BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 3 2010, pages 341–347
doi:10.1093/bioinformatics/btp671

Gene expression

Dynamically weighted clustering with noise set

Yijing Shen[1], Wei Sun[2] and Ker-Chau Li[1,3,*]

[1]Department of Statistics at University of California, Los Angeles, CA 90095, [2]Department of Biostatistics, Genetics, University of North Carolina, NC 27516, USA and [3]Institute of Statistical Science, Academia Sinica, Taipei, Taiwan, Republic of China

Received on June 16, 2009; revised on October 28, 2009; accepted on December 3, 2009

Advance Access publication December 9, 2009

Associate Editor: Joaquin Dopazo

ABSTRACT

Motivation: Various clustering methods have been applied to microarray gene expression data for identifying genes with similar expression profiles. As the biological annotation data accumulated, more and more genes have been organized into functional categories. Functionally related genes may be regulated by common cellular signals, thus likely to be co-expressed. Consequently, utilizing the rapidly increasing functional annotation resources such as Gene Ontology (GO) to improve the performance of clustering methods is of great interest. On the opposite side of clustering, there are genes that have distinct expression profiles and do not co-express with other genes. Identification of these scattered genes could enhance the performance of clustering methods.

Results: We developed a new clustering algorithm, Dynamically Weighted Clustering with Noise set (DWCN), which makes use of gene annotation information and allows for a set of scattered genes, the noise set, to be left out of the main clusters. We tested the DWCN method and contrasted its results with those obtained using several common clustering techniques on a simulated dataset as well as on two public datasets: the Stanford yeast cell-cycle gene expression data, and a gene expression dataset for a group of genetically different yeast segregants.

Conclusion: Our method produces clusters with more consistent functional annotations and more coherent expression patterns than existing clustering techniques.

Contact: yshen@stat.ucla.edu

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

1 INTRODUCTION

Gene expression variations are expected to provide insights into the cellular roles of the genes in an organism. Clustering analysis methods are useful tools for extracting information from massive microarray gene expression datasets by grouping together genes that share similar expression profiles (Eisen *et al.*, 1998). Such co-expressed genes are likely to be regulated by common cellular signals. The result of clustering analysis can be used to reveal the organization of the genes in different biological processes and to predict the functions of new or poorly annotated genes (Marcotte *et al.*, 1999). There are many clustering methods available: the hierarchical clustering (Eisen *et al.*, 1998; Jain *et al.*, 1999), K-means algorithm (Tavazoie *et al.*, 1999), Self Organization Maps (SOM) (Conrads *et al.*, 2003; Tamayo *et al.*, 1999), mixture model approaches (Ghosh and Zhong, 2003; MacLachlan and Basford, 1988; Yeung *et al.*, 2001) and many others (Cheng *et al.*, 2004; Hastie *et al.*, 2000; Tseng, 2007). To improve the performance of such general purpose clustering methods in microarray studies, some researchers have proposed two additional ideas: (i) allowing for a set of scattered noise genes to remain un-clustered (Hanisch *et al.*, 2002; Pan, 2006; Thalamuthu *et al.*, 2006; Tseng, 2007; Tseng and Wong, 2005); (ii) incorporating the functional annotation information of the genes (Basu *et al.*, 2004; Cheng *et al.*, 2004; Hanisch *et al.*, 2002; Pan, 2006; Segal *et al.*, 2003; Tseng, 2007).

