple clusters. Second, correlation in expression pattern is measured over all conditions, although genes are typically regulated only in specific experimental contexts. Several methods have been put forward to address these issues (Cheng and Church 2000; Gasch and Eisen 2002; Getz et al. 2000; Hastie et al. 2000; Tanay et al. 2002). In a recent work (Ihmels et al. 2002), we proposed a new approach that overcomes these drawbacks by incorporating additional biological information such as sequence data or functional annotation. Such *a-priori* information can be used to assemble putative groups of co-regulated genes. We proposed a simple algorithm, termed the *signature algorithm*, that refines such a set by identifying a co-regulated subset, removing genes that are in fact not co-regulated with this subset and adding genes from the genome that exhibit similar expression patterns. We have shown that this algorithm is capable of identifying small subsets of co-regulated genes even if the putative set contains a large number of unrelated genes. However, the applicability of this approach to global analyses is constrained by the comprehensiveness of relevant external data and by its lacking capability of capturing hierarchical organization of the transcription network.

Here, we propose a novel scheme that retains the advantages of the signature algorithm, while providing a global decomposition of the expression data into a hierarchy of transcription units at various resolutions. This approach is suitable for cases where no *a-priori* information is available, and can also be used to integrate external data in a natural way (Ihmels et al. 2004). In contrast to most clustering methods, where genes are grouped by optimizing all clusters simultaneously (Duda et al. 2001), our approach focuses on the properties of the individual co-regulated units themselves. We provide a rigorous definition of a *transcription module* as a self-consistent regulatory unit consisting of co-regulated genes and the regulating conditions. A threshold parameter controls the stringency of co-regulation between the module genes. We propose an efficient method for identifying and visualizing transcription modules at different resolutions. Within our approach, each module is evaluated individually, allowing the assignment of genes or conditions to several modules. We apply our method to over 1,000 expression profiles of the yeast *S. cerevisiae* and analyze the results. Finally, we compare the results with those of commonly used clustering methods that were applied to the same dataset, using an

objective biological figure of merit which is based on conservation of putative *cis*-regulatory elements between four related yeast species (Kellis et al. 2003).

METHODS AND ALGORITHM

Transcription modules

Definition: A transcription module consists of a set of co-regulated genes (a subset G_m of all genes G) and an associated set of regulating conditions (a subset C_m of all conditions C). Optionally, each gene g and each condition c may also be characterized by scores s_g and s_c respectively, that indicate their relative importance. If no preference is given to any of the genes or conditions all scores are set to unity. The defining property of a transcription module is self-consistency, which is introduced as follows: First, we assign new scores to both genes and conditions that reflect their actual degree of association with the module. The gene score is the average expression of each gene over the module conditions, weighted by the condition score:

$s_g = \left\langle s_c E_C^{gc} \right\rangle_{c \in C_m}$, where $\langle \rangle_i$ denotes the average over the subscript i . Analogously, the

condition score is the weighted average over the module genes, $s_c = \left\langle s_g E_G^{gc} \right\rangle_{g \in G_m}$. Here, E_G^{gc}

and E_C^{gc} are the log-expression ratios of gene g in condition c normalized over genes and

conditions, respectively, such that $\left\langle E_G^{gc} \right\rangle_{g \in G} = 0$, $\left\langle (E_G^{gc})^2 \right\rangle_{g \in G} = 1$ for each c and $\left\langle E_C^{gc} \right\rangle_{c \in C} = 0$,

$\left\langle (E_C^{gc})^2 \right\rangle_{c \in C} = 1$ for each g . Self-consistency denotes the property that the genes of the module

are exactly those genes of the dataset that receive the highest scores s_g while the module conditions are those conditions of the dataset with the highest scores s_c .

Identification through iterative signature algorithm: To identify transcription modules, we iteratively apply the signature algorithm introduced in (Ihmels et al. 2002). The signature algorithm consists of the following two steps: First, all conditions in the dataset are scored, following the scoring procedure described above, using a given set of genes as reference. In the iterative scheme, this initial reference set of genes $G^{(0)}$ is chosen at random, and we assign a uniform score to its genes. The conditions whose absolute score $|s_c|$ exceeds the condition threshold t_C are selected. This set of conditions is denoted $C^{(0)}$. In the second step, all genes are scored, using $C^{(0)}$ as the reference condition set. The genes with a score s_g greater than the gene threshold t_G are selected. This set of genes is denoted as $G^{(1)}$ and (together with their associated score) defines the output of the signature algorithm. Subsequently we apply the signature algorithm to $G^{(1)}$ to obtain $G^{(2)}$. We repeat this procedure until convergence is reached, i.e. $G^{(n+1)} = G^{(n)}$. By definition, the fixed point $G^{(n)}$ defines a transcription module. The threshold values are given in units of the expected standard deviation (corresponding to uncorrelated genes or conditions).

Note that while conditions may have negative scores, gene scores are always positive, such that only positively correlated genes are assigned to the same module. Negative correlations can be captured through correlations between separate modules (see discussion below and supplementary data).

Fixed points are identified heuristically, by initiating the signature algorithm with a large number of random initial sets. The gene threshold t_G determines the resolution of the modular decomposition, and was varied over the range from 1.8 to 4.0 in steps of 0.1 in the present work. The condition threshold was found to have a minor effect on the resulting fixed points over a comparable range, and was set to $t_C = 2$ throughout the analysis.

Module fusion

Application of the iterative scheme to random input sets frequently produces a number of highly similar fixed points that differ only by a few genes. Since such small differences are unlikely to be of biological origin, we fused those fixed points whose correlation coefficients (calculated according to gene and condition scores) exceeded some threshold (taken here as 0.8). The results are not sensitive to the exact threshold value. A representative transcriptional module was found by reiterating the average of all these fixed points.

