Title

A mathematical and computational framework for quantitative comparison and integration of large-scale gene expression data

Journal

Nucleic Acids Research, 33(8)

ISSN

1362-4962

Authors

Hart, Christopher E.
Sharenbroich, Lucas
Bornstein, Benjamin J.
et al.

Publication Date

2005-05-10

2580–2594 *Nucleic Acids Research, 2005, Vol. 33, No. 8*
doi:10.1093/nar/gki536

A mathematical and computational framework for quantitative comparison and integration of large-scale gene expression data

Christopher E. Hart, Lucas Sharenbroich[1], Benjamin J. Bornstein[1], Diane Trout, Brandon King, Eric Mjolsness[2,3] and Barbara J. Wold*

Division of Biology, California Institute of Technology, Pasadena, CA 91125, USA, [1]Jet Propulsion Laboratory, Machine Learning Systems Group, Pasadena, CA 91109, USA, [2]Institute for Genomics and Bioinformatics and [3]School of Information and Computer Science, University of California, Irvine, Irvine, CA 92697, USA

Received February 15, 2005; Revised March 25, 2005; Accepted April 6, 2005

ABSTRACT

Analysis of large-scale gene expression studies usually begins with gene clustering. A ubiquitous problem is that different algorithms applied to the same data inevitably give different results, and the differences are often substantial, involving a quarter or more of the genes analyzed. This raises a series of important but nettlesome questions: How are different clustering results related to each other and to the underlying data structure? Is one clustering objectively superior to another? Which differences, if any, are likely candidates to be biologically important? A systematic and quantitative way to address these questions is needed, together with an effective way to integrate and leverage expression results with other kinds of large-scale data and annotations. We developed a mathematical and computational framework to help quantify, compare, visualize and interactively mine clusterings. We show that by coupling confusion matrices with appropriate metrics (linear assignment and normalized mutual information scores), one can quantify and map differences between clusterings. A version of receiver operator characteristic analysis proved effective for quantifying and visualizing cluster quality and overlap. These methods, plus a flexible library of clustering algorithms, can be called from a new expandable set of software tools called CompClust 1.0 (http://woldlab.caltech.edu/compClust/). CompClust also makes it possible to relate expression clustering patterns to DNA sequence motif occurrences, protein–DNA interaction measurements and various kinds of functional annotations. Test analyses used yeast cell cycle data and revealed data structure not obvious under all algorithms. These results were then integrated with transcription motif and global protein–DNA interaction data to identify G_1 regulatory modules.

INTRODUCTION

A key step in analyzing most large-scale gene expression studies is clustering or otherwise grouping gene expression data vectors and conditions (individual RNA samples or replicates) into sets that contain members more similar to each other than to the remainder of the data. To do this, biologists now have at their disposal a wide range of computational techniques including supervised and unsupervised machine learning algorithms and various heuristics, such as *k*-means, phylogenic-like hierarchical ordering and clustering, Expectation Maximization of Mixture models, self organizing maps, support vector machines, Fourier analysis and more (1–6). Their purpose in all cases is to detect underlying relationships in the data, but different algorithms applied to a given dataset typically deliver only partly concordant results. As we show below, it is common to find 20–40% of genes from a high-quality dataset classified differently by two algorithms. These differences can be quite meaningful for a first-pass analysis, in which candidate genes will be selected based on their expression pattern for further detailed study. But clustering classifications are also increasingly important, not as results on their own, but as a key preprocessed input to higher level integrative modeling, such as gene network inference. Clustering results are also becoming important as gene annotations for interpreting entirely different kinds of data. For example, classification of genes as being 'cell cycle regulated in G_1 phase' has become part of major databases based on a specific clustering.

*To whom correspondence should be addressed. Email: woldb@caltech.edu

If such annotations are uncertain or simply incorrect, the uncertainty or errors then ramify through future uses of the data.

The sources of difference between clustering algorithm outputs are many and varied, and the biological implications are also diverse, as illustrated below for cell cycle data. The general challenge is to detect, measure, evaluate and mine the commonalities and differences. Specifically, the biologist usually wants to first know whether one clustering is objectively and significantly 'better' than another, and just how big the difference is. If two clusterings are of similar overall quality, yet differ substantially from each other as is often observed, then what specific gene cluster or samples harbor the greatest differences, or are they evenly distributed across the data? At a finer level still, which genes are being assigned to different clusters and why? Importantly, do the distinctions between clusterings highlight properties of biological importance?

To begin to answer such questions, we first needed a way to make systematic quantitative comparisons and then we needed ways to effectively mine the resulting comparisons. We use confusion matrices as the common tool for these comparisons (see below and Methods). A confusion matrix effectively summarizes pairwise intersections between clusters derived from two clustering results. These similarities are quantified by applying scoring functions to the confusion matrix. In this work, we use two different scoring functions for this purpose: (i) normalized mutual information (NMI), which measures the amount of information shared between the two clustering results (7) and; (ii) a linear assignment (LA) method, which quantifies the similarity of two clusterings by finding the optimal pairing of clusters between two clustering results and measuring the degree of agreement across this pairing (8,9). Previous studies have used metrics for evaluating the total number of data point pairs grouped together between two different clusterings to begin to address the need for quantifying overall differences (10–13). Ben-Hur *et al*. (13) used this to help determine an optimal number of clusters (K) and to assess the overall validity of a clustering. These prior techniques did not, however, offer the capacity to isolate and inspect the similarities and differences between two different clusterings, nor did they provide an interactive interface for biology users that would permit them to usefully capture the comparative differences and similarities. We also introduce a new application of receiver operator characteristic (ROC) analysis (14,15). As we use it here, ROC enables one to quantify the distinctness of a given cluster relative to another cluster or relative to all non-cluster members. Implemented in this fashion, ROC provides another measure of local cluster quality and shape, and provides another tool for quantitatively dissecting a cluster. Though the methods and tools were worked out for clusterings of large-scale gene expression data, they are applicable to clusterings of other kinds of large-scale data as well.

We have integrated the algorithms and comparative tools into an interactive analysis package collectively called CompClust 1.0. CompClust enables a user to organize, interrogate and visualize the comparisons. In addition to comparative cluster analysis, an important feature of this software is that it establishes and maintains links between the outputs of clustering analyses and the primary expression data, and, critically, with all other desired annotations. In the sense used here, 'annotations' include other kinds of primary and metadata of diverse types. This gives a biologist crucial flexibility in data mining and permits analyses that integrate results from other kinds of experiments, such as global protein–DNA interactions (ChIP/Array), protein–protein interactions, comparative genome analysis or information from gene ontologies.

CompClust methods and tools are agnostic about the kinds of microarray data (ratiometric, Affymetrix, etc.) and types of clustering algorithms used. We demonstrate the tools by analyzing two different sets of yeast cell cycle expression data representing both major data platforms, clustered by four different methods: a statistical clustering algorithm [Expectation Maximization of a Mixture of Diagonal Gaussian distributions (EM MoDGs)] (this work), a human-driven heuristic (1), a Fourier transform algorithm designed to take advantage of a periodic time-course patterns (16) and an agglomerative version of the Xclust phylogenetic ordering algorithm [Eisen *et al*. (2) modified in this work]. We show that gene groups derived from these comparative analyses can be integrated with data on evolutionarily conserved transcription factor binding sites to infer regulatory modules. These results begin to illustrate how a more quantitative and nuanced understanding of both global and local features in the data can be achieved, and how these can be linked with diverse kinds of data types to infer connectivity between regulators and their target gene modules.

METHODS

CompClust

The maturation of additional large-scale data types (global chromatin immunoprecipitation assays, more complete and highly articulated protein–protein interaction maps, Gene Ontology categories, evolutionarily conserved sequence features and other covariates) shifts the emphasis from analyzing and mining expression data alone to integrating disparate data types. A key feature of any system designed for integration is the ability to provide a many-to-many mapping of labels to data features and data features to other data features in a global way. CompClust provides these capabilities by maintaining and tracking linkages of multiple arbitrary annotations and covariates with data features through almost any data transformation, merger, selection or aggregation. In addition, many supervised and unsupervised machine learning algorithms are easily accessible within CompClust.

CompClust is primarily accessible through an application programming interface (API) and, with the use of Python's exposed interpreter, this provides a very rich command line interface (CLI). The major capabilities illustrated in this paper are accessible through CompClustTK, a set of simple graphical user interfaces (GUIs) to offer a convenient starting point without learning Python commands. These GUIs will permit users to perform the major classes of analyses shown, though we note that these comprise only a fraction of CompClust capabilities. The flexibility and diversity of analysis paths is too great to anticipate them all or commit them to GUI forms. This limitation can be overcome by using the Python command line environment. Python commands can be learned at the level needed in a relatively short time (a few weeks of part time effort) by users who do not have prior programming experience. The benefit is access to remarkable flexibility in interrogating datasets. This is a much better match to the

diversity of questions and comparisons that biologists usually want to make than in any GUI-based system.

The choice to implement CompClust in Python over other languages was made for several reasons which, considered in aggregate, argue it is the best available language to support the capabilities and analysis goals of CompClust: (i) using Python's exposed interpreter, our API becomes immediately useful for analysis without the construction of a complex GUI. The exposed interpreter also speeds the development time. (ii) Python's syntax is fairly straightforward and easy to learn for even non-programmers. (iii) It is freely available and distributable under an open-source license. (iv) Python has an extensive and standard library and in addition third party extensions, including the Numeric package which provides powerful numeric analysis routines. (v) Python is also platform neutral and runs on the majority of systems, including unix/linux, Microsoft Windows and the Mac OS.

