http://www.psi.toronto.edu

Factorgrams: A tool for visualizing multi-way associations in biological data

Vincent Cheung, Inmar Givoni, Delbert Dueck, Brendan J. Frey

May 15, 2006 **PSI TR 2006–44**

Abstract

Effective visualization of biological data is often critical for subsequent analysis. The popular clustergram/dendrogram visualization rearranges rows and columns of a data matrix so as to highlight clusters of similar responses, but assumes each row or column belongs to only one cluster and cannot associate each row or column with multiple clusters. Such multi-way associations occur frequently, *e.g.*, when a gene plays multiple biological roles. We describe the 'factorgram' visualization, which rearranges the data into an expanded view, associating each row (or column) with multiple clusters of rows (or columns) and elucidating potentially new biological relationships. Factorgrams for mouse gene expression and yeast synthetic-lethal gene-interaction datasets detect a larger number of statistically-significant clusters than clustergrams, plus a larger number of clusters enriched for gene ontology annotations. Experimentally-verified associations previously identified by manual rearrangement of rows and columns not grouped together by clustergrams, are readily identified by the factorgram.

Factorgrams: A tool for visualizing multi-way associations in biological data

Vincent Cheung[1,2], **Inmar Givoni**[1,2], **Delbert Dueck**[1,2], **Brendan J. Frey**[2,3,4]

1. These authors contributed equally
2. Probabilistic and Statistical Inference Group, Departments of Electrical and Computer Engineering and Computer Science, University of Toronto, 10 King's College Rd., Toronto, ON, Canada, M5S 3G4
3. Banting and Best Department of Medical Research, University of Toronto, 112 College St., Toronto, ON, Canada, M5G 1L6
4. To whom correspondence should be addressed: Brendan J. Frey, frey@psi.toronto.edu, 416-978-7001 (office), 416-978-4425 (fax).

Effective visualization of biological data is often critical for subsequent analysis. The popular clustergram/dendrogram visualization rearranges rows and columns of a data matrix so as to highlight clusters of similar responses [1], but assumes each row or column belongs to only one cluster and cannot associate each row or column with multiple clusters. Such multi-way associations occur frequently, *e.g.*, when a gene plays multiple biological roles. We describe the 'factorgram' visualization, which rearranges the data into an expanded view, associating each row (or column) with multiple clusters of rows (or columns) and elucidating potentially new biological relationships. Factorgrams for mouse gene expression and yeast synthetic-lethal gene-interaction datasets detect a larger number of statistically-significant clusters than clustergrams, plus a larger number of clusters enriched for gene ontology annotations. Experimentally-verified associations previously identified by manual rearrangement of rows and columns not grouped together by clustergrams, are readily identified by the factorgram.

High-throughput biological assays produce huge quantities of data, which can often be organized into a matrix where rows and columns correspond to separate control variables, and each element in the matrix is a binary- or real-valued measured response. For example, the rows may correspond to microarray probes for genes, exons or tiled DNA segments, the columns to time points or tissue samples, and each element to a real-valued transcript abundance estimated using a microarray. Or, the rows and columns may correspond to yeast genes and each element to a binary value indicating whether or not there is a synthetic lethal interaction in the yeast strain with both genes deleted. While customized techniques based on machine learning and statistical inference can be used to analyze this kind of data, even simple rearrangements of the data or straightforward transformations of the data can reveal biologically-significant patterns that are directly visualized in the rearranged dataset. Currently, the most common visualization tool for biological matrix data is the clustergram, in which the rows and the columns of the data matrix are re-ordered using a clustering method such as hierarchical agglomerative clustering (HAC) [11] or *k*-means clustering [12]. Similarities between nearby rows or columns may be indicated by dendrogram. The rearranged matrix of data is shown as an image where the color at each coordinate corresponds to the appropriate binary- or real-valued response [1].

A major deficiency of clustergrams is that rows (or columns) are grouped together based on similarity of response across their entire columns (or rows). In many biological assays, it is frequently the case that different biological processes will impact overlapping, but not disjoint, sets of variables. For example, consider a yeast synthetic gene interaction matrix in which rows correspond to gene deletion mutants of different non-essential yeast genes, columns also correspond to gene deletion mutants of non-essential yeast genes, and each element in the matrix measures the viability of a yeast strain with both genes knocked out. If genes A and B exhibit similar viability patterns across a subset of other genes, they can be grouped together, indicating that they may be part of the same functional pathway involving all genes in the subset. However, gene B may additionally play a different biological role as part of another pathway that does not involve gene A, but involves gene C. Genes B and C may exhibit similar viability patterns across a different subset of genes, namely those that are involved in the second pathway. In this case, there are two clusters

corresponding to the two pathways and they overlap in the sense that they both contain gene B. This creates a problem for clustering methods, which assume that clusters do not overlap. Standard clustering techniques and the clustergram visualization do not provide a general way to link gene B to both genes A and C, what we refer to as making 'multi-way associations'.

The failure of the clustergram to indicate multi-way associations is evident from the data of Tong *et al.* [13], shown in Fig. 1A by ordering the rows and columns according to the dendrograms produced by HAC (see Methods). Each row or column can only be associated with nearby rows or columns, so overlapping clusters of genes (*i.e.*, multi-way associations) are not properly revealed. In Fig. 1B we highlight two examples of data clusters that are significantly enriched for gene function annotations, but that were not properly identified by hierarchical clustering and were thus broken apart in the clustergram. In some cases, a row or column can be visually associated with two clusters if by coincidence it is on the boundary between the two clusters or if a computational method is used to move it to a position that is closer to another cluster. However, clustergrams cannot generally reveal all such double-associations, because only a small number of cases can be placed near cluster boundaries.