Clusters defined by other methods

Pair-wise average linkage, K-means and SOM: Clusters were generated using the program "Cluster" (Eisen et al. 1998) using the default parameters and normalizations as prescribed in the manual. The program is available at <http://rana.lbl.gov>.

Singular value decomposition (SVD): The standard Matlab SVD function was used to decompose the expression matrix into eigengenes and eigenconditions. Modules were extracted by defining a cut-off for the eigenconditions, normalized such that the maximum absolute component for each eigencondition was unity. For each eigencondition, the genes whose value exceed the cut-off constitute the corresponding cluster. Several values for the cut-offs were tested (0.1, 0.3, 0.5, 0.7 and 0.9), as well as choosing genes by their signed or absolute value. Shown in this work are the best results (cut-off 0.1, signed values).

Bi-clustering: Bi-clusters were generated using the program "Bicluster" published in (Cheng and Church 2000) (available at <http://cheng.eecs.uc.edu/biclustering>). The program was applied to the expression data both in the form of ratios as well as the logarithm of ratios. The results shown were obtained from the log-based expression matrix, which were slightly better.

Coupled two-way clustering (CTWC): The clustering algorithm used was SPC (Blatt et al. 1996). Minimum sample size was 5, ignore drop out size was 9 for genes and 4 for samples, stable delta T was 1 for genes and 2 for samples, K=15 for genes and automatic determination for samples, depth was maximal.

Definition of overlaps

The symmetric overlap between clusters A and B was defined as $OL^{sym}(A, B) = N_{A \cap B} / \sqrt{N_A \cdot N_B}$. Here, N_A and N_B denote the sizes of A and B , respectively, and $N_{A \cap B}$ denotes the size of their intersect. The asymmetric overlap is normalized by the size of only the first cluster: $OL^{asym}(A, B) = N_{A \cap B} / N_A$.

Sequence information for the four yeast species

The sequence information for the four yeast species *S. cerevisiae*, *S. paradoxus*, *S. mikatae* and *S. bayanus* were downloaded from the supplementary information page of (Kellis et al. 2003) at <http://www-genome.wi.mit.edu/personal/manoli/yeasts/>. For each ORF in *S. cerevisiae*, a 600bp sequence upstream of the start site was identified in each of the species (provided the homolog existed).

Calculation of enrichment p-values

To test whether a hexamer h is significantly overrepresented in the upstream region of a given set of genes, we calculated the probability of obtaining the observed enrichment by chance (p-value). Consider a group of N genes, where a number z of these genes contain the hexamer h in their upstream region. The significance of this observation depends on the overall number of genes in the genome containing h , denoted by K . The corresponding p-values were computed using the hypergeometric probability density function given by

$$P(h; cluster) = 1 - \sum_{x=0}^{z-1} \binom{K}{x} \cdot \binom{M-K}{N-x} / \binom{M}{N},$$

where M is the total number of genes in the genome and K denotes the total number of genes containing the sequence. We use the form $P = \sum_{x=z}^N \binom{K}{x} \cdot \binom{M-K}{N-x} / \binom{M}{N}$ for numerical precision, and use $-\log_{10}(P)$ throughout the work.

Definition of the biological figure of merit (BFM)

To calculate the BFM for a given set of genes g we proceeded in three steps:

First, we identified all hexamers that are significantly overrepresented in the 600bp upstream region of all genes in *g* in *S. cerevisiae*. To this end, we calculated for each possible hexamer the enrichment p-value, denoted P_{cer} , as described in the previous section.

In the second step, we repeated the same procedure for four related yeast strains (Kellis et al. 2003), where the enrichment of genes containing the hexamer in the promoter region of all four species simultaneously was evaluated, yielding the conservation p-value P_{all} . Thus, after the first two steps every hexamer h was associated with two p-values $P_{cer}(h)$ and $P_{all}(h)$.

In the last step, the most significant hexamer h_{sig} with the smallest p-value P_{cer} in *S. cerevisiae* was identified, and the BFM is defined as the conservation p-value P_{all} of this hexamer, $BFM \equiv -\log_{10} P_{all}(h_{sig})$. In order to take into account not only the most enriched motif, but allow for either motif-variations or multiple motif dependencies, we additionally considered the hexamers whose value of $-\log_{10}(P_{cer})$ was at least 50% of the maximum value identified (i.e. for h_{sig}). Note that therefore more than one BFM may be associated with one gene set. Only gene sets with between 20 and 400 genes were considered in the analysis. For clusters of the same algorithm whose overlap exceeded 80%, one representative with the largest BMF was chosen and the remaining clusters were excluded.

Web site

The web site containing detailed information about all transcription modules, figures of merit, sequences, alignments and expression data references is available at <http://www.weizmann.ac.il/~barkai/Modules>.

RESULTS

Transcription Modules

Our approach for decomposing the expression data into sets co-regulated genes focuses on the properties of the individual co-regulated units. Such *transcription modules* consist of a set of genes that are co-regulated in a specific cellular context, and the set of experimental conditions where this co-regulation is most stringent (Fig. 1a). The defining property of a transcription module is its self-consistency: The module includes all the genes that are most tightly co-regulated over the conditions assigned to the module. All genes outside the module are less correlated than the included genes. Similarly, the module conditions are those conditions in the dataset that regulate the expression of the module genes most tightly, and conditions not included in the module have less influence on the module genes. A mathematical definition of a transcription module is given in the Methods.

