Supplement Materials

APPENDIX

Proof of Proposition 1:

(a) Suppose $k_1 > k_2$. Let $C^{\#}(k_1,\lambda) = \{C_1^*(k_2,\lambda),\cdots,C_{k_2}^*(k_2,\lambda),\phi,\cdots,\phi,S^*(k_2,\lambda)\}$. By definition of equation (2), $W(C^*(k_2,\lambda);k_2,\lambda) = W(C^{\#}(k_1,\lambda);k_1,\lambda)$. Since $C^*(k_1,\lambda)$ is the minimizer of $W(C;k_1,\lambda)$, we have $W(C^{\#}(k_1,\lambda);k_1,\lambda) \geq W(C^*(k_1,\lambda);k_1,\lambda)$. Thus $W(C^*(k_2,\lambda);k_2,\lambda) \geq W(C^*(k_1,\lambda);k_1,\lambda)$.

(b) Suppose $\lambda_1 > \lambda_2$ and denote

$$W(t;k) = \min_{C:\sum_{i=1}^{k}|C_i|=t, C_i\subset X} \sum_{j=1}^{k}\sum_{x_i\in C_j} w(x_i)\cdot d(x_i,C_j).$$

We note that $\min_C W(C;k,\lambda) = \min_t\{W(n-t;k)+\lambda t\}$ and thus $W(C^*(k,\lambda);k,\lambda) = W(n-|S^*(k,\lambda)|;k)+\lambda|S^*(k,\lambda)|$. Since $C^*(k,\lambda_1)$ is the minimizer of $W(C;k,\lambda_1)$, we have

$$\begin{aligned} &W(n-|S^*(k,\lambda_1)|;k)+\lambda_1|S^*(k,\lambda_1)| \\ &\leq W(n-|S^*(k,\lambda_2)|;k)+\lambda_1|S^*(k,\lambda_2)|. \end{aligned} \quad (9)$$

Now suppose $|S^*(k,\lambda_2)| < |S^*(k,\lambda_1)|$. We have

$$\begin{aligned} &W(n-|S^*(k,\lambda_2)|;k)+\lambda_2|S^*(k,\lambda_2)| \\ &> W(n-|S^*(k,\lambda_2)|;k)+\lambda_2|S^*(k,\lambda_2)| - (\lambda_1-\lambda_2)(|S^*(k,\lambda_1)|-|S^*(k,\lambda_2)|) \\ &= W(n-|S^*(k,\lambda_2)|;k)-\lambda_1(|S^*(k,\lambda_1)|-|S^*(k,\lambda_2)|)+\lambda_2|S^*(k,\lambda_1)| \\ &\geq W(n-|S^*(k,\lambda_1)|;k)+\lambda_2|S^*(k,\lambda_1)| \quad \text{(from equation (9))}. \end{aligned}$$

This, however, contradicts with the assumption that $C^*(k,\lambda_2)$ is the minimizer of $W(C;k,\lambda_2)$. Thus $|S^*(k,\lambda_2)| \geq |S^*(k,\lambda_1)|$.

(c) $W(C^*(k,\lambda_2);k,\lambda_2)$

$$\begin{aligned} &= W(n-|S^*(k,\lambda_2)|;k)+\lambda_2|S^*(k,\lambda_2)| \\ &\leq W(n-|S^*(k,\lambda_1)|;k)+\lambda_2|S^*(k,\lambda_1)| \\ &< W(n-|S^*(k,\lambda_1)|;k)+\lambda_1|S^*(k,\lambda_1)| \\ &= W(C^*(k,\lambda_1);k,\lambda_1). \end{aligned}$$

CLUSTERING METHODS

Hierarchical clustering Hierarchical clustering was the first method used to cluster genes and samples in microarray data. It starts by considering the *n* data points as *n* nodes. Instead of partitioning into a number of clusters, a pair of nodes with the shortest distance between them are agglomerated to form a new node (agglomerative method) or the *n* nodes are successively separated into finer groups (divisive method) at each iterative stage. Thus a hierarchical tree is constructed after *n*-1 steps. In this paper we only consider agglomerative hierarchical clustering. To define the distance between two nodes, different linkages including single linkage (shortest pair-wise distance), complete linkage (largest distance), or average linkage (average distance) may be chosen in the method. In this paper, complete linkage is used which was found superior in gene clustering of microarray data (Thalamuthu et al., 2006).