Different strategies have been used to incorporate functional annotations. Pan (2006) used a model-based clustering method by assuming that genes in one Gene Ontology (GO) term have the same prior probability of belonging to one cluster. This may not be optimal because genes sharing the same functional annotation may have different expression patterns. Indeed, by studying the coexpression patterns of protein complexes in yeast for four large scale microarray gene expression databases, Liu *et al.* (2008) found that except for large protein complexes such as cytoplasmic and mitochondrial ribosomal complexes, proteasome, or ATP synthase, most genes from the same complexes do not have strong correlation. In addition, the method does not allow for scattered noise genes. Tseng (2007) proposed a new clustering method, Penalized and Weighted K-means (PW-Kmeans), which is an extension of K-means that incorporates functional annotations and allows for scattered genes. The loss function of PW-Kmeans is the summation of the weighted dispersions of each clustered gene augmented by penalties for each scattered gene. The weight of each gene is computed based on its minimum distance to any known functional group (e.g. a pathway) where the distance is defined as the average square distance from this gene to all the genes in the group. Intuitively, if one gene is close to one functional group, the weight is small, thus the gene is less likely to be claimed as a scattered gene. Note that all weights are fixed throughout the clustering procedure. The weight assigned to a given gene remains the same no matter which cluster the gene is assigned to. Therefore, while weighting does help identifying the scattered genes, it does not enhance the clustering of genes with similar functions. Furthermore, as discussed above, functionally related

*To whom correspondence should be addressed.

genes are not necessarily co-expressed. Therefore, the weights predetermined by the average distance to every gene in a functional group may not be appropriate.

In this article, we develop a novel weighted clustering method, Dynamically Weighted Clustering with Noise set (DWCN), which allows for the presence of scattered noise genes and makes use of functional annotation data in a different way. Similar to the PW-Kmeans, the loss function we use is also the summation of the weighted dispersions of the clustered gene plus penalties for scattered gene. However, unlike the PW-Kmeans, which uses gene-specific weights, DWCN uses cluster-specific weights: all the genes within one cluster share the same weight. The weight of one cluster represents a penalty term which is inversely proportional to the homogeneity between the cluster and the functional categories. Both the cluster memberships and the weights are iteratively updated in the clustering algorithm.

The remainder of this article is organized as follows. We first describe our method and the datasets we used. Then, using both simulated data and real data, we present and compare the results of DWCN with three other clustering algorithms: K-means, tight clustering and PW-Kmeans. We use the GO system to define functional categories throughout our discussion, although our method is general enough to incorporate any functional category information such as pathway annotation.

2 METHODS

A flow chart describing our method is shown in Figure 1. We first initialize a group of clusters by forming tight clusters within each functional group (e.g. GO term) and within the set of remaining genes. Then, we iteratively select a gene and decide whether it should belong to one of the clusters or the noise set. The decision is made by evaluating a loss function which involves cluster-specific weights. The weights are simultaneously updated together with the cluster memberships so that if the genes within one cluster share similar biological functions, the cluster is assigned a smaller weight in the loss function. Because the weights depend on the cluster memberships, in order to find the optimal solution, one may compare the loss function for all possible ways of cluster membership reassignment. This, however, is not computationally feasible. To overcome this difficulty, we used simulated annealing to minimize the loss function (Bryan *et al.*, 2006; Chakraborty, 2005).

2.1 Model setup

Let $C=\{x_1,x_2,\ldots,x_n\}\subset R^p$ be a set of n data points in a p-dimensional space (e.g. expression of n genes measured in p conditions). The purpose of our clustering method is to partition these n genes into r clusters, $C_1, C_2, \ldots, C_r$, and a set of scattered noise genes, S. We define the loss function W as

$$W(C;r,\lambda)=\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{r}u_{ij}w_j d(x_i,C_j)\right)^{1-\beta_i}(w_j\lambda)^{\beta_i}\right], \qquad (1)$$

where β_i is 1 if x_i is a scattered point, and 0 otherwise. Let u_{ij} be the cluster membership such that $u_{ij}=1$ if $x_i\in C_j$, and 0 otherwise. The parameter λ is the 'penalty' for the scattered genes. It can be viewed as a tuning parameter controlling the size of the noise set because larger λ lead to smaller noise sets. We take the distance $d(x_i,C_j)$ to be squared L_2 norm from x_i to the center of the cluster C_j:

$$d(x_i,C_j)=||x_i-\frac{(\sum_k u_{kj}x_k)}{|C_j|}||^2,$$

where $|C_j|$ is the number genes in cluster C_j. The 'weight' w_j for cluster j is defined via the minimum hyper-geometric P-value, g_j, obtained from testing