Having introduced a rigorous definition of a transcription module, all possible sets of genes and conditions could in principle be evaluated for their compatibility with this criterion. However, since the number of all sets scales exponentially with the size of the data, such a procedure is computationally infeasible. Instead, we follow a heuristic approach for identifying modules by iteratively refining a random set of genes (or conditions) until a self-consistent unit is obtained. Specifically, modules are fixed-points of the recently introduced signature algorithm (Ihmels et al. 2002) (Methods). Applying the signature algorithm iteratively refines the co-regulated unit with each step until convergence is reached. Computationally, identifying individual modules is efficient for two reasons. First, computation time for each iteration step scales only linearly with the number of genes and conditions (Bergmann et al. 2003). Second, the capacity of the signature algorithm to separate co-regulated genes from unrelated genes ensures rapid convergence of the algorithm (typically within only a few iterations).

Our definition of a transcription module allows for different degrees of co-regulation stringency. Within the iterative procedure a threshold parameter t_G imposes the minimal level of co-regulation within the module.

A set of genes that is self-consistent at a particular threshold in general remains approximately a fixed point also upon increasing or decreasing the threshold by a small amount. When the threshold is reduced, additional genes that are related to the module are included. However, if the threshold is lowered beyond a certain limit, the inclusion of remotely related

genes can shift the co-regulation pattern, resulting in convergence to a different module. This new fixed point may not include any genes of the original module (Fig. 1b). Conversely, when the threshold is increased, genes that are the least co-regulated with the bulk of the module genes are excluded. Raising the threshold beyond the actual level of co-regulation between the module genes results in the disappearance of the module upon iteration.

Transcription modules of *S. cerevisiae* expression data

We applied the iterative signature algorithm to a large dataset of over 1,000 genome-wide expression profiles of the yeast *S. cerevisiae*, including practically all publicly available expression measurements (see accompanying web-site for full list of references). Modules were identified for 23 threshold values ranging from $t_G = 1.8$ to $t_G = 4.0$ (in units of expected standard deviations, see Methods). At each threshold, we obtained a collection of fixed points by iterating from 20,000 initial gene sets of various sizes that were chosen at random from the whole dataset. All input sets converged to only a limited number of fixed points. Highly similar fixed points were fused into a representative transcription module (see Methods). Although the same numbers of initial gene sets were used at each threshold, the number of distinct modules that were identified differed greatly. At the lowest resolution ($t_G = 1.8$) all input sets converged to one of just five modules. Biologically, these modules correspond to the central functions of yeast (protein synthesis, stress, cell cycle and amino-acid biosynthesis). Their gene content (between ~100 and ~300 genes in each module) is highly consistent with their associated functions (according to annotations taken from the YPD (Costanzo et al. 2001) and SGD (Issel-Tarver et al. 2002) databases). Together, these five modules capture the most significant features of the expression profiles. About 90% of the input sets converged either to the stress or to the protein synthesis module, reflecting the activation of those modules under numerous conditions in the dataset. As t_G is increased, the number of modules initially rises and their average size decreases, reflecting the finer resolution at which the data is viewed (Fig. 2d,e). At the intermediate resolution $t_G = 3.1$, the number of modules reaches a maximum. As the threshold is increased further, modules are lost since the required stringency exceeds the actual strength of co-regulation in the dataset. The total number of co-regulated genes identified at each threshold does not depend strongly on t_G , however at each threshold, different genes are identified (Fig. 2f). A total of 2,956 genes and all conditions are associated with at least one transcription module.

The fraction of initial sets that converged into a given module provides a measure of the relative importance of this module in the expression data. We denote this fraction as the *module recurrence*. Modules that consist of a large number of tightly co-regulated genes and many regulating conditions are identified repeatedly and receive high recurrence values. Genes assigned to modules with high recurrence tend to exhibit greater changes in their expression values (Fig. 2a). In contrast, convergence towards small and weakly co-regulated modules is rare and associated with low recurrence.

Since each module is evaluated individually, genes and conditions can be assigned to several, overlapping modules. Indeed, most experimental conditions were associated with several modules, resulting in a significant overlap between module conditions (Fig. 2b). In contrast, overlap between the module genes is moderate, with most modules sharing no or only a few genes (Fig. 2c). Because of the symmetry of our algorithm with respect to genes and conditions it is likely that the low extent of overlap between module genes reflects a true tendency of the yeast transcription program to avoid overlapping regulation, rather than being an artifact of the algorithm.

Out of the 2,956 genes that have been assigned to a transcription module, 1,112 correspond to ORFs of uncharacterized function. While this number is in agreement with the overall fraction of uncharacterized yeast ORFs (40%), the fraction of known genes within most individual modules in fact deviates significantly from what would be expected from a random assignment of genes to modules of the same size (Fig. 2g). Thus, most modules are either enriched with annotated genes, indicating their involvement in a well-studied function, or contain

primarily genes of unknown function, possibly pointing to a new function of *S. cerevisiae* about which there is little information. Similarly, in most individual modules, the fraction of genes that are part of a larger complex, and the fraction of essential genes, significantly deviate from what is expected from random gene assignment (Fig. 2h-i). This bias can be explained by the fact that several modules consist mainly of complex-associated genes (such as rRNA processing or proteasome), while the majority of modules do not. Similarly, a significant number of modules is associated with a central function of the organism (e.g. protein synthesis) and therefore enriched with essential genes. Most modules, however, are associated with specific responses, such as mating, sporulation or various stress responses and contain primarily non-essential genes.