Pairwise comparison of clusterings (partitions) using confusion arrays and matrices

Confusion arrays and matrices were used to make pairwise comparisons between different clusterings (mathematical partitions). A set of metrics were then applied to the confusion matrix to measure the nature and degree of similarity between two dataset partitions. Briefly, a confusion matrix is the matrix of cardinalities of all pairwise intersections between two partitions, where a partition of a dataset is defined as a set of disjoint subsets whose union contains all elements of the dataset. We define a confusion array simply as an array of all pairwise intersections between two partitions of a dataset. The cardinalities of these intersection sets form the confusion matrix, whose elements are given by Equation 1:

$$C_{i,j} = |A_i \cap B_j|, \quad \mathbf{1}$$

where A_i: the data members of class i in A, and B_i: the data members of class j in B.

Linear assignment. The LA value for a confusion matrix is calculated between two partitions (clusterings) and by generating an optimal pairing so that there is, at most, a one-to-one pairing between every class in partitions, and this pairing is calculated by optimizing the objective function in Equation 2, using the constraints given in Equation 3, thus defining a linear assignment problem. Next, the maximum-cardinality bipartite matching of maximum weights algorithm (Gabow, 1973) was implemented for the optimization. After finding the optimal pairing, the LA score is simply the proportion of vectors (e.g. gene expression trajectories or conditions) included in the optimally paired clusters (Equation 4). It is important to note that LA, unlike NMI, is a symmetric score so that LA(A,B) = LA(B,A). In addition to quantifying the degree of similarity or difference between two partitions, the adjacency matrix (Equation 3) also provides a way to identify pairs of clusters that are globally most similar to each other between two partitions of the data. As illustrated for clusterings of yeast cell cycle regulated genes, this is especially useful for interactive examination of two clusterings.

$$E = -\sum_{ab} M_{ab} C_{ab}, \quad \mathbf{2}$$

where,

$$M_{ab} \in \{0,1\} \wedge \sum_a M_{ab} \leqslant 1 \wedge \sum_b M_{ab} \leqslant 1. \quad \mathbf{3}$$

Now,

$$\mathrm{LA} = \frac{\sum_{a,b} M_{ab} C_{ab}}{\sum_{a,b} C_{ab}}, \quad \mathbf{4}$$

where M is the adjacency matrix describing the pairing between A and B, and C is the confusion matrix (Equation 1).

Normalized mutual information. The NMI index (7) quantifies how much information is lost, on average, when one clustering is regenerated from a second classification (Equation 5). A noteworthy difference from LA is that NMI is asymmetric.

$$\mathrm{NMI}(A,B) = \frac{I(A,B)}{H(A)} = \frac{H(A) - H(B) - H(A,B)}{H(A)} = 1 - \frac{H(A,B) - H(B)}{H(A)}, \quad \mathbf{5}$$

where $I(A, B)$ is the shared information between the two partitions and it is normalized by the entropy of partition A; $H(A)$ is defined as:

$$H(A) = \sum_{i \in \text{partitions}} p_i \cdot log \cdot p_i, \quad \mathbf{6}$$

and

$$p_i = \frac{\sum_j C_{i,j}}{n}, \quad \mathbf{7}$$

and the joint-information is:

$$H(A,B) = H(C) = \sum_j \sum_i \frac{C_{i,j}}{n} \log\left(\frac{C_{i,j}}{n}\right), \quad \mathbf{8}$$

and

$$n = \sum_{i,j} C_{i,j}. \quad \mathbf{9}$$

EM MoDG clustering

EM MoDG was implemented with a diagonal covariance matrix model because the number of samples in the (1) cell cycle dataset was too small to fit a statistically valid full covariance matrix to each cluster (17). In order to ensure a near optimal initialization, each EM MoDG result was a result of selection of the best of 30 runs, each initialized by placing the initial cluster centroids on K randomly selected data points. The run with best fit to the data (i.e. had the lowest log-likelihood score) was used for the final clustering. Multiple best-of-30 runs were performed to verify that the quantitative measures and gene lists results reported here did not vary significantly. The EM MoDG code used here was developed by the NASA/JPL Machine Learning Systems Group.

XclustAgglom

We agglomerate the hierarchical tree returned by Xclust (18) based on a maximal cluster size threshold. Starting from the root, any subtree within the tree with less than the maximal cluster size threshold is agglomerated into a cluster. In order to work with the familiar parameter K (number of clusters), we iteratively find the size threshold that will return as close to K clusters as possible. In practice, this simple heuristic works best when K is over specified by ~2–4 times the expected number of clusters because it will generate several very small (often singleton) clusters that are outliers to core major clusters in the data.

Data preprocessing

Each microarray dataset was obtained from the cited authors. For the Cho *et al*. (1) data, we removed any gene that did not show a sustained absolute expression level of at least 8 for 30 consecutive minutes. For each gene vector, we then divided each time point measurement by the median expression value for the gene. For the data of Spellman *et al*. (16), we linearly interpolated missing values using the available adjacent time points. For both datasets, we $\log_2$ transformed the resulting gene expression matrices. The datasets were then annotated with the original clustering results as published.

Motif conserved enrichment score (MCS)

For each motif, we translated the IUPAC consensus (Swi5/Ace2: KGCTGR; MCB: ACGCGT; SCB: CACGAAA) into a position weight matrix (PWM) where the probabilities or frequencies in the PWM is determined by the degeneracy of the IUPAC symbol. We calculate a log-odds ratio for the PWM occurring at every position in the 1 kb upstream as described in Equation 10

$$\mathrm{MCS} = \frac{1}{N} \sum_{\forall windows} \frac{\prod_{i=0}^{W} p_{ni}}{bg} \qquad \mathbf{10}$$

of each open reading frame (ORF) for each species available. We then sum the log-odds ratio over all possible positions, where the log-odds ratio is >7. The summed log-odds ratios for each species is then averaged together to generate an ORF-specific motif enrichment score. In Equation 10, N is the total number of species compared, W is the length of the motif, p is the probability from the PWM of position i being the nucleotide n, and bg represents the probability of the window being generated from a background sequence model based on a second-order hidden Markov model.

RESULTS

Mathematical tools for organizing and quantifying microarray clusterings: confusion matrices and comparative metrics

A confusion matrix can effectively summarize all pairwise intersections between all clusters from any two clusterings of the same data. Confusion matrices, as used in this work, are defined as the matrix of cardinalities of all pairwise intersections between two different expression clusterings (see Methods). Once the confusion matrix is constructed, we can then apply different scoring functions to the confusion matrix to quantify similarity: (i) NMI measures the amount of information shared between two clusterings (7); and (ii) LA optimizes the number of data vectors in clusters that correspond to each other, thereby identifying the optimal pairing of clusters. LA also reports the percentage of data vectors contained within those clusters, and this can be used to assess similarity of results globally over the entire dataset and locally on a cluster pair by cluster pair basis (8,9) (for mathematical descriptions of confusion matrices, NMI and LA, see Methods). The combined use of LA and NMI metrics can provide a biologist with immediate insight into the magnitude and nature of global differences between two microarray clusterings by capitalizing on the fact that NMI is asymmetric and LA is symmetric (see Table 1 for details). Specifically, this discriminates instances in which two clusterings are very similar to each other from a comparison in which one clustering is different from the other, but is essentially a refinement of the first. Both of these can be discriminated from the third relationship, in which two clusterings deliver fundamentally different views of the data structure.

Table 1. Interpretations of commonly observed combinations of LA and NMI scores

NMI(A,B)	NMI(B,A)	LA	Implies
Low	Low	Low	Poor similarity
Low	High	Low	B refines A
High	Low	Low	A refines B
High	High	High	Good similarity

Given two clustering results A and B, for which both NMI(A,B), NMI(B,A) and LA(A,B) values are high (nearing the maximum value of 1.0), the two clusterings are very similar, and when all three are significantly lower, they are very different. But when NMI(A,B) is high, NMI(B,A) is low and LA is low, then it is likely that A is a refinement of B. In this case, many clusters in B have been broken into two or more clusters in A (possible combinations summarized in here) (1). The magnitude of dissimilarity that is important is defined by the user and may vary considerably with the dataset, although values <0.7 for both LA and NMI are usually viewed as quite different. Additional interpretation of differences measured by LA and NMI depends on more detailed analysis of the dissimilarities and their distribution over the dataset, as outlined above.

Confusion arrays organize and display comparative analyses. Given two different clusterings of a dataset and a global evaluation of their similarity via NMI and LA, we then needed a method to systematically compare clusters in a manner that is more effective and intuitive than mere inspection of gene lists. To do this, we define the confusion array, which is a direct extension of a formal confusion matrix. For two different clusterings, each cell in the confusion array contains the intersection set between the two parent clusters (as opposed to the cardinality of this set, as in a confusion matrix; see Methods). In the context of the CompClust system, the confusion array cells can then be interactively mined. Confusion arrays for two different clusterings, one using an Affymetrix yeast cell cycle dataset (1) and the other using a deposition ratiometric dataset (16), are shown in Figures 1 and 2, and are analyzed further below.