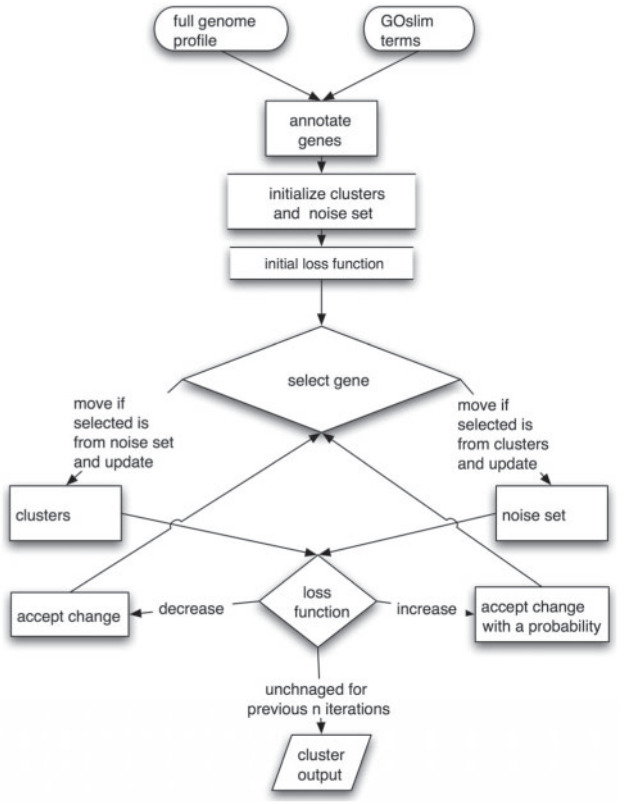

Fig. 1. Flow chart of the DWCN algorithm.

the functional enrichment of cluster j with respect to each of the GO terms. Specifically, let

$$g_j=\min_m \sum_{y=|G_m\cap C_j|}\binom{|G_m|}{y}\binom{n-|G_m|}{|C_j|-y}\Big/\binom{n}{|C_j|}, \qquad (2)$$

where $|G_m|$ is the number of genes in the m-th GO term. Then,

$$w_j=\begin{cases}P & \text{if } g_j<0.05/\kappa\\ 1 & \text{if } g_j>0.05\\ b+a(\log g_j) & \text{otherwise}\end{cases} \qquad (3)$$

where $0.1<P<0.5$ is a fixed parameter, κ is the number of GO terms, and a,b are scaling constants, which can be solved for by noting that they are the slope and intercept of the line passing through [log(0.05), 1] and [log(0.05/κ), P]. Thus,

$$a=\frac{1-P}{\log\kappa},\quad b=\frac{\log(\kappa/0.05)+P^*\log(0.05)}{\log(\kappa)}.$$

In summary, the loss function is a weighted sum of the within cluster distances plus penalties for the noise set. The weight of one cluster measures the over-representation of genes within this cluster in a group of functional categories. In other words, it is a measure of homogeneity between clusters and functional categories.

2.2 Initialization

First, all annotated genes are clustered using the tight clustering method (Tseng and Wong, 2005) within each GO term. Only the tightest resulting cluster for each GO term is kept, defining the initial cluster configuration $\{C_j\}_G$ where j stands for j-th cluster and G indicates these clusters are from GO terms. The remaining annotated genes together with the non-annotated

genes define the initial noise set $S^{(0)}$. We refine the initial configuration based on the distance between each gene in $S^{(0)}$ and its nearest cluster. Specifically, let x_i be one gene in $S^{(0)}$, and let d_i be its distance to the nearest cluster. Given two constants τ_1 and τ_2 ($\tau_1 \leq \tau_2$), x_i is assigned to its nearest cluster if $d_i < \tau_1$, and x_i is assigned to a set T if $d_i > \tau_2$. Next we identify tight clusters from T, denoted as $\{C_j\}_T$. The number of tight clusters in T is chosen so that it gives the smallest initial loss function. The initial clusters are obtained by combining $\{C_j\}_G$ and $\{C_j\}_T$. The remaining genes in $S^{(0)}$ constitute the initial noise set.