The extent of self-consistency of clusters identified by common algorithms

By definition, all transcription modules identified by the Iterative Signature algorithms are self-consistent. We asked if also clusters generated by common methods satisfy the self-consistency criterion. To this end we applied available methods to the same data set. Methods that were tested include hierarchical clustering (pair-wise average-linkage (Eisen et al. 1998)), deterministic annealing (Alon et al. 1999), K-means (Tavazoie et al. 1999), self-organizing maps (Tamayo et al. 1999), SVD (Alter et al. 2000) and two bi-clustering methods (Cheng and Church 2000; Getz et al. 2000). The resulting clusters were analyzed and compared to the modules obtained by our method. We find that most of the clusters identified by the pair-wise linkage hierarchical clustering (Eisen et al. 1998) are approximately self-consistent and comply to large extent with our definition of transcription modules (Fig. 3a,b). In contrast, most clusters identified by the other algorithms do not obey the self-consistency requirement.

Biological merit of transcriptional modules

Since the underlying structure of the yeast transcription network is unknown, quantifying the biological significance of the results from different clustering methods is difficult. We devised a measure for the biological merit of each module that is based on the conservation of putative *cis*-regulatory elements between four related yeast species (Kellis et al. 2003) (Methods). The approach is illustrated in Fig. 5a. First, we identify a hexamer that is over-represented in the promoter sequence of the genes in the module. Such an over-represented sequence is generally found also in the promoter regions of numerous genes outside the module. We therefore next examine the conservation of the motif in the promoter regions of homologous genes in three related yeast species. Regulatory motifs are mostly distinguished by an increased conservation between related species. This conservation of overrepresented motifs can thus serve as a more reliable measure for the biological coherence of the module. The precise definition of the biological figure of merit (*BFM*) is given in the Methods.

We measured the *BFMs* of all modules identified by our analysis and of clusters identified by other commonly used clustering methods (Fig. 4b-c). Considerable fractions of the clusters identified by the iterative signature algorithm, the pair-wise average-linkage (Eisen et al. 1998) and coupled-two-way clustering (Getz et al. 2000) methods were found to have highly significant *BFMs* (Fig. 4b). The iterative signature algorithm was successful in identifying the clusters with the highest *BFMs* (Fig. 4b), as well as yielding the highest overall average.

Examples of *BFMs* assigned to the transcription modules and the over-represented motifs are summarized in the supplementary information. The majority of the motifs correspond to experimentally verified *cis*-regulatory elements, known to be involved in the regulation of the associated genes. Several putative motifs were identified, providing predictions for future analysis. A complete list of *BFMs* and consensus motifs for all transcription modules can be found on our website.

From simple to complex modular description

By gradually varying the threshold parameter that controls the degree of co-regulation within the module, we produced a sequence of modular decompositions ranging from simple to highly differential. We developed two complementary representations for convenient visualization of the data. The layered representation (Fig. 5a) reflects the properties of modules identified at a given

resolution, while the module tree summarizes the entire modular structure over all resolutions. Within the layered representation, modules identified at a specific threshold are arranged in a plane based on the correlation between their condition scores, which captures the experimental context regulating the module genes. Correlated modules are placed close to each other, while modules that are inversely correlated are separated.

While the modular representation at the lowest resolution is concise and simple, it becomes significantly more complex for the larger number of modules identified at higher thresholds (Fig. 5a). However, if the modules of higher thresholds are iterated at a lower threshold, they converge into one of the low-resolution modules. This association reveals the relationships between the module structures of different resolutions. For example, most modules involved in carbon metabolism are related to the stress module, with the exception of glycolysis which is in fact associated with protein synthesis, reflecting the rapid growth of yeast cells in the presence of glucose.

The module tree captures these relationships and provides an overview of the entire modular structure at all resolutions (Fig. 5b). Sequences of related modules that are self-consistent over a range of thresholds are represented by lines. For example, the five modules of the lowest resolution remain stable for all thresholds with gradual changes in their content. The module tree preserves the intuitive representation offered by a dendrogram used in hierarchical clustering algorithms. However, in contrast to the usual hierarchical representation, here distinct branches may include common genes (Fig. 2f). Specifically, new branches can appear by the splitting of existing modules, reflecting the separation of a module into two subparts (Fig. 5c). Alternatively, the modules of the new branch may not share any genes with the module to which they converge at low resolution. This type of relationship may reflect the activation of two different processes under similar conditions (Fig. 5d).

DISCUSSION

We have presented a novel method for the analysis of large-scale data gene expression data. The main conceptual novelty of our approach is to focus on the desired property of the individual co-regulated unit that we wish to extract from the expression data. According to our definition such a transcription module consists of all genes that are similar when compared over the conditions of the module, and all conditions that are similar when compared over the module genes. We refer to this property as self-consistency. Importantly, this approach allows for an independent identification of each module. This is unlike commonly used clustering algorithms, which optimize the global data partition.

Our approach offers several advantages. First, any set of genes can be tested for compliance with our definition, or used for revealing a closely related module. Second, since modules are identified individually, genes and conditions can be assigned to several, overlapping modules. Third, our approach avoids the full partitioning of the data, such that only genes that are indeed co-regulated are associated. Finally, our method is computationally efficient and suitable for large datasets, whose analysis poses serious problems for other methods concerning memory requirements and execution time.

To assess if clusters identified by commonly used methods are self-consistent according to our definition we applied several algorithms to the full dataset of yeast expression profiles. We found that most algorithms do not produce self-consistent clusters, with the exception of the pair-wise linkage hierarchical algorithm. Within the pair-wise linkage procedure, clusters are assembled sequentially, such that at each step, genes or clusters that are most similar to each other are fused. Due to the continual requirement of maximum similarity, this procedure largely produces self-consistent clusters. However, clusters are not built up individually; at each step, every cluster (or gene) can be associated with only one of the existing clusters. In systems with combinatorial regulation, this may eventually compromise the self-consistency criterion

(Bergmann et al. 2003). In the dataset of *S. cerevisiae* analyzed here, we found a very low level of overlap between the genes of different modules, and accordingly the results produced by the pair-wise linkage method are largely self-consistent. The set of clusters contains most transcriptional modules identified in the ISA scheme and vice versa (not shown). The differences in the performance of the two algorithms is likely to be larger in organisms with a higher degree of combinatorial regulation.