SOM Self-organizing-maps (SOM)(Kohonen, 1990; Tamayo, et al., 1999) has been applied in many microarray analyses. It first maps *K* nodes in a low-dimensional (usually two-dimensional) grid space from the *d*-dimensional space in which the data set is situated and then the nodes are adjusted iteratively. Each time, a point from the data is randomly chosen. The movement of the nodes in *d*-dimensional space depends on their distance to the chosen point and the two-dimensional geometry of the nodes. The magnitude of movement decreases as iterations goes on. Usually the process continues more than 20,000 iterations for the nodes to converge and serve as cluster centers to form clustering. Essentially SOM can be viewed as a *K*-means criterion restricted on the two-dimensional grid geometry. Thus clusters generated from nodes close to each other in the two-dimensional grid geometry will have similar expression patterns. We not only can visualize expression patterns within each cluster but also can observe relations and connections between clusters on the two-dimensional node space.

Model-based clustering **(MCLUST)** In this approach, clustered data are fitted by a finite Gaussian mixture model (Fraley and Raftery, 2002). Each cluster is represented by a Gaussian probability distribution component and the data is viewed as a realization of a mixture distribution of the components. Let $\theta_j = (\mu_j, \Sigma_j)$ be the parameter associated with the probability distribution $f_j(x|\theta_j)$, μ_j the center and Σ_j the covariance structure of cluster *j*. Denote by π_j the probability that an observation belongs to the *j*-th cluster

($\pi_j \geq 0, \sum_{j=1}^{K} \pi_j = 1$), the classification likelihood of the n independent multivariate observations, $x_1, \cdots x_n$, is given as:

$$L = \log\left\{\prod_{i=1}^{n}\sum_{j=1}^{K}\pi_j f_j(x_i \mid \theta_j)\right\}$$

where $x = (x_1, \cdots x_n)$ is the dataset, $\theta = (\theta_1, \cdots, \theta_K)$ is the parameter vector and *K* is the number of clusters. The parameters then can be estimated by maximizing the classification likelihood through the EM algorithm. To select the complexity of Σ_j and the number of clusters *K*, the Bayesian Information Criterion (BIC) is often used. However, the BIC is an approximate measure for model selection and the optimization problem through the EM algorithm often falls into a local minimum, which make this method unstable in complex data. In general model-based clustering provides a sound framework for statistical inference and interpretation but may encounter difficulty of optimization and model selection in data with complex structure.

It is worth noting that model-based clustering can be modified to handle noise points present in the data. The classification likelihood to be maximized becomes:

$$L = \log\left\{\prod_{i=1}^{n}\left[\sum_{j=1}^{K}\pi_j f_j(x_i \mid \theta_j) + \pi_{K+1}\cdot u(x_i)\right]\right\}$$

where the noise points are generated from homogeneous Poisson process (i.e. $u(x_i)$ is the density of a uniform distribution). **The R package, mclust, was used for implementation. In the evaluation of cell cycle data in Figure 5, the best model with scattered objects among "'EII", "VII", "EEI", "VEI", "EVI" and "VVI" are automatically selected by BIC for comparison.**