The 'factorgram' is a visualization that expands the input data matrix so as to enable explicit visualization of data in overlapping clusters. We refer to each such overlapping cluster as a 'factor', because techniques that explain each row or column of data as a composition of multiple components are called 'factorization' methods in the machine learning and statistics communities, and each component is called a 'factor' (c.f. [2]). The output from a variety of factorization techniques [2-10] can be further processed to produce the factorgram. The purpose of this paper is not to advocate a particular factorization technique, but to illustrate how the factorgram visualization can be useful to biologists for further analysis and detection of a larger number of statistically and biologically significant associations than cannot be detected using clustergrams. Software for producing factorgrams is available at http://www.psi.toronto.edu/factorgram.

To give the reader a sense for how factorgrams work, in Fig. 2 we show a cartoon illustrating one way to construct a factorgram. The clustergram of a synthetic binary data matrix is shown in the upper-left corner, and includes blocks of data that have been broken apart because specific rows and columns (shown with arrows) belong to multiple groups. To produce the factorgram, the rows and columns are rearranged so that the dominant factor appears as a contiguous block of data. This factor is then removed from the matrix and placed in the factorgram. The rows and columns of the resulting matrix are again rearranged so that the next dominant factor appears as a contiguous block of data. This factor is removed and placed in the factorgram. This procedure is repeated until no more significant factors remain in the data matrix. The factorgram is a figure showing the raw data associated with all extracted factors and may additionally include appropriate row and column labels. The factors may be shown along the diagonal of a matrix to reflect similarities to the clustergram, or the factors can be arranged more compactly. Since the same row (or column) can appear in multiple factors, the factorgram provides a way to visualize clusters with overlapping variables.

Techniques for computing factorgrams should properly address how to efficiently search over possible solutions and assess the statistical significance of the detected factors. In contrast to clustergrams, where each row or column is associated with one cluster, in factorgrams, each row or column can be associated with multiple factors through different subsets of responses. This capability comes at the cost of an exponentially larger space of possible solutions, since if there are k factors and each row or column can belong to n factors at once, there are k^n possible ways of assigning the row or column to the factors. Approaches for efficiently searching over these assignments include computing linear approximations and using more powerful probabilistic inference techniques. A binary assignment variable with value 0 or 1 can be used to indicate whether or not a row or column belongs to a factor. If these assignment variables are relaxed to be real-valued, data elements are described by linear combinations of continuous variables. Continuous solutions to the factorization problem can be computed using an eigen-vector method such as principal components analysis [3] or an iterative technique such as factor analysis [2], independent component analysis [4] and non-negative matrix factorization [5]. Once the

real-valued ‘assignment’ variables have been computed, they can be thresholded to identify components in the factorgram. One problem with these approaches is that a good binary solution often cannot be obtained by thresholding the best continuous solution. An alternative approach is to retain the binary representation of the assignment variables but constrain the solution space. Bi-clustering [6] imposes the constraint that any two clusters containing the same column (or row) must be defined by exactly the same set of columns (or rows); this constraint makes extraction of factors easier, but is only appropriate when different factors are defined by the same subsets of rows or columns.

Recently, methods have been proposed that retain the binary representation of the assignment variables and avoid simplifying assumptions about the factors by using techniques developed in the probabilistic and statistical inference communities to search for the most probable setting of the assignment variables. The plaid method [7] works by extracting one factor at a time using the linear approximation described above, but then thresholds the assignment variables before proceeding to extract the next factor. Probabilistic sparse matrix factorization [8,9] and matrix tile analysis [10] take a direct approach and retain the binary representation, but find solutions by accounting for uncertainties in the assignments and iteratively revisiting and refining factors until convergence. The labelled latent Dirichlet allocation process analysis method [15] takes as input the data matrix along with gene function annotations and performs Bayesian inference in a hierarchical probabilistic graphical model to extract the factors.

The statistical significance of the factors in a factorgram can be assessed in a variety of ways, but here we describe a general procedure that can be applied to any technique that computes a factorgram. Denote the data matrix by X and the particular method used to compute the factorgram by μ. We assume the method takes as input the data matrix and a parameter θ that indirectly has an impact on the false detection rate. For example, θ could be a real-valued threshold on a cost function that penalizes small factors (which are less likely to be significant) but rewards factors containing highly similar data elements. For method μ applied to data X using parameter θ, denote the number of extracted factors by $N(\mathrm{X},\mu,\theta)$. The null hypothesis is that a computed factor arose from random data with the

same distribution as the source that produced X. Assuming the data matrix is large enough, this distribution can be approximated by generating permuted data matrices (*i.e.*, randomly rearranging elements). Denote a data matrix obtained in this way by X^R – since the data elements were randomly rearranged, X^R is a random variable. The expected number of falsely detected factors can be estimated by $E[N(X^R,\mu,\theta)]$, where E[ ] denotes an expectation w.r.t. the set of random matrices, X^R. In this procedure, $N(X^R,\mu,\theta)$ is random because X^R is random, but also possibly because the method μ may produce different solutions each time it is applied due to varying initial conditions.

We demonstrate the factorgram visualization using two publicly available datasets, the yeast synthetic genetic array (SGA) dataset from Tong *et al.* [13] and the mouse gene expression dataset from Zhang *et al.* [16]. The former is binary valued while the latter is real-valued, and each dataset was factorized using a different method. However, the final results are both readily visualized by the factorgram.