We proposed to identify modules by iteratively refining random input gene sets, using the Signature Algorithm introduced previously (Ihmels et al. 2002). By definition, self-consistent transcription modules emerge as fixed points of this algorithm. The stringency of co-regulation between the genes is determined by a threshold parameter. To obtain a modular decomposition at different resolutions, we scanned over different values for this parameter. In a complementary publication (Bergmann et al. 2003) we argue that the ISA scheme is in fact a generalization of Singular Value Decomposition (Alter et al. 2000) and demonstrate analytically the central role of the threshold in distinguishing co-regulated genes in the presence of noise.

To evaluate the capacity of our method to extract biological information for actual expression data, we compared its performance with that of publicly available clustering methods. We define a biological figure of merit for each cluster, based on the conservation of putative *cis*-regulatory motifs between four related yeast species. We find that our method and agglomerative hierarchical clustering (Eisen et al. 1998) (both producing transcription modules by our definition) and the coupled two way clustering method (Getz et al. 2000) on average yield the highest figures of merit. In addition to the assessment of the output of novel clustering methods, this approach can be used for optimizing clustering parameters on a particular experimental dataset.

We developed two complementary presentation schemes to visualize the large-scale modular structure and provide detailed information of the genes and conditions assigned to each module. The 'layered presentation' focuses on the relations between the modules at a specific threshold, while the 'module tree' summarizes the full modular structure at all resolutions. Both representations are available on the accompanying web site providing a detailed description of all the modules identified in our analysis. Programs for applying our method to novel expression data are also available. Finally, we provide an additional application, which enables the projection of novel yeast genome-wide expression data on the modular structure identified by our analysis. This provides a rapid and efficient way to compare novel expression measurements with all existing expression profiles.

Application of our method is computationally efficient, since the computation time scales linearly with the number of genes and conditions. Calculation of correlation matrices, which poses a serious problem for large datasets, is not required. Application of some of the methods to the expression matrix used for this work requires large amounts of memory and long execution times. This is likely to impede the use of these methods on datasets of organisms with larger genomes, or many conditions. In contrast, our method can easily be run on an average desktop computer. The transcription modules presented in this work were computed in a run time of less than one day, while less comprehensive results with lower resolution can be obtained within minutes.

The approach described in this paper complements the *recurrent signature* method we proposed recently (Ihmels et al. 2002), which provides an intuitive way for integrating external biological information, such as sequence information, functional annotation or protein-protein interactions. In both approaches, modules contain the set of regulating conditions in addition to co-regulated genes. This regulatory context provides important biological insight into the module function. Correlation and dependencies between the conditions assigned to different modules reflect higher-order organization in the expression data and may be used to elucidate system-level properties of the transcription programs (see supplementary information).

In this work, we analyzed expression data of the model organism *S. cerevisiae*. Since our approach can identify overlapping clusters, it is well-suited for the analysis of data in higher eukaryotes (Bergmann et al. PLoS, in press), where combinatorial regulation is likely to play a more prominent role. Exploration of the rapidly accumulating expression data with our methods should provide interesting insights into the common design features of transcriptional regulation.

ACKNOWLEDGEMENTS

We thank G. Getz for providing us with the results of the CTWC. We thank E. Domany, G. Getz and G. Friedlander for discussions. This work was supported by the NIH Grant No. A150562, the Israeli Science Ministry and the Y. Leon Benozio Institute for Molecular Medicine.

FIGURE LEGENDS

Figure 1: Definition and properties of transcription modules. (a) A transcription module is a self-consistent regulatory unit consisting of co-regulated genes together with the experimental conditions that induce their co-regulation. (b) Modules usually remain stable over a range of thresholds with gradual changes in their content. However, significant modifications occur once the threshold is changed beyond this range of stability. The genes assigned to a module at each threshold are represented by small rectangles and are arranged in a vertical sequence according to their gene score (Methods). The glycolysis module (brown colors) remains stable up to $t_G = 3.6$. At lower thresholds, genes coding for ribosomal proteins (green/blue colors) are added to this module, reflecting some degree of correlation between the glycolysis and ribosomal protein modules. These genes shift the co-regulation pattern of the module, eventually leading to its convergence into the ribosomal protein module. Sequences of stable modules are represented by a line (c.f. module tree, Fig. 3).

Figure 2: Statistical properties of transcription modules. (a) The recurrence (*Rec.*) of a module denotes the fraction of input sets that converged to that module. This measure is strongly correlated with the change in module expression over the full dataset. For each gene in a particular module, the standard deviation of its expression values over all conditions in the dataset was measured. Plotted here is the mean of this value over all genes in the module, for all modules identified at threshold $t_G = 2.1$. The correlations become weaker at higher thresholds. (b,c) Distribution of overlaps between transcriptional modules, according to their condition (b) or gene (c) content. Shown are the maximum overlaps between each module and all the other modules of the same resolution. (d-f) Properties of modules identified at different thresholds. Shown are the number of modules (d), their average size (e), and the total (●) or accumulative (○) number of genes assigned to at least one module (f). (g-i) The fractions of non-essential genes, genes of known function and genes that are part of complexes were calculated for each module. The respective distributions (solid bars) are significantly different from those expected for random assignments (open bars). Annotations are according to the YPD database (Costanzo et al. 2001). Complex data was taken from (Gavin et al. 2002).