Understanding cluster relatedness

ROC measures cluster overlap. Whatever algorithm has been used to cluster data, it is useful to find out how distinct each cluster is from all the others and how distinct any particular cluster is from another specific cluster. This is especially

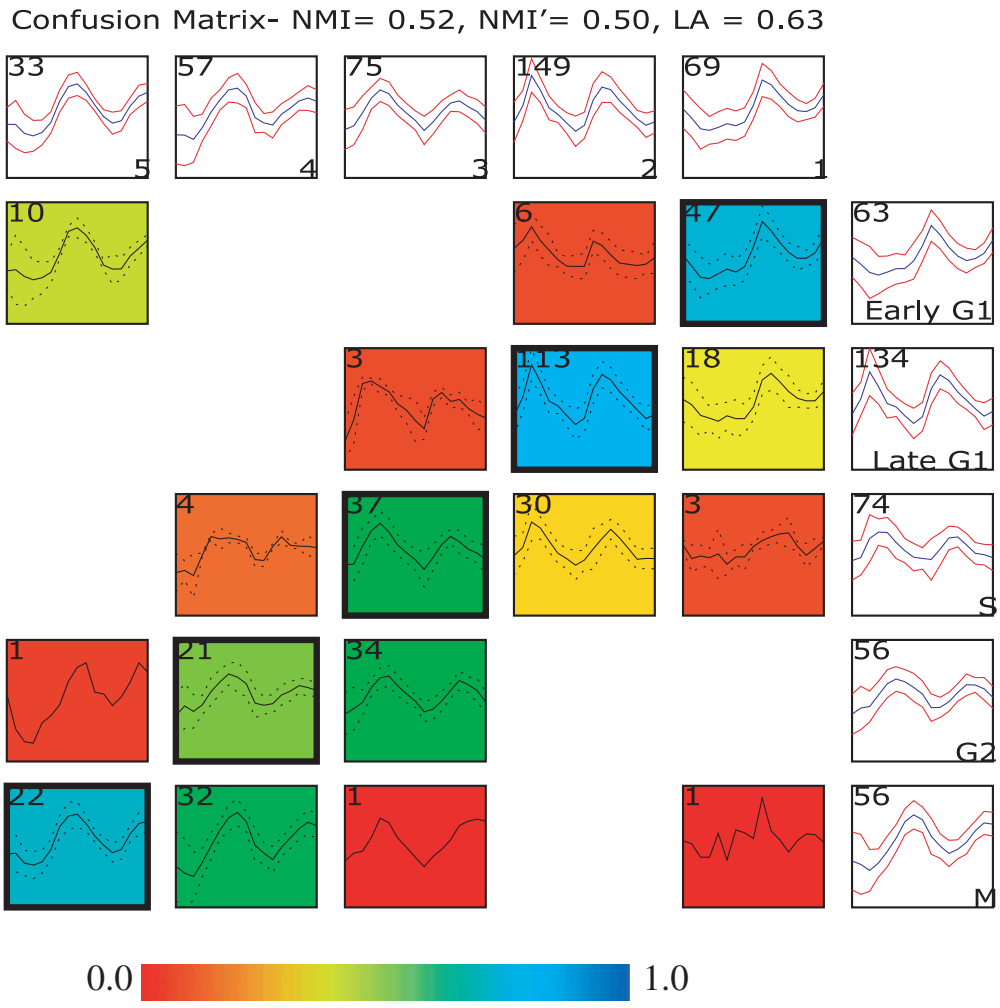

Figure 1. Comparing two clustering results using a confusion array. Shown in this comparison is a supervised clustering result published in the original study by Cho *et al*. (1) and results from running an unsupervised clustering (EM MoDG, see Methods) on the same Affymetrix microarray dataset profiling yeast gene expression through two cell cycles. The confusion array is composed of a grid of summary plots. Each summary plot displays the mean (blue color or solid line) expression level of a group of genes as well as the standard deviation (red color or dashed line). Summary plots with a white background represent clusters from either the Cho *et al*. (1) clustering result (along the right most column) or the EM MoDG clustering result (along the top row); cluster names are in the lower right corner; and the number of genes in each cluster is displayed in the upper left corner. Summary plots with a colored background represent cells within the confusion array (see Methods), where each cell represents the intersection set of genes that are in common between the Cho *et al*. (1) cluster and the EM MoDG result cluster. Again, the upper left hand corner displays the number of genes within a confusion matrix cell. The background of each plot is colored according to a heat-map (scale below) that registers the proportionate number of genes in the cell compared with the corresponding cluster in the EM MoDG result. Intersection cells with dark outlines indicate the optimal pairings between the two data partitions, as determined from the LA calculation (Equation 2). Quantitative measures of overall similarity between the two clustering results using both LA and NMI are displayed in the graph title (see Methods).

pertinent when membership in a cluster will be translated into a gene list that ultimately becomes a functional annotation or defines genes that will be input into higher-order analyses. To address this issue, we applied classical ROC analysis (see Methods). In this context, cluster assignment is used as the 'diagnosis' and the distance of each expression vector from the cluster mean vector is the 'decision criterion'. The corresponding ROC curve plots the proportion of cluster members versus the proportion of non-cluster members as the distance from the cluster centroid increases (Figure 3). This can be interpreted geometrically as expansion of a hypersphere from the cluster centroid until all members of the cluster are enclosed. Thus, when one cluster is completely separate from all other data, all of its members are closer to the cluster center than all non-members and the area under the ROC curve is 1.0 (Figure 3B). When a cluster is not fully separable from the remainder of the data, the ROC curve rises more slowly and the area under the ROC curve is <1.0. In the limit, when the two classes are perfectly mixed, the ROC curve closely follows $X = Y$, and the area under the curve drops to 0.5 (Figure 3D). The shape of the ROC curve also contains additional information about how cluster overlap is distributed, and this information can be used by the biologist to choose useful data mining cut-offs that mark discontinuities and cluster substructure (see below and Figure 4). It can also be used interactively within CompClust to explore and select data vectors (genes) that are closer or more distant from the cluster center. Selection of vectors not assigned to the cluster, yet positioned at overlapping distances from its center, is also possible and is often instructive (Figure 4 and text below).

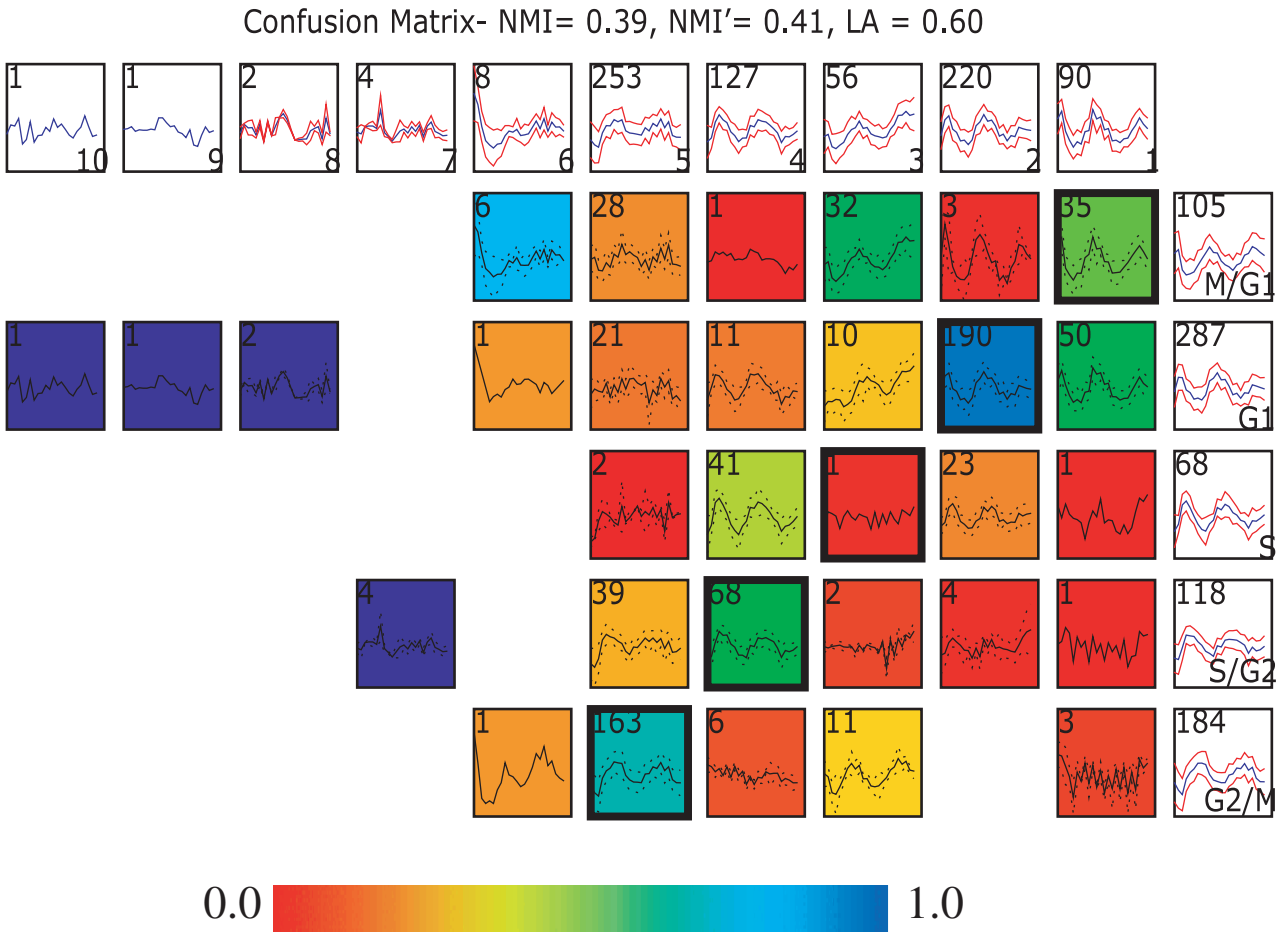

Figure 2. Comparing two clustering results on a ratiometric microarray dataset using a confusion array. Shown in this comparison is a Fourier clustering result published in the original study by Spellman *et al*. (16) and results from running an unsupervised clustering (Xclustagglom, see Methods) on the same ratiometric microarray dataset as the Fourier analysis was run on. Details of the figure layout are discussed in the legend of Figure 1. Here, the 5 Fourier clusters are shown along the rows, while the 10 Xclustagglom clusters are displayed across the columns.