<u>Gaussian infinite mixture model (GIMM)</u> In contrast to finite mixture model (FMM) above, this approach does not fix the number of components (K) a priori and allows it to go to infinity (Medvedovic and Sivaganesan, 2002; Medvedovic et al., 2004). The mean and covariance structure of each cluster as well as K are generated through prior distributions with non-informative hyperparameters. Gibbs sampler is performed to simulate the posterior distribution of the clustering result. The pair-wise probabilities for two genes to share the same cluster pattern (p_{ij}) are first calculated in the "burn-in" period of Gibbs sampler and then are input as the similarity measure to construct a hierarchical tree (average linkage). The hierarchical tree can be cut to obtain a clustering result with desired K. Compared to traditional hierarchical clustering, this method generates tree with more singleton or small nodes that are considered as "sporadic" or "scattered" genes. To generate unbiased comparison with other methods in Figure 4., we cut the hierarchical tree generated by GIMM and only select the clusters with more than 20 genes. All the nodes with less than 20 genes are considered as "sporadic" genes. When the tree is cut at increasingly deeper position, the desired K=5,…20 clustering results are obtained and used for comparison in Figure 4. We ran 10,000 Gibbs sampler iterations in the cell cycle example and estimated p_{ij} with the last 5,000 "burnt-in" clustering iterations.

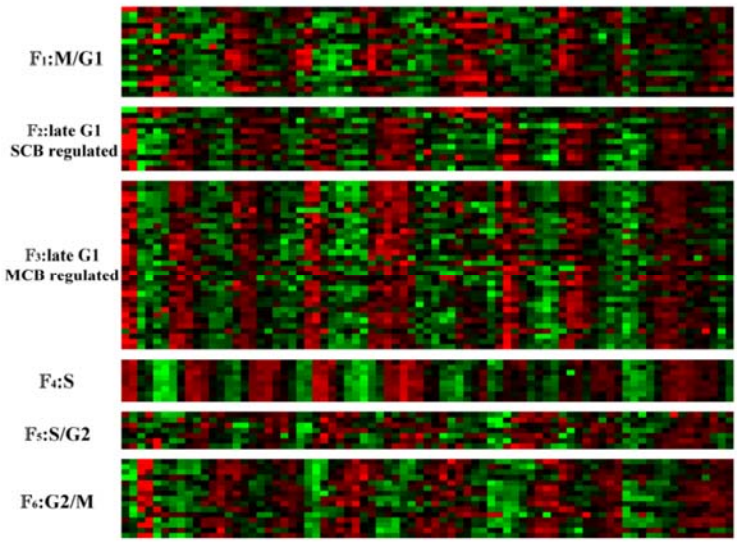
<u>Tight clustering</u> Tseng and Wong (2005) proposed a resampling-based method, called "tight clustering", for clustering complex data sets possibly containing a large number of scattered points. The idea is to perform clustering in repeated subsamples and identify candidate tight clusters that are almost always clustered together. The number of clusters *K* used in subsampling judgment is increasingly varied until a candidate tight cluster is identified in the consecutive *K*. This candidate tight cluster is robust to the selection of *K* and is identified as a tight cluster. The tight cluster is then removed from the data and the same procedure is repeated to search for the next tight cluster. In the original paper, *K*-means was used in subsampling clustering while conceptually any clustering method can be used.

The advantage of tight clustering is that it directly searches for tight clusters, making it possible to leave remaining points as noises. It can be viewed as a higher order re-evaluation mechanism built upon any existing clustering method. Although in the procedure tight clusters are robust to the selection of *K*, the starting value of *K* (denoted as k_0 in the original paper) in subsampling clustering largely defines the tightness of clusters to be obtained. As suggested by the authors, the starting value of k_0 is set as the number of clusters desired plus five in this paper. The method has shown particular success in gene clustering of

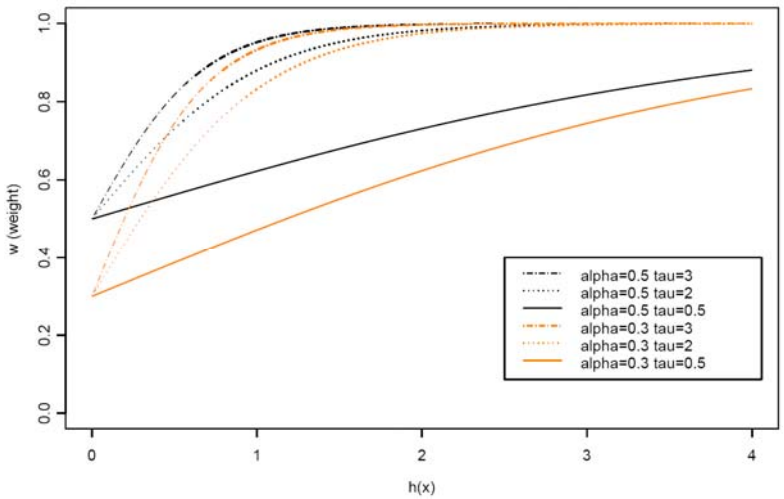
microarray data where abundant numbers of scattered genes are present, complex gene inter-correlations exist and a lack of prior knowledge of the exact number of clusters *K*. Some possible drawbacks of tight clustering include heavy computation due to repeated subsampling and that different results may be obtained in two different runs due to the nature of random subsampling.