Figure 3: Self-consistency of clusters produced by various algorithms. (a) The self-consistency for a cluster A consisting of N_A genes, was measured as follows: Each gene in the genome was scored by applying the signature algorithm to the cluster A (without gene threshold, and starting with unit gene scores for all genes). The degree of self-consistency was then quantified by the fraction of the N_A top-scoring genes that were also part of the original cluster A . By definition, this fraction is unity when the procedure is applied to transcription modules, where the top-scoring genes correspond exactly to the module genes. (b) The histograms of self-consistency measures are shown for the different clustering methods. Each line corresponds to a different clustering method, according to the color-coding specified in the legend. Most of the clusters obtained using the pair-wise average-linkage hierarchical clustering (Eisen et al. 1998) are self-consistent.

Figure 4: Biological figure of merit. (a) The biological figure of merit (*BFM*) for each cluster was defined based on conservation of an over-represented motif within four related yeast species. The scheme illustrates how the enrichment p-values P_{cer} and P_{all} were calculated. The precise procedure is described in the Methods. (b) The distribution of *BFMs* for the clusters produced by various clustering algorithms. The height of each bar at position x represents the fraction of clusters whose associated *BFM* fall into the interval $x \leq BFM \leq (x+10)$. For each clustering algorithm, the distribution in the regime of low *BFM* is shown on the left and for higher *BFM* on the right. Color coding is as in Fig. 3. (c) Shown is the mean *BFM* for each algorithm tested.

Figure 5: Modular decomposition of the yeast expression data. (a) Layered presentation of yeast modules identified at three different resolutions. Modules are represented by colored circles positioned in a plane according to the correlation between their condition scores. Correlated modules are close to each other, while inversely-correlated modules are separated. High-resolution modules are colored according to the module to which they converge when iterated at lower resolution. The module recurrence (see text) is presented by pie charts using the same color scheme. (b) Module tree summarizing the modules obtained at different resolutions. Branches represent modules (rectangles) that remain fixed points over a range of thresholds. Fixed points that emerge at high threshold converge into an existing module when iterated at a lower level (thin transversal lines), and are colored accordingly. For clarity, only a sub-tree is shown. (c,d) Modules can be thought of as local minima of some energy function whose shape depends on t_G . New modules at higher thresholds correspond to new local minima of this energy function. (c) New minima can appear by the splitting of existing minima. The two minima move gradually away from each other as the threshold is increased, reflecting the separation of a module into two subparts. In the example shown, the protein synthesis (PrSynt) module splits into two modules associated with ribosomal protein (RP) and rRNA processing. Shown is the asymmetric overlap between the protein synthesis module and the modules corresponding to rRNA (red) and ribosomal proteins (blue). The black line represents the symmetric overlap between the two modules after the splitting (see Methods for definition of overlaps). (d) Alternatively, new local minima can appear in a distant region such that the new module is disjoint from the module to which it converges at low resolution. This type of transition may reflect the activation of two different processes under similar conditions. The module of oxidative phosphorylation converges at low threshold to the stress, but does not share any genes with this module. Shown are the directional overlap between the stress module, and the modules corresponding to stress (Str, blue) and oxidative phosphorylation (OxP, red). The black line shows the overlap between the two modules after the splitting.

REFERENCES

- Alon, U., N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A* **96**: 6745-6750.
- Alter, O., P.O. Brown, and D. Botstein. 2000. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A* **97**: 10101-10106.
- Bergmann, S., J. Ihmels and N. Barkai. 2003. Iterative signature algorithm for analyzing large scale gene expression data. *Phys. Rev. E*, **67**: 031902.
- Bittner, M., P. Meltzer, and J. Trent. 1999. Data analysis and integration: of steps and arrows. *Nat Genet* **22**: 213-215.
- Blatt, M., S. Wiseman, and E. Domany. 1996. Superparamagnetic clustering of data. *Physical Review Letters* **76**: 3251-3254.
- Bussemaker, H.J., H. Li, and E.D. Siggia. 2001. Regulatory element detection using correlation with expression. *Nat Genet* **27**: 167-171.
- Cheng, Y. and G.M. Church. 2000. Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol* **8**: 93-103.
- Costanzo, M.C., M.E. Crawford, J.E. Hirschman, J.E. Kranz, P. Olsen, L.S. Robertson, M.S. Skrzypek, B.R. Braun, K.L. Hopkins, P. Kondu, C. Lengieza, J.E. Lew-Smith, M. Tillberg,