Comparing clusterings of yeast cell cycle microarray datasets

We performed comparative analyses on clustering results from two different yeast microarray time-course datasets (one Affymetrix and one ratiometric), each composed of genes that are differentially expressed over the cell cycle (1,16). These comparisons provide a useful perspective because the gene classification results from the original gene clusterings have been introduced as gene annotations in widely used databases [Incytes' YPD (19), SGD (http://yeastgenome.org)] and have been mined or used as starting points in many subsequent works. We generated a new clustering for each dataset, in each instance selecting an algorithm that differs substantially from the one used in the original publication but one that should also be entirely appropriate for the dataset. For the Cho *et al*. (1) dataset, we used EM MoDGs (17), which is an unsupervised method that searches for the best statistical fit to the data modeled as a mixture of Gaussian distributions. The heuristic used in the original report (1) is a supervised method based on biologist's knowledge of cell cycle phases. The heuristic focused on the time of peak expression for each gene trajectory to guide assignment of each gene to one of the five time domains associated with Early G_1, Late G_1, S, G_2 and M phases of the cell cycle. For the second dataset (16), we performed an agglomerative phylogenic hierarchical clustering of the tsCDC15-mutant synchronized data. This algorithm is based on the widely used Xclust phylogenetic ordering algorithm (2), onto which we grafted an agglomeration step designed to establish objective boundaries in the tree (see Methods). This effectively turns an ordering algorithm into a clustering algorithm, in which group boundaries are imposed computationally rather than by a user's pattern recognition skills, as is performed with Xclust by itself. This result was compared with the result reported by Spellman *et al*. (16), in which they used a Fourier transform-based algorithm, which was designed to take maximal advantage of the time-course pattern.

Global similarity measures

Comparison of the two clusterings of Affymetrix data from Cho *et al*. (1) gave a global LA score of 0.63 and NMI scores of 0.52 and 0.50, immediately indicating that EM MoDG and the heuristic classification have produced substantially different results. The LA value of 0.63 says that the optimal pairing of clusters still classifies 37% of the genes differently between the two algorithms. ROC curves and ROC areas were

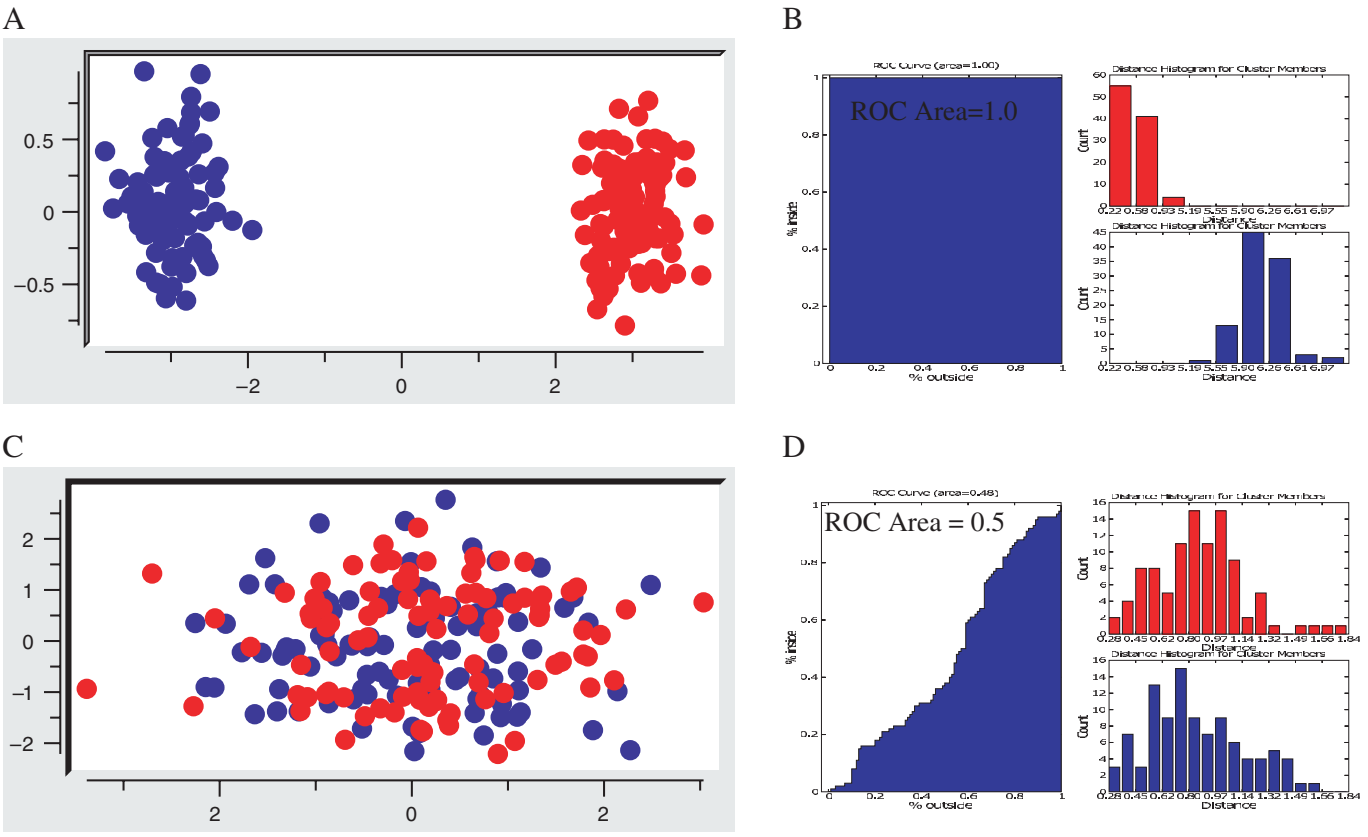

Figure 3. Example ROC curves to assess cluster overlap. An ROC curve (B and D, left side) is drawn as a function of moving outward from a cluster center and counting the proportion of cluster members (blue points) encountered along the *y*-axis versus the proportion of non-cluster members (red points) encountered along the *x*-axis. The collection of distances from every point within a cluster and every point outside a cluster is binned and used to create the distance histograms (B and D, right side). Shown in red is the distance histogram for cluster members and cluster non-members are shown in blue. Two extreme cases are exemplified in this figure. (**A**) Example expression data falling into two completely discrete clusters highlighted in red and blue. (**B**) The corresponding ROC curve (left) and distance histograms (right) for the sample data shown in (A). Note that since all cluster members are encountered before any non-cluster members the area under the ROC curve is 1.0. The distance histograms also show this perfect separation. (**C**) Example expression data falling into two completely overlapping clusters highlighted in red and blue. (**D**) The corresponding ROC curve (left) and distance histograms (right) for sample data shown in (B). Note that since cluster members and non-cluster members are encountered at an equal rate as a function of distance from the cluster center, the ROC curve approximates the line $x = y$ and the area under the ROC curve is 0.5. This overlap is also highlighted in the distance histograms because the distributions of distances for cluster members completely overlap with that of the distribution of distances for non-cluster members.

generated for each cluster (Figure 5). Viewed in aggregate, this ROC analysis showed that clusters from EM MoDG are all better separated from each other than are any clusters from the original Cho *et al*. (1) heuristic. Thus, the ROC indices for EM MoDG are all 0.96 or above, and four of the five clusters are >0.98. In contrast, the heuristic classification groups had ROC values as low as 0.82 for S phase and none was better than 0.97 (M phase). By this criterion, we can argue that EM clustering is an objectively superior representation of the underlying data structure.

How are these differences between clustering results distributed over the dataset? We used PCA (Principle Component Analysis) to determine whether the two clusterings were globally similar or different in the way they partitioned the dataspace. PCA projects high-dimensional gene expression vectors (each dimension here corresponding to a different RNA sample) into a different and low-dimensional space (usually two or three) (20), in which the new PCA dimensions have each been selected to explain the maximal amount of variance in the data. A common feature of microarray datasets is that the first few principle components often capture most of the variations in the data (here 64%). Using CompClust to view the cluster means in PCA space allowed us to assess relationships between clusters from the two algorithms. Relative positions of cluster means in the PCA display the cell cycle progression in a counterclockwise pattern that is quite similar for the two algorithms. The absolute positions of the cluster centers in PCA space differ, though not extravagantly, for most clusters. This is interesting because the coherence in overall structure would seem to contradict the rather high dissimilarities in cluster composition measured by the criteria LA and NMI, and shown graphically in the confusion array (Figure 1). Considered together, the results argue that the overall data structure, reflecting phases of the cell cycle, is robust and has been treated rather similarly by the two algorithms, even though 37% of individual gene expression vectors were assigned differently. This raises the question of which gene vectors have been differentially assigned and what biological meaning, if any, should be attached to the differences. These questions are addressed in the Discussion by examining specific gene groups in the confusion array.