The SGA data is a binary-valued matrix of 135 by 1023 elements, where both rows (query genes) and columns (array genes) correspond to yeast genes. Each element represents the synthetic lethality of a double mutant whose corresponding row and column genes have been knocked out. We applied HAC to both the rows and the columns of the data matrix (see Methods) and in Fig. 1A we show the clustergram. The factorgram generated from this data matrix (see Methods) contained 17 factors, 5.4 of which are expected to be false detections based on the method described above, using ten random permutation tests. The factorgram for the SGA data is shown in Fig. 3A, where each box corresponds to one of the factors. While in the data matrix, only 3.35% of the gene pairs exhibit synthetic lethality, in the identified set of factors 88.87% of the gene pairs exhibit synthetic lethality. The identified factors account for 49.57% of the total number of synthetic lethal interactions; 22.75% of the observed synthetic lethal interactions correspond to array genes have three or less synthetic lethal interactions with query genes and thus provide only weak evidence. In Fig. 3B, we show the clustergram obtained using HAC (see Methods). We tried several clustering techniques and chose the one that produced the largest number of biologically significant clusters (see below). In this figure,

each element assigned to a factor is coloured so as to indicate the factor in Fig. 3A to which it belongs. The clustergram breaks apart almost every identified factor, thus failing to associate all genes within each factor.

An example of a well known biological pathway that is not readily identified in the clustergram in Fig. 3A is the chitin synthase III pathway (*SKT5, CHS3, CHS5, CHS7*), which is grouped together in the factorgram (dark green factor in Fig. 3). In some cases, the clustergram correctly groups query genes, but fails to group array genes. For example, query genes in the prefoldin complex (*GIM3, GIM4, GIM5, PAC10, YKE2*) were correctly grouped in the clustergram, but corresponding array genes were not properly identified (red factor in Fig. 3). The clustergram produced by Tong *et al.* used a different metric and successfully grouped genes in the synthase III pathway, but failed to group together array associated with the prefoldin complex. Instead, this complex was identified by laborious manual rearrangement of the partial clusters. The large red factor shown in Fig. 3 successfully brings together similarities of array gene profiles and query gene profiles, thus capturing both the row-based similarities and the column based similarities of the gene profiles. Furthermore, the dark purple factor shows how the query genes of the prefoldin complex are also similar based on their symmetric interaction as array genes, a relationship that again was neither captured in our clustergram nor reported by Tong *et al.*

We next asked how many of the extracted factors are of biological significance in terms of functional enrichment. We analysed whether the genes in each cluster were significantly enriched for gene ontology (GO) annotations of biological process, molecular function, and cellular component. We found 9 of the clusters to be enriched ($p < 0.01$) for at least one annotation, with a total of 18 significant enrichments across all three categories (see Methods). In comparison, when we used HAC to find the same number of clusters, we were able to find only 4 clusters to be enriched for at least one annotation, with a total of only 7 significant enrichments across all three categories. We tried a variety of different clustering techniques, but these resulted in lower numbers of enriched clusters (data not shown).

The second dataset we analyzed and applied the factorgram visualization to is the mouse mRNA expression dataset from Zhang *et al.* [16]. This dataset contains profiles for over 22,709 known and predicted genes across 55 mouse tissues, organs, and cell types. The factorgram generated from this dataset (see Methods) contained 37 factors and based on the random permutation test we found that 0.5 factors are expected to be false detections. The clustergram and factorgram are shown in Fig. 4. The factorgram correctly identified many associations that are identified in the clustergram. For example, the cluster containing nervous system tissues (inside the dark green bounding box in Fig. 4A) is contained within the factor shown in Fig. 4D. Fig. 4C shows an example where the discovered factor has been severely broken apart in the clustergram. This factor includes genes that are expressed in both mature and embryonic nervous system tissues, and includes genes not present in the previously described factor. Both of these factors have statistically significant ($p < 0.05$) enrichment in gene annotations (see Methods).

Many factors reveal profile similarities across tissues which are not obviously similar, and may seem surprising at first. However, this is one of the advantages of factorization methods and of the factorgram, as they have the potential to enable the researcher to scrutinize more interesting relationships. To answer the question of whether the additional relationships detected in the factorgram are of biological significance, we analyzed the factors for enrichment in GO biological process (GO-BP) annotations (see Methods). We studied how many biologically significant groups were revealed in the factorgram compared to other techniques for a fixed number of detected groups. In this case, parameter θ controls the number of factors discovered in the data. We varied θ from 20 to 60 and compared the genes in each factor with GO-BP annotations. Table 1 reports the number of statistically significant factors ($p < 0.05$) for each θ-value; we also report corresponding results for HAC. In general, the factogram reveals a larger number of significantly enriched clusters.

Table 1: Comparison of number of clusters or factors enriched for GO-BP annotations for different numbers of extracted clusters/factors.

Number of clusters/factors, θ	Number of enriched factors in factorgram	Number of enriched clusters in clustergram
20	16	13
30	21	15
40	26	17
50	28	19
60	32	20

Just as Eisen et al. [1] demonstrated that the clustergram provides a visual tool enabling biologists to gain leads to interesting associations, our goal in writing this paper was to demonstrate the advantages of the factorgram visualization in revealing multi-way associations and enabling biologists to detect multiple associations. Unlike clustergrams, where the number of salient associations is limited by the two-dimensional arrangement of the data, factorgrams extract many two-dimensional arrangements so as to identify multiple associations. We are not advocating a specific computational technique for factorizing the data matrix, but are instead introducing and advocating the factorgram as a way to visualize the outputs from a variety of computational techniques [2-10], each of which is appropriate in specific situations. Factorgrams can be used to visualize the output of these techniques, regardless of whether each element in the generated factors belongs to only one factor or to multiple factors. In our experiments on synthetic lethal yeast gene interaction data and mouse gene expression data, we found that factorgrams reveal a larger number of statistically significant and biologically significant clusters compared to the number revealed in clustergrams by HAC.