- and J.I. Garrels. 2001. YPD, PombePD and WormPD: model organism volumes of the BioKnowledge library, an integrated resource for protein information. *Nucleic Acids Res* **29**: 75-79.
- Duda, Hart, and Stork. 2001. *Pattern Classification*. John Wiley & Sons, Inc., New York.
- Eisen, M.B., P.T. Spellman, P.O. Brown, and D. Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* **95**: 14863-14868.
- Gasch, A.P. and M.B. Eisen. 2002. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol* **3**: RESEARCH0059.
- Gavin, A.C., M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A.M. Michon, C.M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M.A. Heurtier, R.R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**: 141-147.
- Getz, G., E. Levine, and E. Domany. 2000. Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci U S A* **97**: 12079-12084.
- Gollub, J., C.A. Ball, G. Binkley, J. Demeter, D.B. Finkelstein, J.M. Hebert, T. Hernandez-Boussard, H. Jin, M. Kaloper, J.C. Matese, M. Schroeder, P.O. Brown, D. Botstein, and G. Sherlock. 2003. The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Res* **31**: 94-96.
- Hastie, T., R. Tibshirani, M.B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W.C. Chan, D. Botstein, and P. Brown. 2000. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol* **1**: RESEARCH0003.
- Hughes, J.D., P.W. Estep, S. Tavazoie, and G.M. Church. 2000a. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J Mol Biol* **296**: 1205-1214.
- Hughes, T.R., M.J. Marton, A.R. Jones, C.J. Roberts, R. Stoughton, C.D. Armour, H.A. Bennett, E. Coffey, H. Dai, Y.D. He, M.J. Kidd, A.M. King, M.R. Meyer, D. Slade, P.Y. Lum, S.B. Stepaniants, D.D. Shoemaker, D. Gachotte, K. Chakraburty, J. Simon, M. Bard, and S.H. Friend. 2000b. Functional discovery via a compendium of expression profiles. *Cell* **102**: 109-126.
- Ihmels, J., G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. 2002. Revealing modular organization in the yeast transcriptional network. *Nat Genet* **31**: 370-377.
- Ihmels, J., R. Levy and N. Barkai. 2004. Principles of transcriptional control in the metabolic network of *Saccharomyces cerevisiae*. *Nat Biotech* **22**: 86:92.
- Issel-Tarver, L., K.R. Christie, K. Dolinski, R. Andrada, R. Balakrishnan, C.A. Ball, G. Binkley, S. Dong, S.S. Dwight, D.G. Fisk, M. Harris, M. Schroeder, A. Sethuraman, K. Tse, S. Weng, D. Botstein, and J.M. Cherry. 2002. *Saccharomyces Genome Database*. *Methods Enzymol* **350**: 329-346.
- Kellis, M., N. Patterson, M. Endrizzi, B. Birren, and E.S. Lander. 2003. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **423**: 241-254.
- Lander, E.S. 1999. Array of hope. *Nat Genet* **21**: 3-4.
- Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A* **96**: 2907-2912.
- Tanay, A., R. Sharan, and R. Shamir. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **18 Suppl 1**: S136-144.
- Tavazoie, S., J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. 1999. Systematic determination of genetic network architecture. *Nat Genet* **22**: 281-285.
- Wang, W., J.M. Cherry, D. Botstein, and H. Li. 2002. A systematic approach to reconstructing transcription networks in *Saccharomyces cerevisiae*. *Proc Natl Acad Sci U S A* **99**: 16893-16898.

Wu, L.F., T.R. Hughes, A.P. Davierwala, M.D. Robinson, R. Stoughton, and S.J. Altschuler. 2002. Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nat Genet* **31**: 255-265.

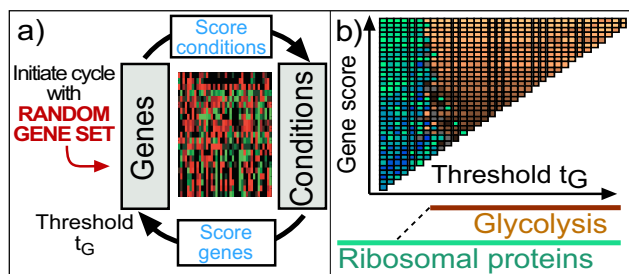
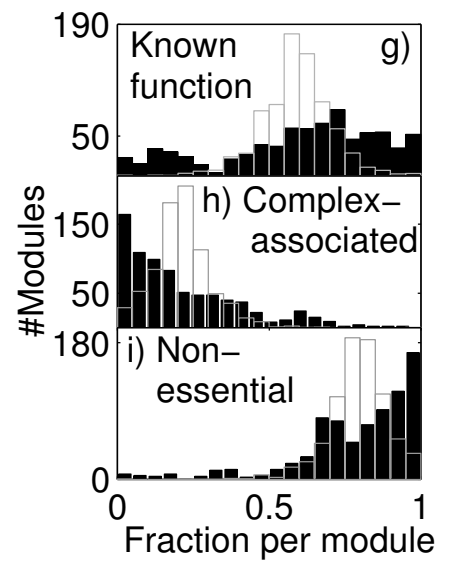
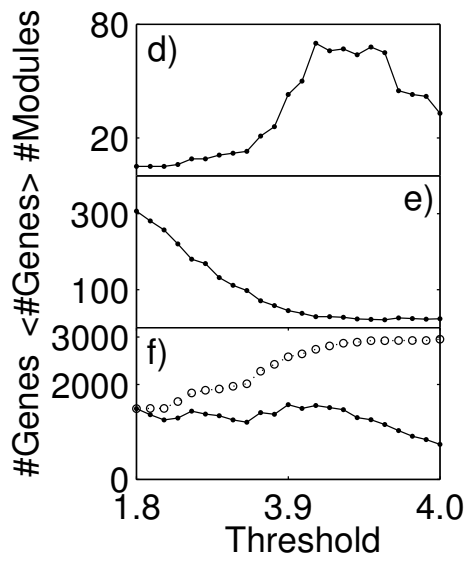
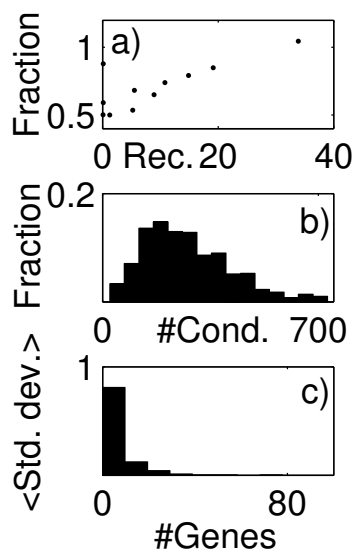