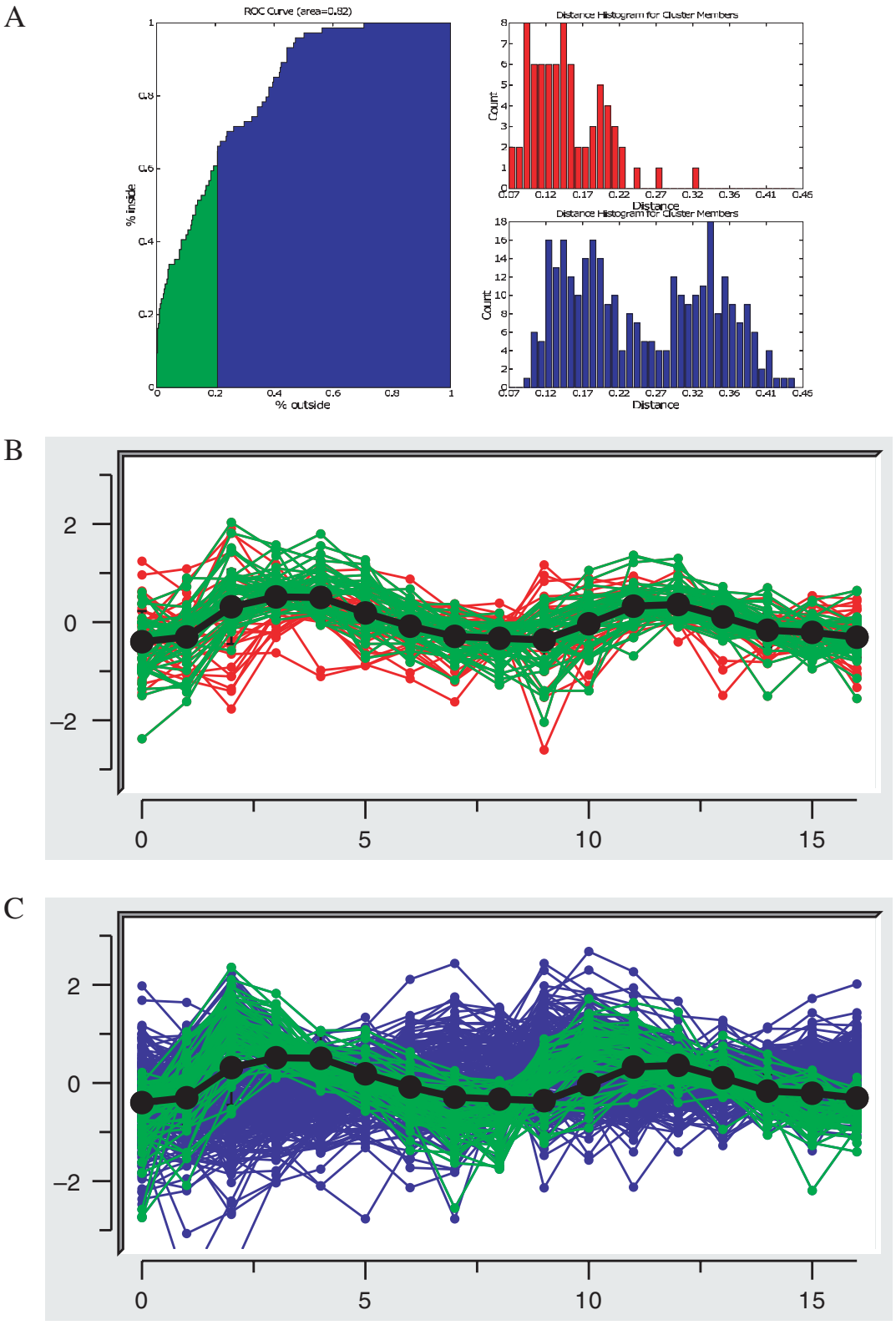

Figure 4. ROC analysis of the S-phase cluster of Cho *et al*. (1). (**A**) ROC curve (left) shows the overlap between this cluster of 74 genes and genes from all other clusters in the time-course analysis [383 genes in total, selected by inspection by Cho *et al*. (1) for cycling behavior]. The area under the ROC curve is 0.82. The area under the curve highlighted in green demonstrates selection of genes from S phase that overlap with other clusters least. At the shown distance threshold, 66 genes from the Cho determined S-phase cluster are selected, and the overlap with only non-S-phase genes. (**A**) Right: correlation distance histograms illustrating the distribution of distances to the center of the S-phase cluster for non-cluster members (bottom/blue) and for all S-phase cluster members (top/red). (**B**) Expression trajectories for the 74 genes in the S-phase cluster, highlighting in green cluster members represented by the green highlight in (A). (**C**) Expression trajectories for all genes outside the S-phase cluster of the Cho clustering highlighting in green non-cluster members represented by the green highlight in (A).

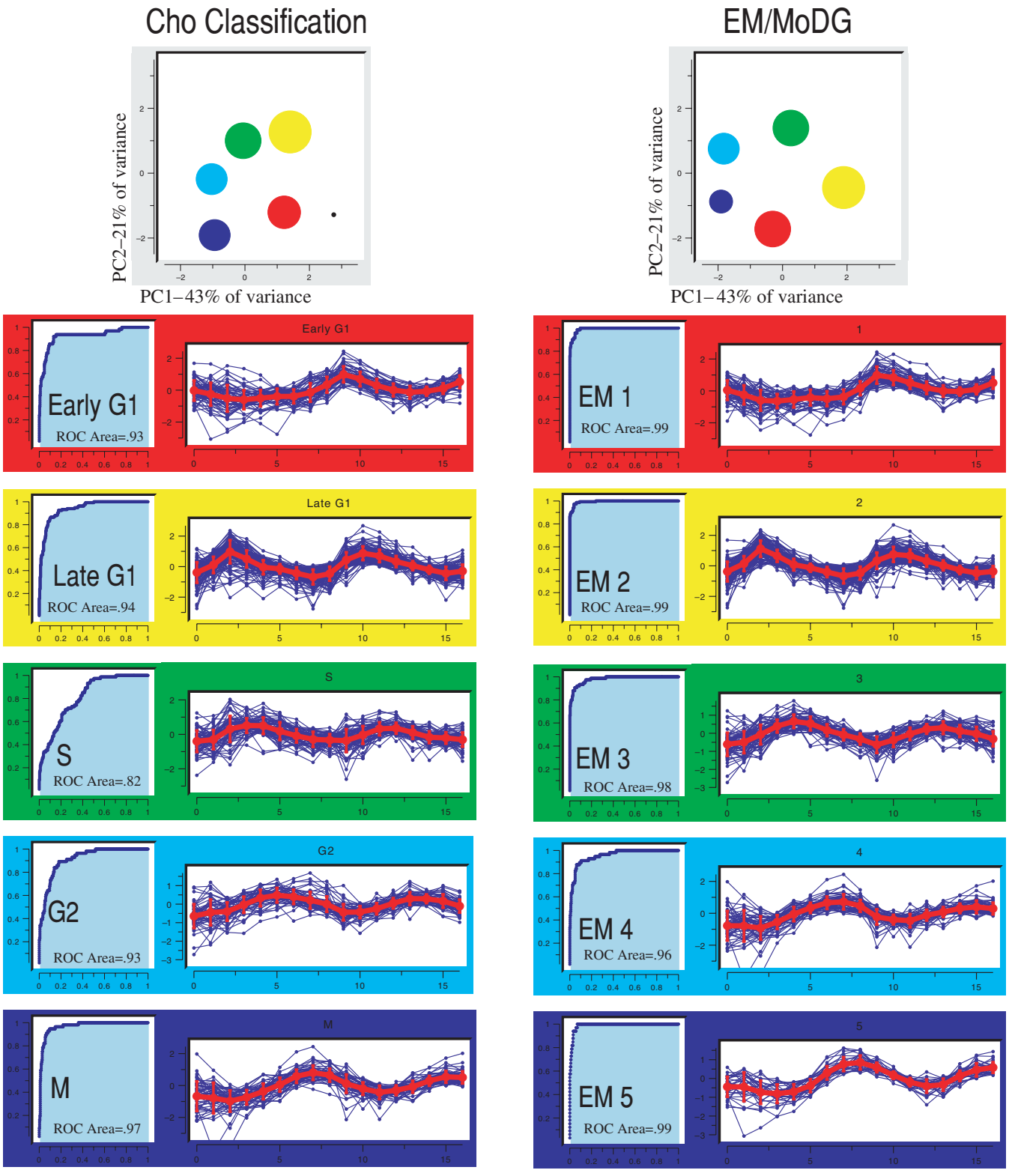

Figure 5. PCA, ROC plots and trajectory summary views of clusters from the Cho classification and an unsupervised clustering (EM MoDG) of an Affymetrix yeast cell cycle time course (1). The top panel for each clustering results shows cluster means projected into the top two dimensions of the principle component space defined by the expression data (capturing 64% of the variance). The area of the marker size for each cluster is proportional to the number of genes in each cluster. Below are ROC curves (left) and trajectory summaries (right) for each cluster. The trajectory summaries display every gene's expression profile within a cluster as a blue line with time along the *x*-axis and expression along the *y*-axis. The red line within each trajectory summary represents the mean expression level for the cluster. ROC area values are displayed within the ROC curve for each cluster. The background colors for the trajectory summaries and the PCA projection have been matched within each clustering result. In addition, LA was used to find the optimal mapping of clusters between the Cho classification and the EM MoDG result and the colors have been set accordingly.

Using the ratiometric data of Spellman *et al*. (16), a comparison of the original Fourier-based algorithm versus agglomerated Xclust produced NMI and LA scores of 0.39, 0.41 and 0.60, respectively. These scores indicate that the two clusterings are even more different in membership assignment, with 40% of genes falling outside the optimal LA pairing. Since both NMI and LA scores are low, gene memberships for some clusters must be truly scrambled rather than being simple combinations of cluster unions and subsets (see Methods and Table 1). PCA projection (Figure 6) showed that some major cluster centers from the two algorithms are positioned very differently, both absolutely and relatively (note the yellow cluster corresponding to the Fourier S-phase group). The confusion array shows that XclustAgglom clusters often combine genes that are members of adjacent Fourier clusters, though in some cases it joins vectors from non-adjacent groups (the confusion around S phase is complex). ROC curves and scores also indicate that XclustAgglom has performed a slightly better job of segregating data into discrete groups that reflect underlying data structure, while the Fourier analysis groups are less coherent and often seem to mix members of kinetically adjacent groups as detailed in the confusion array (Figures 2 and 6). This may be due, in part, to the use of a small number of then known genes to center landmark phases by the Fourier algorithm. The fact that this phase assignment was a 'somewhat arbitrary' step in the original analysis was pointed out by Spellman *et al*. (16).

High-resolution cluster comparisons

Confusion arrays can be used to explore in more detail the issues raised by global analyses and to mine relationships between individual clusters. This can then be used to make refined gene lists based on either the expert opinion or on the application of computationally objective criteria. We applied LA to the confusion matrix of the Cho heuristic and the EM MoDG results and produced the corresponding adjacency matrix (see Methods and Equation 3). This delivered an objectively optimized pairing of EM cluster 1 with Cho 'Early G_1', EM cluster 2 with Cho 'Late G_1' and so on, as shown in the array visualization (Figure 1). Each cell in the confusion array contains the corresponding gene vectors and displays the calculated mean vector for each intersect cell in the array.