As a general tool for analyzing matrices of data, the factorgram can potentially have broad application in detecting associations in biological data. The factorgram has been recently applied to chemical-genetic interaction data in yeast to identify associations between chemical compounds and genes [17]. It has also been used to identify relationships between hundreds of genes involved in transcription [18]. New large-scale microarray datasets for the study of mammalian alternative splicing have also recently been analyzed

using factorgrams [19]. Software for automatically producing factorgrams is available at http://www.psi.toronto.edu/factorgram. Due to its simplicity, ease of computation, and potential for revealing novel biological insights, we believe the factorgram will prove to be a useful tool for visualizing multi-way associations in high-throughput biological data.

Methods

1. Yeast gene deletion data analysis.

The clustergram was produced using the MATLAB implementation of hierarchical agglomerative clustering (HAC). We used Hamming distance, Euclidean distance, correlation and city-block pseudo-distance functions with average and single linkage and report results for Hamming distance with average linkage, which obtained the largest number of clusters enriched for GO annotations.

The factorgram was produced using 'matrix tile analysis' based on an iterated conditional modes technique [10]. Matrix tile analysis takes as input a matrix, where each element is the log-ratio of probabilities that the element belongs in a factor and does not belong in a factor. If a synthetic lethal interaction was observed between two pairs of genes in the dataset of Tong et al. [13], we set the log-ratio to be $\log((1-\varepsilon)/\varepsilon)$ and otherwise we set it to be $\log(\varepsilon/(1-\varepsilon))$, where ε is the noise probability set to 0.0335, based on the average number of observed synthetic lethal interactions. The parameter θ, which indirectly determines the false detection rate by weighting the benefit of explaining the data to the cost of introducing additional factors, was set to 0.15.

2. Mouse gene expression data analysis.

The clustergram was produced using the MATLAB implementation of HAC with Pearson correlation and average linkage, which was selected for visualizing results by Zhang *et al.* [16].

For the factorgram, we tried a different factorization technique from above called 'probabilistic sparse matrix factorization' (PSMF) [8,9]. Each expression measurement of a

gene in each tissue in the input matrix is represented by the arcsinh (approximately the logarithm) of the ratio of normalized intensity of the gene in the given tissue to the gene's median normalized intensity across all 55 tissues. Observing that the majority of genes expressed in any tissue were expressed in less than half of the tissues, Zhang *et al.* set ratios less than one equal to one, reasoning that these ratios represent noise rather than down-regulation thus the data is entirely non-negative.

PSMF discovers a large predetermined number of factors (set to 50 here) and then prunes them until every factor has a standard deviation less than an input threshold, θ, which was set to 9.5. To determine the expected number of false detections, we reran the analysis 135 times on randomly permuted data and computed the expected number of false detections, which was 0.5. Based on the permutation tests, we also estimate the distribution of factor element intensities for non significant factors (as those expected to be found on permuted data) and discard of any factor elements found in the unpermuted data which have an intensity below that of the 97.5^{th} percentile of permutated factor elements. For visualization purposes, Fig. 4B shows only those genes with individual reconstruction error of less than θ.

3. Gene ontology (GO) enrichment analysis

GO annotation labels for the yeast genes were retrieved from *Saccharomyces* Genome Database at ftp://genome-ftp.stanford.edu/pub/yeast/data_download/literature_curation/. The functional category labels for the genes with known biological function in the Zhang *et al.* database were derived from Gene Ontology Biological Process (GO-BP) category labels assigned to genes by the European Bioinformatics Institute and Mouse Genome Informatics. Bonferroni-corrected p-values for each factor/cluster were computed using the hyper-geometric distribution, testing the probability of observing by chance the overlap of a subset of genes in each factor or cluster with a subset of genes sharing a specific GO annotation. We assign p-values for each factor/cluster by comparing it with all GO annotations and choosing the most significant p-values.

Acknowledgements

We thank C Boone, M Escobar, J Greenblatt, N Krogan, QD Morris and S Roweis for helpful conversations. This work was supported by NSERC and CIAR.

Correspondence and requests for materials should be addressed to Brendan Frey at frey@psi.toronto.edu.

Figure Legends

Figure 1. (A) A clustergram of the 135 x 1023 yeast gene interaction data from Tong *et al.* [13], where rows correspond to 'query' genes, columns correspond to 'array' genes and each data element is white if a double knockout of the two genes is lethal. For visual clarity, only columns with a non-negligible number of interactions are shown. (B) Synthetic lethal interactions in two groups of data that we detected and are significantly enriched for gene function annotations are coloured; the remaining interactions are shown in grey. The clustergram cannot associate genes in multiple ways, so these groups are broken apart. For example, the array genes (columns) corresponding to the data shown in red all have synthetic lethal interactions with query genes (rows) GIM3, GIM4, GIM5, PAC10 and YKE2 (the prefoldin complex), but this association is broken apart because some of the array genes also have synthetic lethal interactions with query genes BIM1, CTF4, KAR3 and CIN8, while others do not.

Figure 2. A cartoon illustration of one way a factorgram can be constructed. The clustergram of a binary data matrix is shown in the upper-left corner. The small arrows indicate rows and columns having multi-way associations not evident in the clustergram. The factorgram is created by recursively reordering rows and columns to identify a block of data and then extracting the block of data. Each block is called a 'factor' to differentiate it from a 'cluster', because more than one factor can contain the same row or column, *e.g.*, the red factor and the green factor both include rows 15-19. The factorgram is a visualization of the raw data in the form of factors that may be placed in an expanded

matrix (where multiple rows or columns can have the same label) or shown as a collection of sub-matrices with labelled rows and columns.