Figure 1



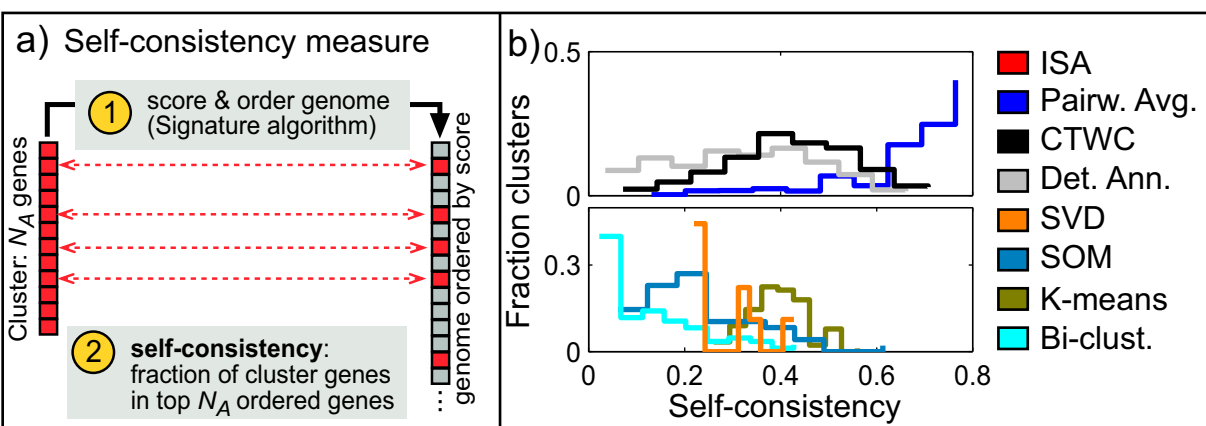


Figure 3

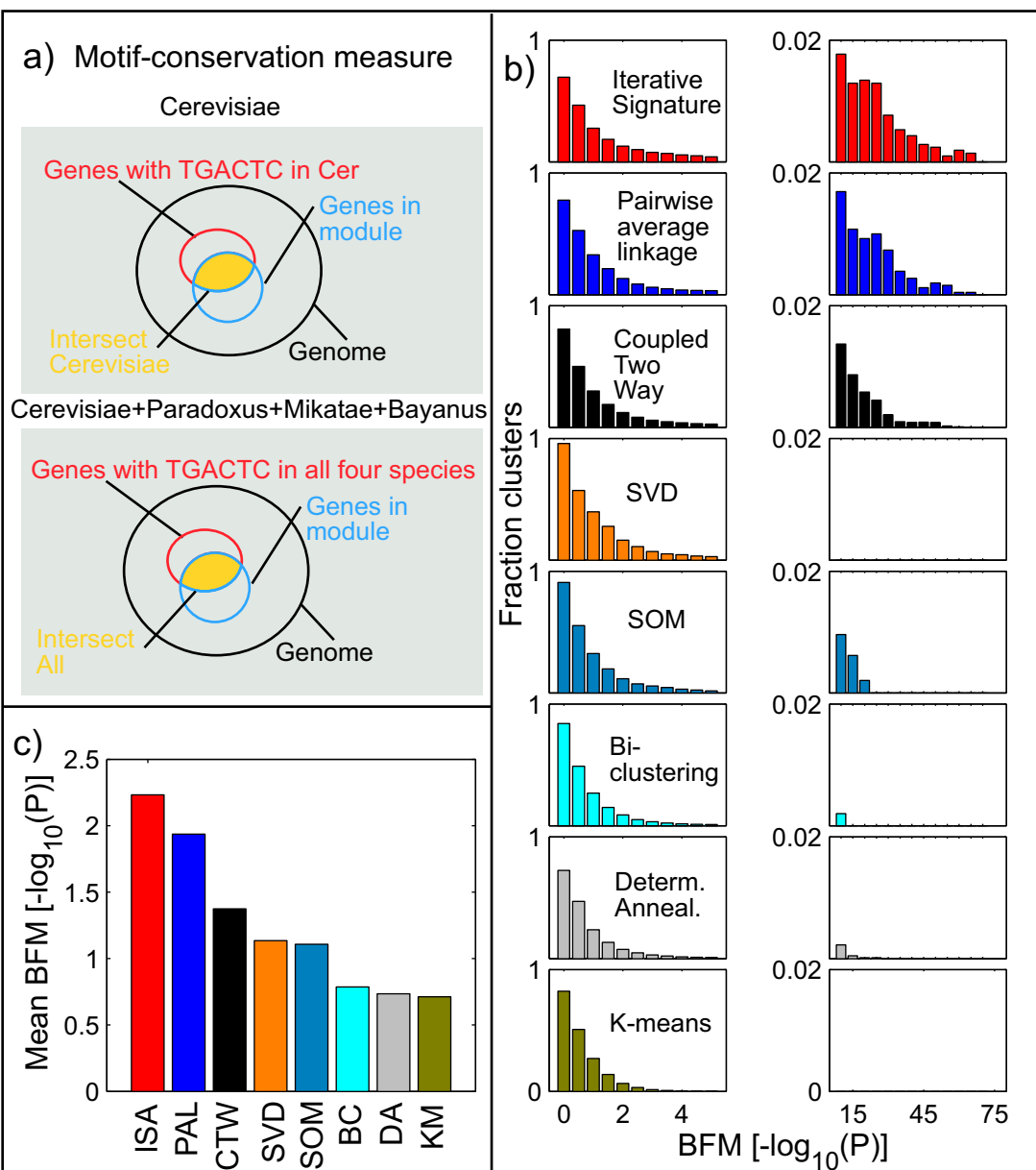


Figure 4