The confusion array highlighted relationships that were not clear from Figures 5 or 6. For example, in the Affymetrix data (1), both algorithms identified two gene classes within G_1 (red and yellow, respectively, in the PCA analysis of Figure 5). However, the EM1 cluster shares only 67% of its content with the Cho 'Early G_1', and most remaining genes fall into the Cho 'Late G_1' cluster (Figure 7A). A straightforward hypothesis is that the statistical EM algorithm simply could not justify dividing G_1 vectors into early and Late G_1 kinetic groups as the heuristic had done. The confusion array, however, makes it clear at a glance that a different data feature is driving the G_1 sub-groupings. EM1 cluster members are upregulated only in the second cycle, while EM2 genes are upregulated in both cycles. The array also shows that the Cho Early G_1 group contains a set of 10 genes that appear much more consistent with a coherent M-phase group that corresponds to EM5.

Because the focus of the heuristic classification was mainly on the second oscillation, it suppressed the distinction between single cycle and two cycle G_1 patterns, while 'paying more attention' to fine-structure kinetic differences of the second cycle. On the other hand, EM MoDG treats all features with equal weight across the time course and so centers clusters without prior guidance about their relationship to cell cycle phase. The confusion array intersect cells then effectively parsed the fine kinetic differences within EM1 by separating 47 vectors that more closely resemble the Early G_1 cluster from 18 that are more like Late G_1 cluster. Thus, the intersect cells capture and dissect the two distinct ways used by the two algorithms to parse G_1 expression. Both appear valid and reveal in different ways, one based entirely on the kinetics of the second cycle and one focusing on a major difference in the expression that is seen only in the first cycle following release from arrest.

Turning to the S-phase clusters, the comparative analysis highlights a different kind of disagreement between the two algorithm outputs. Members from Cho 'S-phase' cluster overlap almost evenly with either EM2 (41%) or EM3 (49%). A simple biological interpretation is that the kinetic boundary between Late G_1 and S-phase is not very crisp, irrespective of the algorithm used to try to define them. An alternate explanation is that one algorithm is frankly superior to the other in defining coherent expression groups. Examination of the ROC curves (Figure 5) and values (0.98 for EM versus 0.82 for S phase) provided an objective measure of quality that argues the EM clustering of S phase is superior.

Dissecting individual clusters using ROC

Further ROC-based analysis of the S-phase cluster from the Cho *et al*. (1) classification is shown in Figure 4. The ROC curve shows how far from the cluster mean one needs to expand a hypersphere to include a given fraction of vectors from the cluster, and the shape of the curve can be used to understand cluster substructure. Inspection of the ROC curve and the corresponding histogram (Figure 4A) identified a natural discontinuity separating the first 66% of genes that are nearer the cluster center from the remainder. For additional data mining, we therefore set a boundary at 66% on the ROC curve and then inspected all gene vectors from the entire cell cycle dataset that fall within that boundary. Approximately 20% of gene vectors inside this dataspace threshold had been assigned to other clusters. Figure 4B and C allows inspection of gene trajectories that were either interior or exterior to the boundary. This is useful for reviewing and 'pruning' lists of putatively co-expressed genes in an objective manner.

Integration with transcription factor motifs to identify regulatory modules

CompClust is designed to integrate different kinds of data by linking each gene with other data, annotations and results of meta-analyses. There are many ways of using other datasets to identify relationships between, for example, observed patterns of RNA co-expression and other data that help to answer the question: are similarly expressed genes co-regulated? A group of genes that are co-expressed may also be co-regulated, but this is far from assured. Co-expressed genes can instead arrive at the same expression pattern by the action of two (or more) different regulators. Conversely, genes that are co-regulated by the same factor(s) at the transcriptional level may not

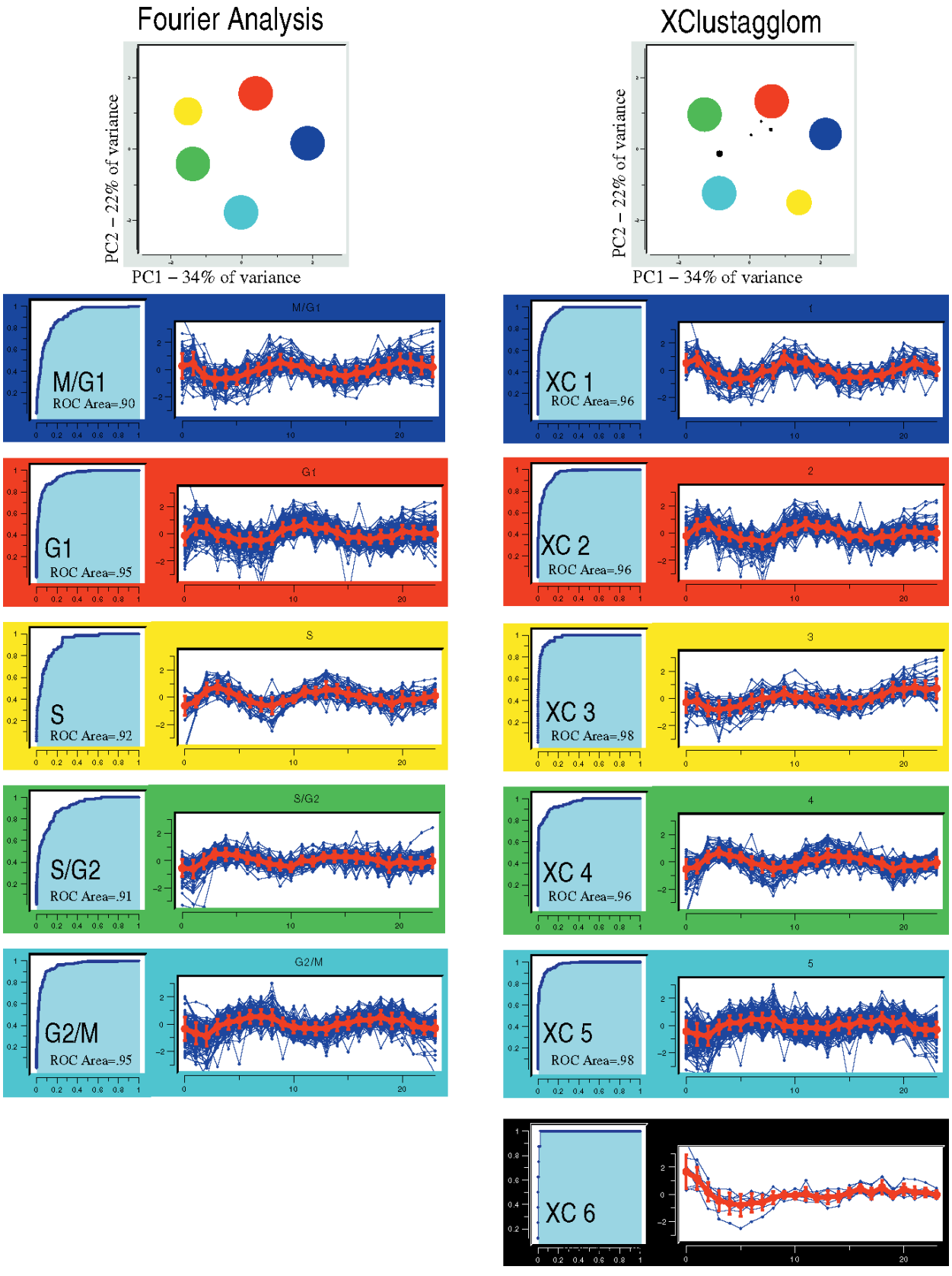

Figure 6. PCA, ROC plots and trajectory summary views of clusters from the Fourier classification and an unsupervised clustering (Xclustagglom) results from the ratiometric yeast cell cycle time course (16). Details of the figure layout are the same as for Figure 5. Only the six largest clusters are shown in the Xclustagglom. Clusters that do not have an optimal pairing by LA with a Fourier cluster are colored black. Note that PCA summary calls attention to the low quality of the S/XC3 pairing, places it between XC5 and XC1.