Figure 3. (A) A factorgram of the yeast genetic interaction data from Fig. 1, where each factor has been colour-coded and non-synthetic lethal interactions are shown in black. (B) The continuation of Fig. 1B, where each observed synthetic lethal interaction has been coloured according to the colour code from A. Almost all factors are broken into multiple parts by the clustergram.

Figure 4. (A) A clustergram of the 22,709 gene x 55 tissue mouse microarray dataset from Zhang *et al.* [16]. (B) Factorgram visualization with the factors placed in a diagonal pattern. Two of the factors are enlarged in (C) and (D), and the general areas in the data set from which they came are indicated. Each factor is composed of a subset of rows and columns of the data, and only in certain cases do these subsets correspond to rows and columns that are adjacent in the clustergram. When this is not the case, the clustergram has broken apart the factor, as in (C).

References

1. MV Eisen, PT Spellman, PO Brown, D Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc National Acad Sci* **95,** 14863-14868 (1999).
2. D Rubin, D Thayer. EM algorithms for ML factor analysis. Psychometrika **47:1**, 69-76 (1982).
3. IT Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York NY (1986).
4. AJ Bell, TJ Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neur Comp* **7**, 1129-1159 (1995).
5. D Lee, S Seung. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788-791 (1999).
6. Y Cheng, GM Church. Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol* **8**, 93-103 (2000).
7. L Lazzeroni, AB Owen. Plaid models for gene expression data. *Stat Sin* **12**, 61-86 (2002).
8. D Dueck, J Huang, QD Morris, BJ Frey. Iterative analysis of microarray data. *Proc 42nd Allerton Conf Communication, Control and Computing*, Champaign-Urbana, IL (2004).

9. D Dueck, QD Morris, BJ Frey. Multi-way clustering of microarray data using probabilistic sparse matrix factorization. *Proc Int Conf Intell Syst Mol Biol 13, Bioinformatics* **21** (Suppl 1), i144-i151 (2005).
10. I Givoni *et al.* Matrix tile analysis, in press (2006).
11. RR Sokal, CD Michener. A statistical method for evaluating systematic relationships. *Univ Kans Sci Bull* **38**, 1409-1438 (1958).
12. SP Lloyd. Least squares quantization in PCM. *IEEE Trans Info Theory* **47**, 129 (2001).
13. AH Tong *et al.* Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* **294**, 2364-2368 (2001).
14. O Alter *et al.* Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci USA* **97**, 10101-10106 (2000).
15. P Flaherty, G Giaever, J Kumm, MI Jordan *et al.* A latent variable model for chemogenomic profiling. *Bioinf* **21**, 3286-3293 (2005).
16. W Zhang *et al.* The functional landscape of mouse gene expression. *J Biol* **3**, 21.1-21.22 (2004).
17. AB Parsons *et al.* Exploring the mode-of-action of bioactive compounds by chemical-genetic profiling in yeast. *Cell*, in press (2006).
18. N Krogan *et al.* A global view of transcriptional pathways in yeast. Under review (2006).
19. BJ Blencowe *et al.* Coordinated alternative splicing in functionally-associated mammalian genes. Under review (2006).