With the initial classification of the cluster genes and the noise genes, the tuning parameter λ can be estimated by minimizing the summation of the misclassification rates for the clustered genes and the noise genes.

In the case where the GO terms used have few overlapping genes, we estimate the number of clusters r by the total number tight clusters within each GO term and within the un-annotated gene set. When a larger set of GO terms are used, the sizes of GO terms may get smaller, and many genes may belong to multiple GO terms. In such cases, the default initialization of one cluster per GO term may be impractical. We propose to cluster GO terms by hierarchical clustering, and then merge the smaller GO terms by choosing a cutoff of the hierarchical tree. An example is presented in the Results section.

2.3 Loss function minimization

Because of the complicated discontinuous, non-linear relationship between the weights w_j and the cluster membership u_{ij}, we optimize the loss function by simulated annealing (Bryan *et al*., 2006; Chakraborty, 2005).

After initialization, we obtain several initial clusters, together with a set of 'noise' genes. The initial values for u_{ij} and β_j are defined at the initialization. Next, we screen all the genes. In each iteration, one gene is randomly selected. If it comes from the noise set, we tentatively assigned it to the nearest cluster, as determined by D_{ij} (the distance between gene i and cluster j) where $D_{ij} = w_j d(x_i, C_j)$; otherwise, it is tentatively moved to the noise set. After updating the loss function [Equation (1)], the proposed change is always accepted if the loss function decreases, otherwise it is accepted with probability $\exp(-|W^{(l+1)} - W^{(l)}|/T)$, where T is the so-called *annealing temperature*. In order to achieve convergence, after an initial 'burn-in' period, T is slowly decreased at each iteration by $T = T/(1+\sigma)$ where σ is a constant controlling the annealing schedule (Bryan *et al*., 2006).

2.4 Cluster evaluation

In order to evaluate the final clustering results, we use Rand indexes (Hubert and Arabie, 1985; Rand, 1971) and Weighted Rand indexes (Thalamuthu *et al*., 2006) to assess the homogeneity between the resulting clusters and the function categories or the true clustering structure. Intuitively, the rand index equations can be explained by the number of agreements between two partitions divided by the total number of agreements and disagreements between two partitions.

A few versions of Rand indexes are used. Rand_1 [Equation (4)] computes the similarity between GO and clustering partitions including the un-annotated set as one of the GO term, and the noise set as one of the clusters. It treats noise genes and clustered genes with equal importance and thus it favors methods without scattered genes. Rand_2 excludes all noise and un-annotated genes, and it is biased against methods with a noise set (Thalamuthu *et al*., 2006). RandW defined in Equation (5), is a weighted average between the two measures, which is used for validating the simulation by comparing the similarity of the true partitions and the clustering results.

$$Rand_1(G,r) = \frac{\sum_{i=1}^{G+1}\sum_{j=1}^{r+1}\binom{Y_{ij}}{2} - \sum_{i=1}^{G+1}\binom{Y_{i\bullet}}{2}\sum_{j=1}^{r+1}\binom{Y_{\bullet j}}{2}/\binom{Y}{2}}{0.5\left[\sum_{i=1}^{G+1}\binom{Y_{i\bullet}}{2} + \sum_{j=1}^{r+1}\binom{Y_{\bullet j}}{2}\right] - \sum_{i=1}^{G+1}\binom{Y_{i\bullet}}{2}\sum_{j=1}^{r+1}\binom{Y_{\bullet j}}{2}/\binom{Y}{2}}, \quad (4)$$

where $Y_{i\bullet} = \sum_{j=1}^{r+1} Y_{ij}$, $Y_{\bullet j} = \sum_{i=1}^{G+1} Y_{ij}$.

Here G is the number of GO terms, G+1 indexes the un-annotated set, r is the number of clusters, $r+1$ indexes the scattered set, and Y_{ij} represents the number of genes belonging to the i-th GO term and j-th cluster.