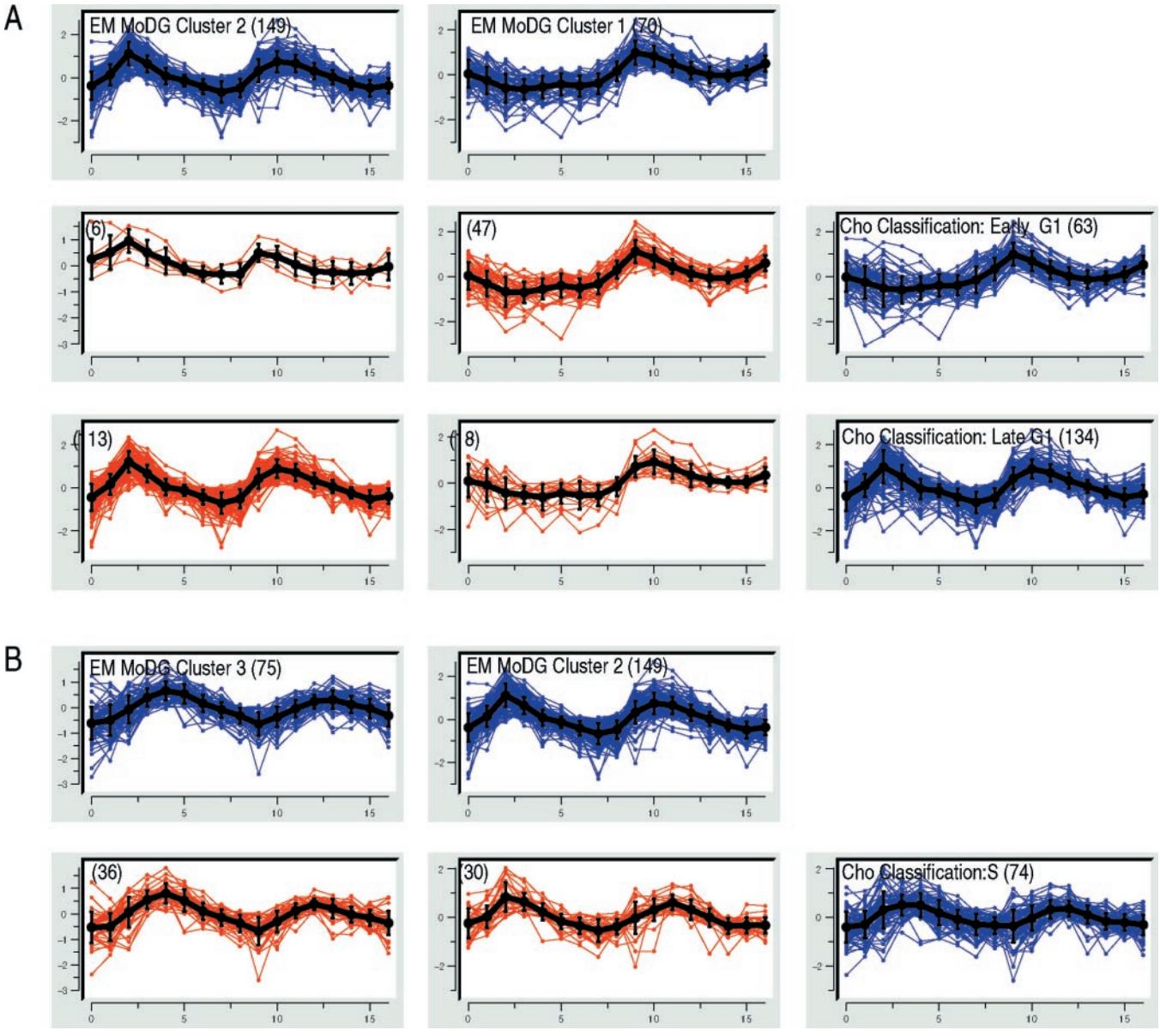

Figure 7. Selected confusion array cells from Figure 1 highlighting cluster membership differences for genes with peak expression during the G_1 and S phases of the cell cycle. The trajectory summaries display an expression profile for every gene with time along the *x*-axis and expression along the *y*-axis. Blue trajectory summaries show parent clustering results (EM MoDG along the columns, the Cho classification along the rows). Intersection cells from the confusion array are shown in red. Mean vectors for each gene set are shown in black with error bars proportional to the standard deviation. The total number of gene expression vectors in each cell is shown in parentheses. (**A**) G_1 genes are subdivided differently by the two algorithms. EM MoDG separates genes upregulated only during the second phase of the cell cycle from those upregulated during both the first and second cycles. The Cho classification separates G_1 based primarily on peak time in the second cycle. Figure 8 illustrates these observed kinetic distinctions being a result of these genes belonging to distinct regulatory modules. (**B**) Detailed comparison from the confusion array of Figure 1 showing the S-phase cluster of the Cho classification is subdivided nearly equally among EM2, EM3 (optimal match by LA) clusters.

display identical RNA expression patterns for a variety of reasons, including differential turnover rates. For these reasons, additional kinds of data are needed to help determine the co-expressed genes that are, in fact, transcriptionally co-regulated and to provide evidence for the identity of factor(s) driving co-regulation. Here, we show how the occurrence of evolutionarily persistent transcription factor binding sites can be mapped informatively onto the gene expression clusters from a confusion array to predict the structure of transcription modules.

The observation of two distinct sets of genes, one that peaks during both the first and second cell cycles after release from arrest, and another restricted to only the second oscillation (Figure 7), suggests that they might be regulated differently at the level of transcription. Prior work has led to the view that MCB and SCB sequence motifs bind Mbp1/Swi6 (MCF) or Swi4/Swi6 (SCF) factor complexes to drive G_1-specific transcription (21–23). Thus, many genes are believed to be selectively and specifically expressed in G_1 owing to their membership in either MCF or SCF regulatory modules. The

two modules are also thought to be partly distinct from each other, with some genes apparently being strongly governed by either Swi4 or Mbp1 [(24,25) and reviewed in (26)].

We, therefore, calculated an MCS (see Methods) to quantify the conserved enrichment of a consensus site within 1 kb of the start ATG in sequence data from the seven available yeast genomes (27,28). We then asked whether different intersecting cells within the confusion array are differentially and significantly enriched for these known candidate motifs. The EM2/Late G_1 intersect cell was highly enriched, above chance, for MCB and SCB. A total of 79 of 113 genes (70%) were enriched for MCB compared with the expectation of 13 such genes for randomly selected samples of 113 yeast genes. A total of 18% are enriched for SCB sites compared with an expectation of only 4% by chance (Figure 8A). CompClust's data linking capabilities were then used to visualize correlations with *in vivo* protein–DNA binding data for Swi4 and Mbp1 (29). The vast majority of genes with the above threshold MCB or SCB MCS scores also showed significant *in vivo* binding activity for either MCF or SCF.

The picture for the EM1/Early G_1 intersect cell, whose genes peak only once during the time course, was surprisingly different. This group showed no significant enrichment for either MCB or SCB (Figure 8A). What other factor(s) could be responsible for the EM1/Early G_1 intersect pattern? We searched and found that the Swi5/Ace2 motif is enriched so that ~30% are above threshold, a value twice that expected by chance. The highest Swi5 MCS scores correlated strongly with very intense expression in the second cycle. This, in turn, correlated well with *in vivo* factor binding by both Swi5 and Ace2 taken from the chromatin immunoprecipitation data of Lee *et al*. (29). Thus, 60% of the EM1/Early G_1 Swi5/Ace2 group had *P*-values <0.05 for Swi5 or Ace2 in the global chromatin immunoprecipitation study of Lee *et al*. (29), and others were relatively strong binders as shown in Figure 8. That most or all of these connections between Swi5/Ace2 and EM1 genes, inferred from three kinds of genome scale data are real is supported by the fact that most previously identified targets of Ace2 and/or Swi5 (30,31) that were in the original Cho cycling dataset are in this group.

DISCUSSION

As illustrated for yeast cell cycle data, differences among clustering algorithms and individual dataset structures make it difficult, and limiting, to simply select one clustering result and expect it to produce a fully informative data model. In the absence of ways to make objective comparisons or to mine comparisons, it has until now been exceedingly difficult to tell by inspection whether one clustering is significantly 'better' than another or to dissect differences between results in a systematic manner. The mathematical, computational and visualization tools that collectively comprise CompClust allow one to run diverse unsupervised and supervised clustering algorithms, compare the results using unbiased quantitative tools and then dissect similarities and differences between specific clusters and between entire clusterings. Specifically, we showed that LA and NMI metrics, ROC analysis, PCA projections and interactive confusion array analysis can be combined to provide a powerful comparative analysis.

By coupling the resulting comparative analyses with a flexible visualization system within CompClust and, especially, by using confusion arrays to organize comparisons, it became relatively straightforward to identify both global and local trends in expression patterns and to find out the features that are fragile to algorithm choice or to other variations. The tools were also useful for investigating substructure within individual gene clusters and for seeing how a cluster from one analysis relates to a cluster from another analysis. CompClust, including all source code, and associated tutorials are available at http://woldlab.caltech.edu/compClust/. The principal capabilities presented here can be used through the GUI of CompClustTK. The tools are introduced by tutorials that use the cell cycle examples presented here. Much richer and almost infinitely varied interactive interrogations can be performed using the command line version of CompClust, which is available for download. And while this demonstration is centered on clustering of large-scale expression data from microarrays, it will be applicable to clusterings of protein interaction measurements, protein–DNA interactions and other large-scale data types.

Comparative analysis showed that for the Affymetrix dataset (1), EM MoDG and the Cho heuristic found a basic data structure dominated by, and consistent with, the major phases of the cell cycle (Figure 5). This presented an apparent paradox, since the overall cell cycle phase structure was highlighted similarly by both algorithms, while the assignment of specific gene vectors to individual clusters was quite different, as shown by the LA and NMI scores (Figure 1). Further investigation of cluster relationships in the context of confusion arrays, local ROC analysis, and ROC curve structure helped to resolve the paradox. In specific cases, ambiguity was simply a data quality issue for particular gene data vectors, and highlighting these affords a biologist the opportunity to trim gene lists based on expert knowledge. In other cases, differences between algorithms portrayed correctly the fact that phases of the cell cycle are not crisply separated with respect to both mRNA synthesis and decay. This leads to a continuum of time-course profiles, especially around S phase. This knowledge of 'fuzzy kinetic boundaries' is important for future uses of gene categories for subsequent gene network modeling. In yet other specific parts of the clustering, the differences between algorithms focused attention on different and valid ways of parsing the data, as in the case of genes regulated strongly by Ace2/Swi5 in G_1, which were separable by one algorithm but not by another.