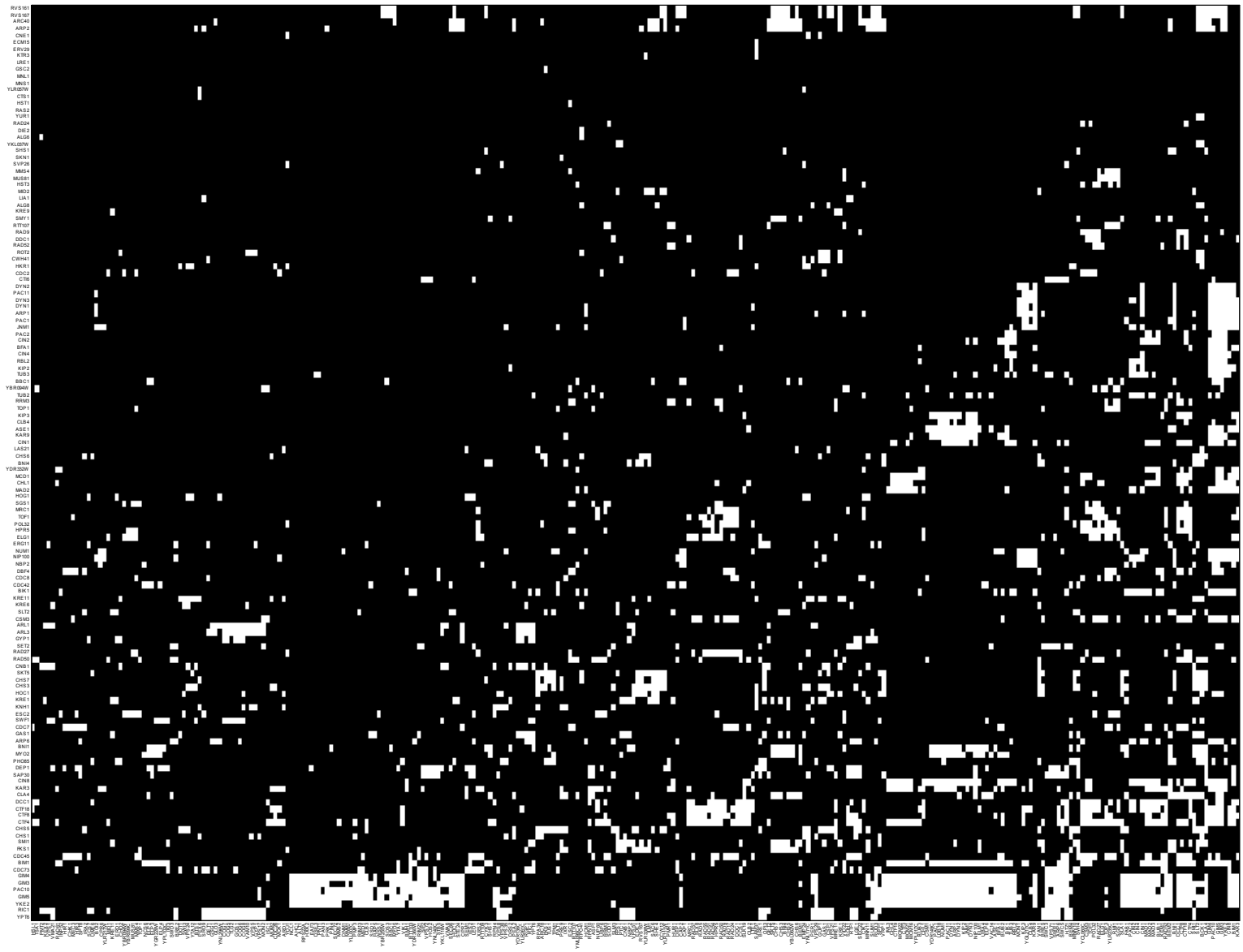

Figure 1A, Cheung et al

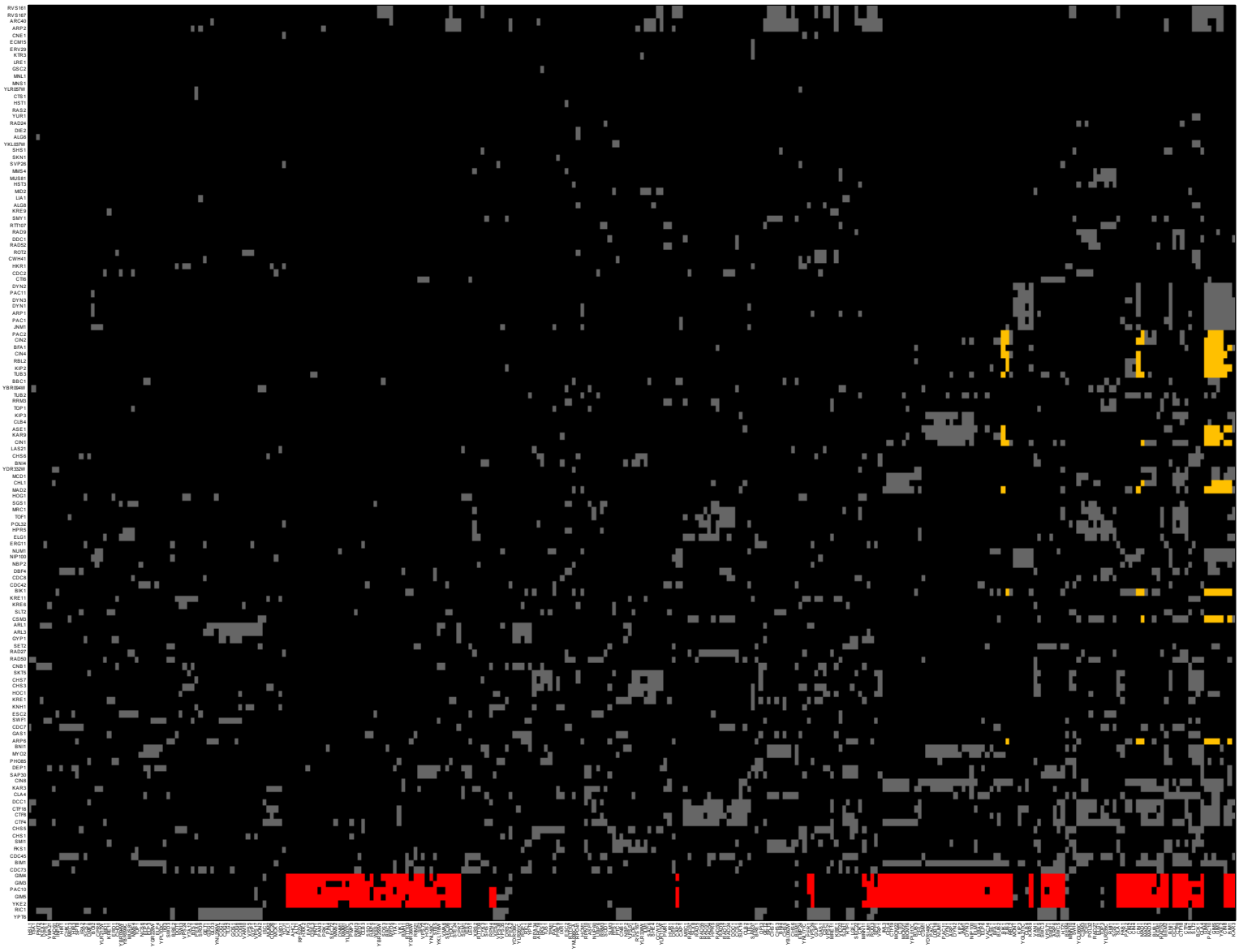