The weighted index is simply a linear combination of Rand_1 and Rand_2 weighted by

$$RandW(G,r) = \alpha \times Rand_1(G,r) + (1-\alpha) \times Rand_2(G,r), \quad (5)$$

where $\alpha = \left(Y_{(G+1)\bullet} + Y_{\bullet(r+1)} - Y_{(G+1)(r+1)}\right)/Y$.

In the absence of the knowledge about true clusters (e.g. for the real datasets), we evaluate the concordance between clustering results and GO terms using another Rand index, Rand_3, which treats each gene in the noise term that comes from a GO term as an individual cluster.

In addition, we also employ a cross-validation approach to evaluate how well the functions of the un-annotated genes were predicted. Specifically, the GO annotations of a subset of genes were deleted prior to clustering. After clustering, the accuracy rate is computed based on how well the testing genes are grouped into the GO terms they actually belong to.

Moreover, we also measure the concordance of the GO terms and the clustering results by quantifying the clustering homogeneity within each GO term through its entropy,

$$e_i = \sum_{j=1}^{r} p_j \log \frac{1}{p_j},$$

where r is the number of clusters, and p_j are defined as $p_j = Y_{ji}/m_i$, with m_i being the total number of non-scattered genes which belong to the ith GO term. For the purpose of consistently comparing different clustering results, entropies of each GO term are pooled together. The total entropy of each clustering method is represented by

$$E = \sum_{i=1}^{G} e_i.$$

The higher the entropy is, the lower the homogeneity.

Because our method used the GO data as prior information to form clusters, evaluation measures such as the entropy for assessing the homogeneity of the clusters with respect to GO terms can introduce some bias. Thus the reported better performance of our method under GO term enrichment related criteria are anticipated.

This also raises the concern of 'overfitting'—namely, our method may fine-tune the clusters to better capture the GO term structure to the extent of over-compromising the quality of clusters. However, as we shall demonstrate by cluster tightness and size criteria, the clusters obtained from our method are indeed tighter and no smaller than the clusters obtained from other methods. Thus our method produces better clusters in terms of quality of the clusters while achieving the aim of better GO term enrichment.

3 RESULTS

3.1 Simulation

We first assess the performance of our method using a simulated dataset of sample size m=50. Consider $r = 10$ clusters corresponding to 10 hypothetical GO terms. We first generated the centers (μ_l, $l = 1, \ldots, r$) of the clusters from normal distribution $N(0, I_m)$, where I_m is an identify matrix of size $m \times m$. Subsequently, for each cluster, expression profiles of 50 genes were generated from $N(\mu_l, \sigma^2 I_m)$. Separately, 500 'noisy' gene expression profiles were generated from the uniform distribution, $U_m(-2, 2)$, making a total of 1000 genes for the simulation. The genes were 'annotated' so that 25 out of 50 genes in each cluster and 10 'noise' genes were assigned to each GO term/category, i.e. 35 genes per GO term. The remaining 650 genes