CLICK CLICK (Sharan et al., 2003) is a popular clustering algorithm developed for gene expression profiles. No prior assumptions are made on the structure or the number of the clusters. The algorithm utilizes graph-theoretic and statistical techniques to identify tight groups of highly similar elements (kernels). A series of statistical procedures are then applied to expand the kernels to form clusters. The method contains a homogeneous parameter $0<\lambda<1$. For larger λ, tighter clusters are obtained and more genes are left without being clustered. Since this method cannot specify the number of clusters *K* as other methods do, we performed 16 clustering with smaller $\lambda=0.20, 0.22,...,0.50$ to generate the "CLICK (low)" result in Figure 4 in the manuscript and similarly larger $\lambda=0.50, 0.52,...,0.82$ to generate "CLICK (high)".

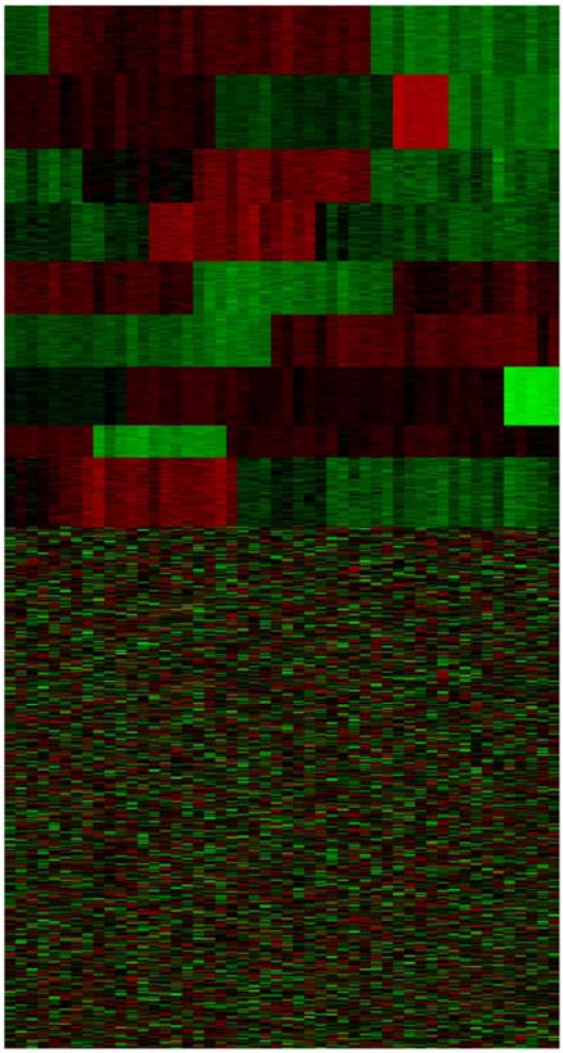
SUPPLEMENT FIGURES AND TABLES



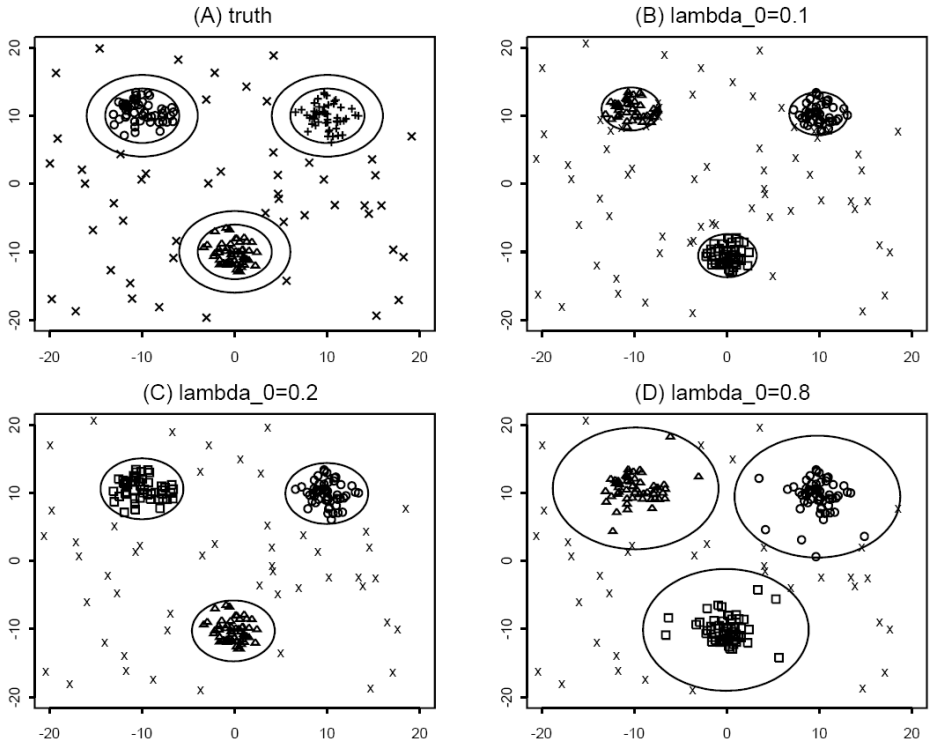
Supplement Figure 1: Heatmap of cell cycle related genes distributed in six functional categories. The log-based relative expression intensities are expressed by gradient colors (red: positive, green: negative).



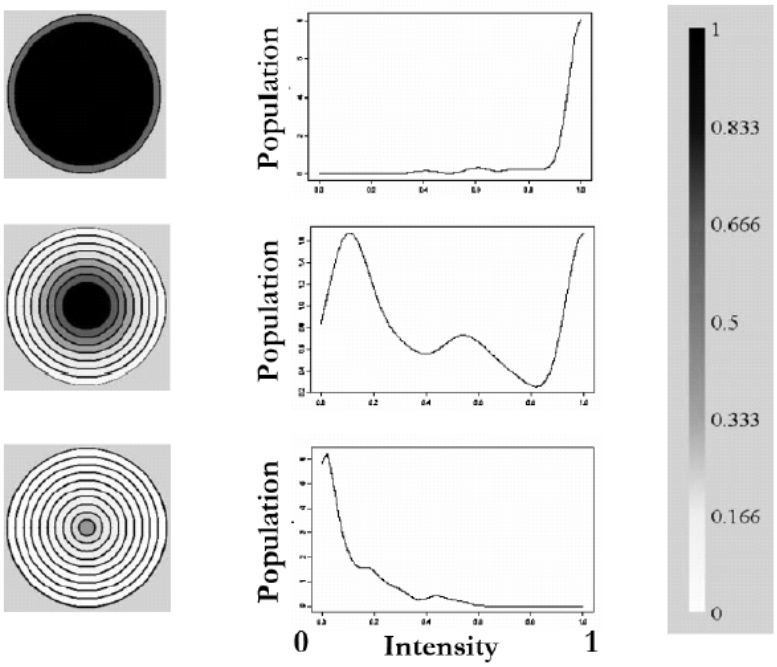
Supplement Figure 2: Design of weights as a function of h with different settings of α and τ . Smaller α and τ gives more dramatic weight reduction.