Figure 1B, Cheung et al

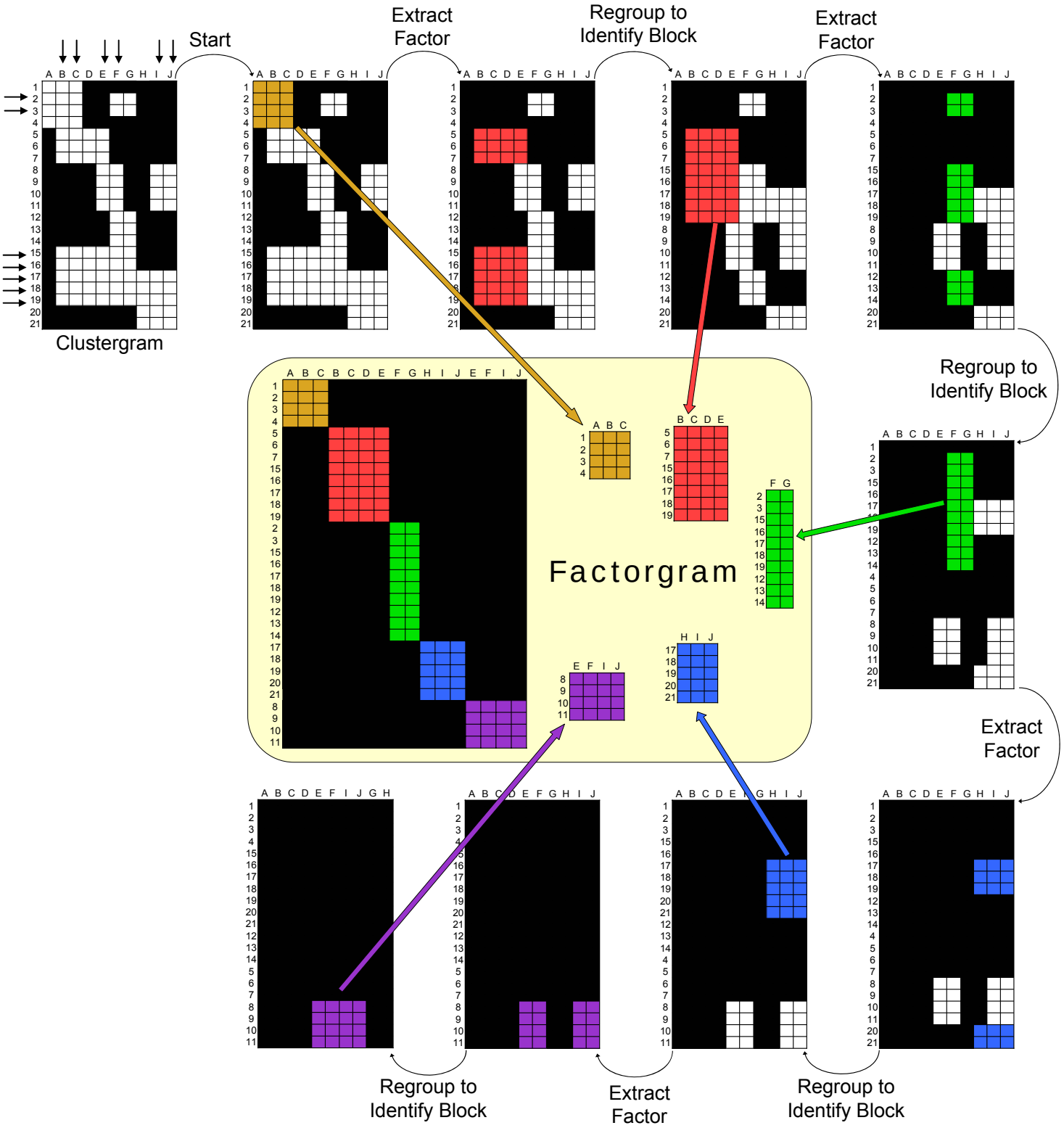

Figure 2, Cheung et al

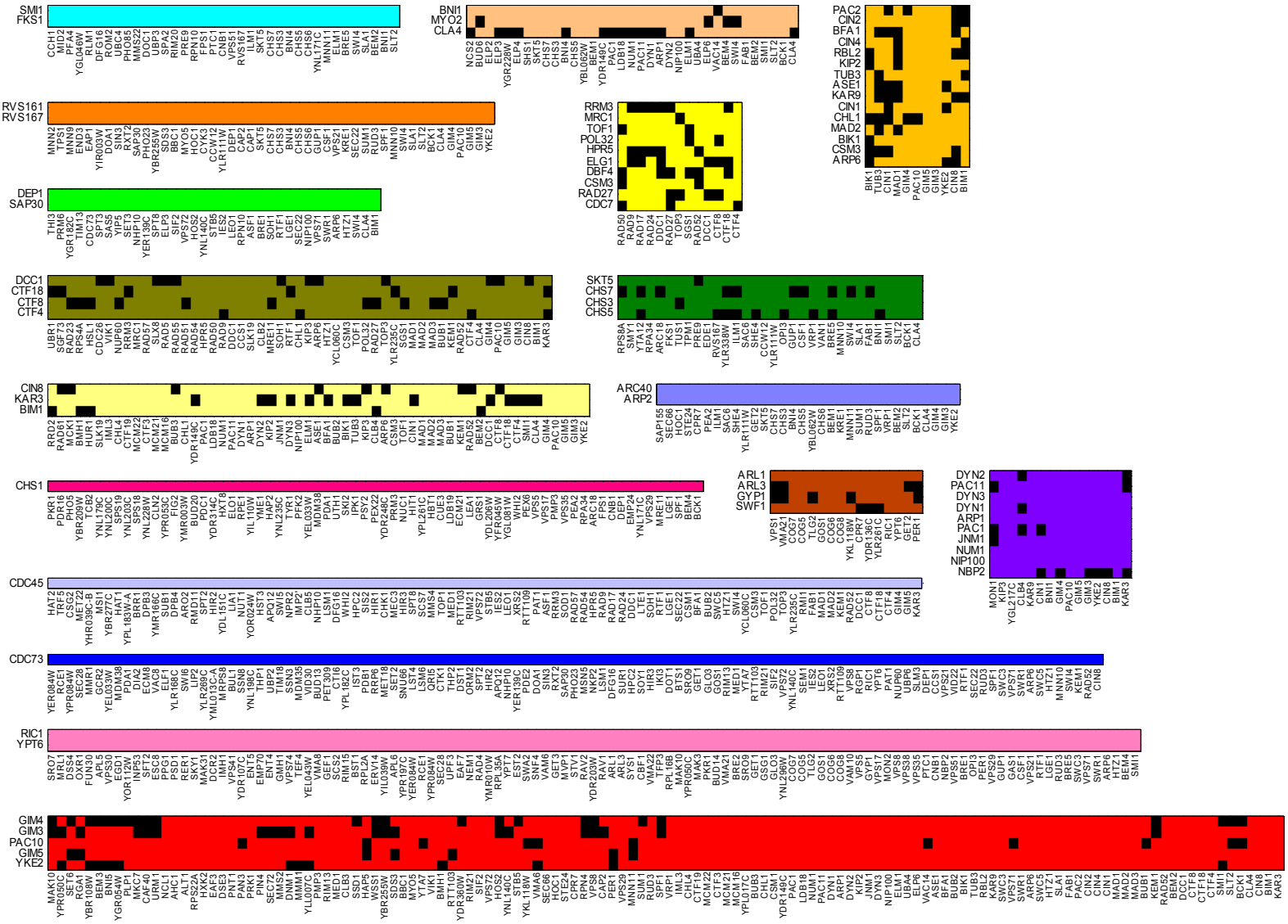