Inference of transcription modules

A key capability of the CompClust computational framework is that it provides the means to integrate many different kinds of data via linking properties (see Methods). This then allows the biologist to detect, organize and further mine relationships. The manner in which relationships, such as a direct connection between a transcription factor and one of its target genes, can be defined and vetted (by other data) is flexible, so that users can specify significance thresholds and apply diverse comparative metrics of their choice. They may also export CompClust data for further automated modeling, e.g. artificial neural

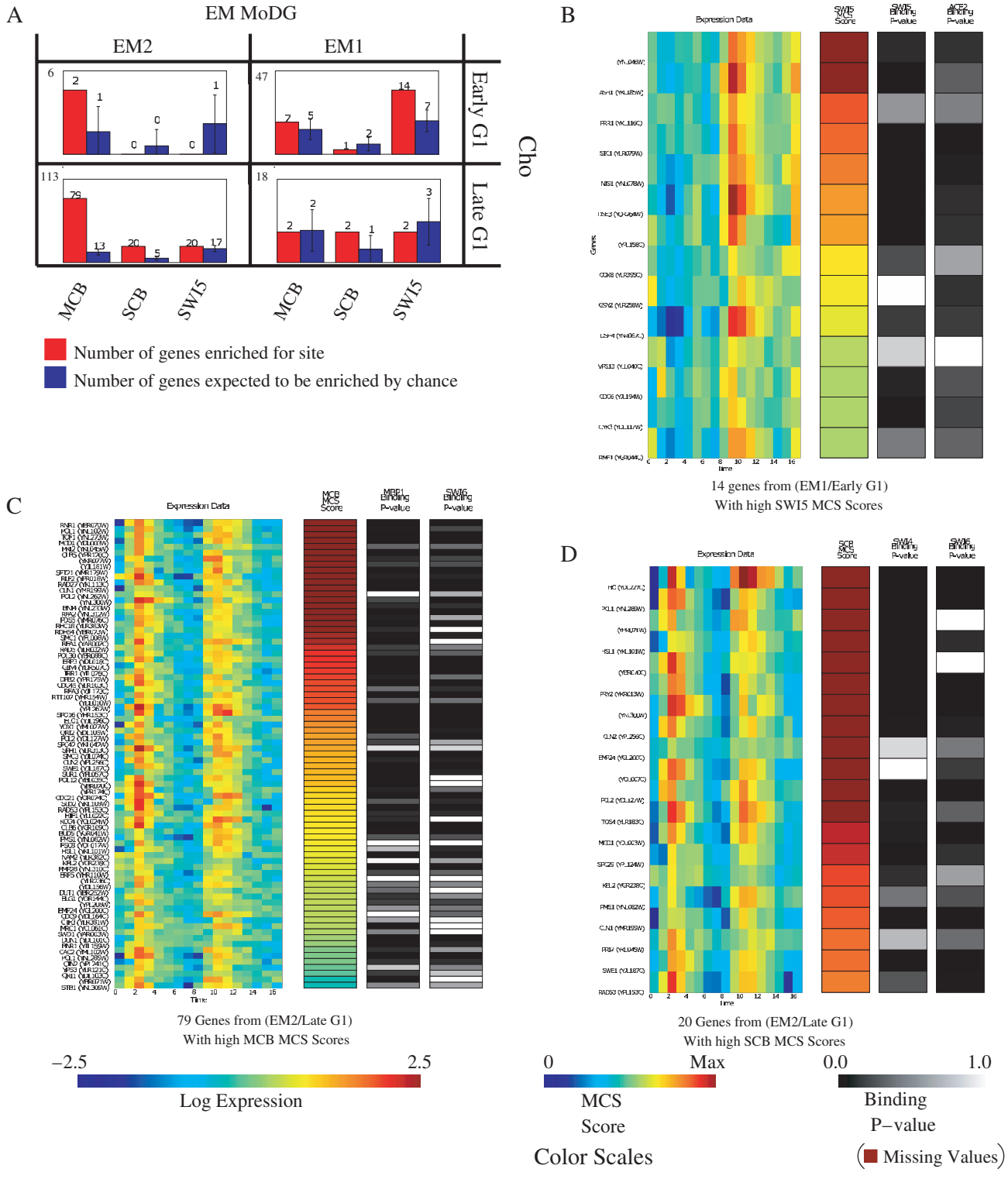

Figure 8. Integrating expression data, regulatory motif conservation and protein–DNA binding information. (**A**) Binding site enrichment in genes from the four confusion matrix cells of Figure 7 that dissect genes in the G_1 cell cycle phase. Shown in red are the observed number of genes with a MCS score above threshold for each motif. Shown in blue are the number of genes expected by chance, as computed by bootstrap simulations. The total number of genes each cell contains is in the upper left. (B–D) Heat-map displays showing expression data on the left, followed by MCS scores for a specified motif, followed by *in vivo* protein–DNA binding data for transcription factors implicated in binding to the specified consensus. Color scales for each panel are at the bottom of the figure. For the MCS scores, the color map ranges from 0 to the 99th percentile to minimize the influence of extreme outliers on interpretation. (**B**) Shown are 14 genes that fall within the EM1/Early G_1 intersection cell and have a conserved enrichment in the presence of the SWI5 consensus as measured by MCS scores (see Methods; Equations 4–9) (**C**) Shown are 79 genes that fall within EM2/Late G_1 intersection cell and have a high MCS score for MCB. (**D**) Shown are 20 genes that fall within EM2/Late G_1 intersection cell and have a high MCS score for SCB. In each heat-map genes are ordered by decreasing MCS score. Significant correlation can be seen between a high MCS score, protein–DNA binding and the expected expression pattern.

networks [(9); C. Hart, E. Mjolsness, B. Wold, manuscript in preparation]. By using CompClust in this way, we were easily able to capture all known regulatory modules governing yeast G_1 transcription and to relate these regulatory connections to specific expression clustering patterns.

ACKNOWLEDGEMENTS

Funding of this work and its open access publication was from the NCI, the NIH, NASA, the Department of Energy, and the LK Whittier Foundation. The authors thank Prof. Joe Hacia, Drs Jose Luis Riechmann and Brian Williams for helpful comments on the manuscript.

Conflict of interest statement. None declared.

REFERENCES

1. Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J. and Davis,R.W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.
2. Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.
3. Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J.R., Caligiuri,M.A. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
4. Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.
5. Ross,D.T., Scherf,U., Eisen,M.B., Perou,C.M., Rees,C., Spellman,P., Iyer,V., Jeffrey,S.S., de Rijn,M.V., Waltham,M. *et al.* (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genet.*, **24**, 227–235.
6. Ihmels,J., Friedlander,G., Bergmann,S., Sarig,O., Ziv,Y. and Barkai,N. (2002) Revealing modular organization in the yeast transcriptional network. *Nature Genet.*, **31**, 370–377.
7. Forbes,A. (1995) Classification-algorithm evaluation—5 performance-measures based on confusion matrices. *J. Clin. Monit.*, **11**, 189–206.
8. Gusfield,D. (2002) Partition-distance: a problem and class of perfect graphs arising in clustering. *Information Processing Letters*, **82**, 159–164.
9. Hart,C.E. (2005) Inferring genetic regulatory network structure: integrative analysis of genome-scale data. PhD thesis, Division of Biology, California Institute of Technology, Pasadena, CA, USA.
10. Rand,W.M. (1971) Objective criteria for evaluation of clustering methods. *J. Am. Stat. Assoc.*, **66**, 846–850.
11. Hubert,L. and Arabie,P. (1985) Comparing partitions. *Journal of Classification*, **2**, 193–218.
12. Levine,E. and Domany,E. (2001) Resampling method for unsupervised estimation of cluster validity. *Neural Comput.*, **13**, 2573–2593.
13. Ben-Hur,A., Elisseeff,A. and Guyon,I. (2002) A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.*, 6–17.
14. Peterson,W.W. (1954) The theory of signal dectectability. *IRE Transactions on Information Theory*, **4**, 171–212.
15. Swets,J.A. (1988) Measuring the accuracy of diagnostic systems. *Science*, **240**, 1285–1293.
16. Spellman,P.T., Sherlock,G., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.
17. Dempster,A., Laird,N. and Rubin,D. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B*, **39**, 1–38.
18. Sherlock,G. (2000) Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, **12**, 201–205.
19. Csank,C., Costanzo,M.C., Hirschman,J., Hodges,P., Kranz,J.E., Mangan,M., O'Neill,K., Robertson,L.S., Skrzypek,M.S., Brooks,J. and Garrels,J.I. (2002) Three yeast proteome databases: Ypd, pombepd, and calpd (mycopathpd). *Methods Enzymol.*, **350**, 347–373.
20. Raychaudhuri,S., Stuart,J.M. and Altman,R.B. (2000) Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac. Symp. Biocomput.*, 455–466.
21. Nasmyth,K. (1985) A repetitive DNA sequence that confers cell-cycle START (CDC28)-dependent transcription of the HO gene in yeast. *Cell*, **42**, 225–235.
22. Breeden,L. and Nasmyth,K. (1987) Cell cycle control of the yeast HO gene: *cis*- and *trans*-acting regulators. *Cell*, **48**, 389–397.
23. Koch,C., Moll,T., Neuberg,M., Ahorn,H. and Nasmyth,K. (1993) A role for the transcription factors Mbp1 and Swi4 in progression from G_1 to S phase. *Science*, **261**, 1551–1557.
24. Horak,C.E., Luscombe,N.M., Qian,J., Bertone,P., Piccirrillo,S., Gerstein,M. and Snyder,M. (2002) Complex transcriptional circuitry at the G_1/S transition in *Saccharomyces cerevisiae*. *Genes Dev.*, **16**, 3017–3033.
25. Iyer,V., Horak,C., Scafe,C., Botstein,D., Snyder,M. and Brown,P. (2001) Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, **409**, 533–538.
26. Breeden,L.L. (2003) Periodic transcription: a cycle within a cycle. *Curr. Biol.*, **13**, R31–R38.
27. Cliften,P., Sudarsanam,P., Desikan,A., Fulton,L., Fulton,B., Majors,J., Waterston,R., Cohen,B.A. and Johnston,M. (2003) Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science*, **301**, 71–76.
28. Kellis,M., Patterson,N., Endrizzi,M., Birren,B. and Lander,E.S. (2003) Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, **423**, 241–254.
29. Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.
30. Kovacech,B., Nasmyth,K. and Schuster,T. (1996) EGT2 gene transcription is induced predominantly by Swi5 in early G_1. *Mol. Cell. Biol.*, **16**, 3264–3274.
31. Doolin,M., Johnson,A., Johnston,L. and Butler,G. (2001) Overlapping and distinct roles of the duplicated yeast transcription factors Ace2p and Swi5p. *Mol. Microbiol.*, **40**, 422–432.