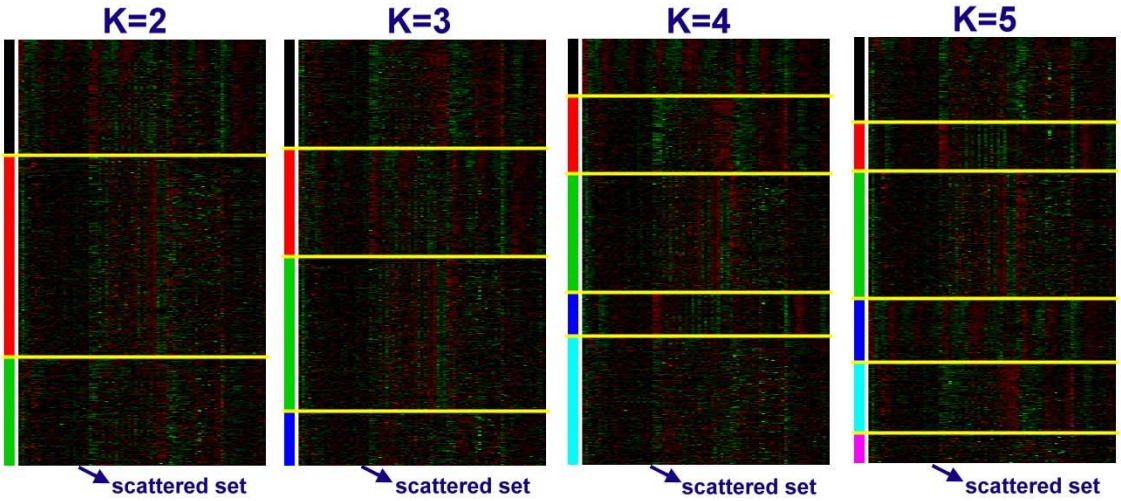
Supplement Figure 3: Heatmap of the second simulation data set. Nine clusters of 392 genes and 392 additional scattered genes are simulated.



Supplement Figure 4: (A) The underlying truth of the simulated data. Clustered points are simulated within the inner circles and noise points are outside the outer circles to avoid any confusion. (B)-(D) Clustering results of P-*K*means with λ_0=0.1, 0.2 and 0.8 (k=3). Larger λ_0 produces larger and looser clusters.



Supplement Figure 5: Left Panel: Quantile maps for visualizing the distributions. Three examples are demonstrated (upper: distribution concentrated close to 1. middle: bi-modal distribution close to 0 and 1. lower: distribution concentrated close to 0.) Middle panel: The corresponding density plots of the three examples. Right panel: The grey-color scale representing intensity strength.



Supplement Figure 6: Heatmap of gene clustering with K=2-5. It shows a multi-resolution clustering result.

Supplement Table 1: 104 Verified cell cycle related genes.

M/G1 Boundary: AGA1 ASH1 CDC46 CDC47 CDC6 CHS1 CLN3 CTS1 EGT2 FUS1 MFA2 PCL2 PCL9 RME1 SIC1 SST2 STE2 SWI4 TEC1
Late G1, SCB regulated: CLN1 CLN2 CSD2 CHS3 FKS1 CWH53 GAS1 HO KAR4 KRE6 MNN1 PCL1 PSA1 SWE1 TIP1 VAN2 GOG5
Late G1, MCB regulated: ASF1 ASF2 CDC21 CDC45 CDC8 CDC9 CLB5 CLB6 DBF4 DPB2 DPB3 GIC2 MCD1 MSH2 MSH6 NIK1 HSL1 PDS1 PMS1 POL1 POL12 POL2 POL3 CDC2 POL30 PRI1 PRI2 RAD17 RAD27 RAD51 RAD54 RFA1 RFA2 RFA3 RNR1 RNR3 SPC110 NUF1 SPC42 SPK1 SRS2 HPR5 UNG1
S-phase: HHT1 HHT2 HHF1 HHF2 HTA1 HTA2 HTB1 HTB2
S/G2-phase: CDC14 CIK1 CLB3 CLB4 CWP1 CWP2 KAR3 NUM1 TIR1
G2/M-phase: ACE2 ASE1 CDC20 CDC5 CLB1 CLB2 DBF2 FAR1 KIN3 MOB1 YRO2 (MST1) MRH1 (MST2) SED1 SPO12 SWI5