Figure 3A, Cheung et al

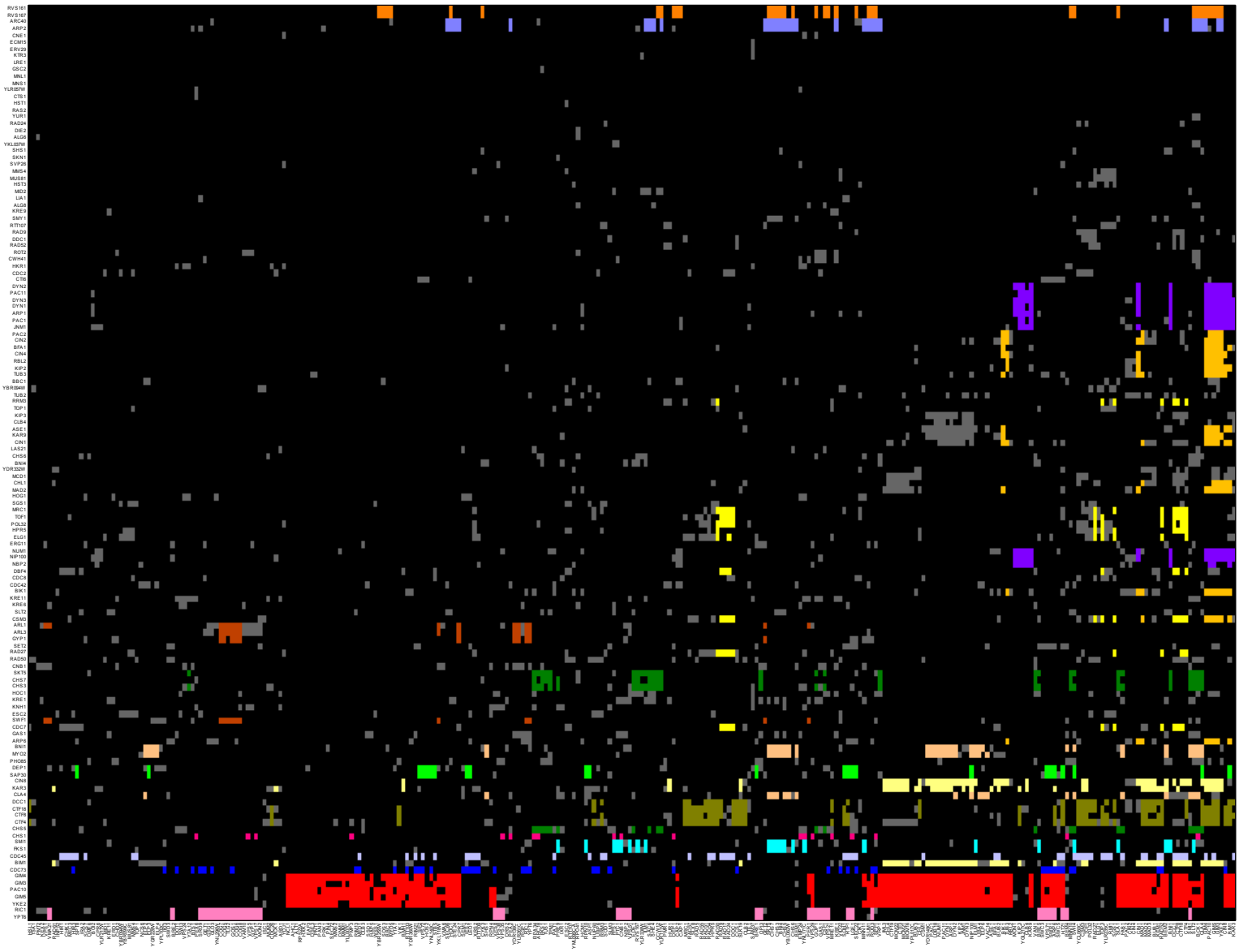

Figure 3B, Cheung et al

Figure 4, Cheung et al