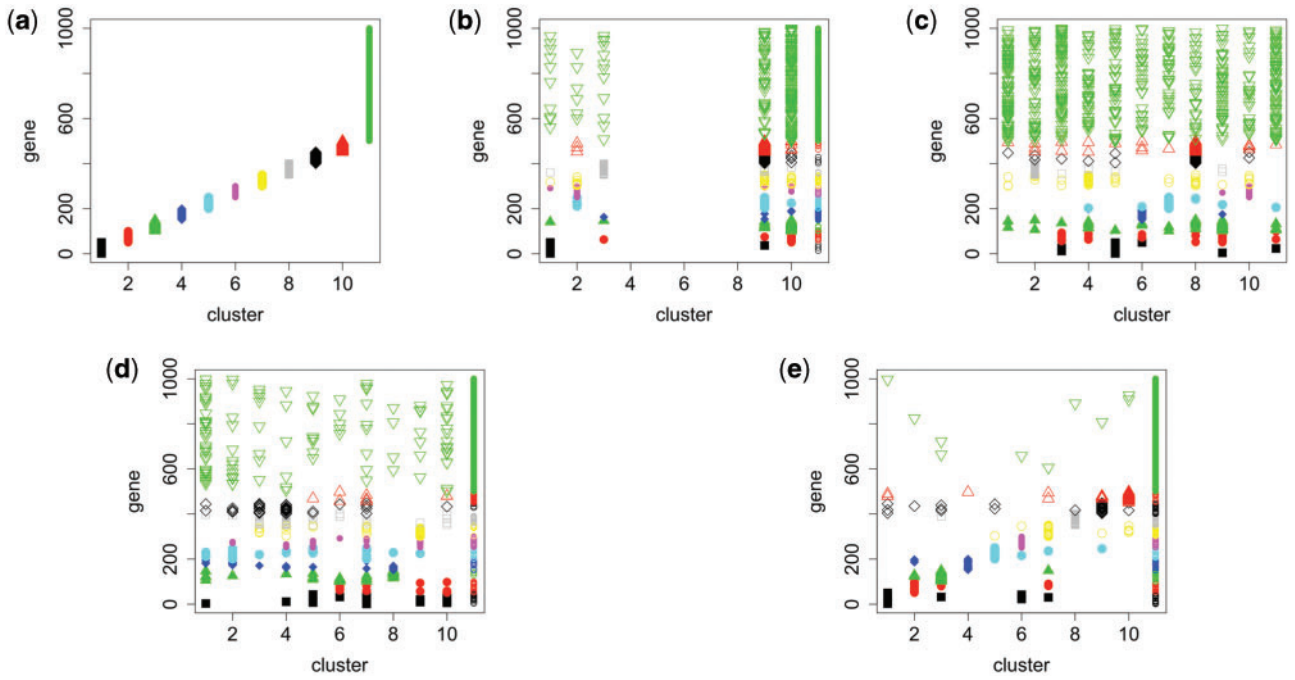

Fig. 2. (**a**) The original clusters of simulated dataset. Gene IDs are displayed by *y*-axis and the *x*-axis displays the clusters. Each true cluster receives a color/shape. Gene IDs from 501 to 1000 are noise points. (**b**) Tight clustering. (**c**) Standard K-means with K=10. (**d**) PW-Kmeans. (**e**) DWCN.

were considered to be un-annotated. More details of the simulation are described in Simulation section in Supplementary Material. The true cluster membership is shown in Figure 2a. The 1000 simulated gene expression profiles are ordered in such a way that the first 500 genes belong to clusters and the rest of them are in the noise set. Figure 2b–e describe the results from tight clustering, traditional K-means, PW-Kmeans and DWCN, respectively. Tight clustering only finds five out of the 'requested' 10 clusters. Because K-means does not allow scattered genes, it is unable to find the correct clusters. PW-Kmeans finds 10 clusters and one noise set; the improvement from K-means is slight. In contrast, DWCN adequately distinguishes most of the noise and forms more correct clusters (see Fig. 2e).

We used the Rand index to compare the clustering results and the underlying true clusters (Table 1, panel A]. DWCN outperforms K-means clustering, PW-Kmeans and tight clustering based on Rand_1 [Equation (4)], Rand_2 and the weighted Rand index [Equation (5)].

3.2 Yeast cell-cycle data

We tested the performance of DWCN on the yeast cell-cycle dataset (Spellman, 1998). A full description and complete datasets are publicly available at http://cellcycle-www.standford.edu. Overall, the dataset consists of 5878 genes on 73 experimental conditions, missing values imputed by the K-Nearest Neighbor (KNN) method (Troyanskaya *et al*., 2001).

3.2.1 GO slim We used GO slim terms of *biological process* as the functional categories, and only concentrate on the GO terms of size 20–300. Altogether 2789 out of 5878 genes were annotated using 27 GO terms. The number of clusters r is set to 30, which comes from the number of GO terms plus the three tight clusters generated from the noise set. A heat map of the expression profiles for the three additional clusters was shown in Supplementary Figure 2.

We ran a total of 24 000 iterations and convergence of the loss function is shown in Supplementary Figure 4. We let the lower bound of the weight be P=0.3.

We first evaluate the performance of different clustering methods in terms of their concordances with the GO term partitions. We use hypergeometric P-values, entropy and several Rand indices to measure the degree of concordance. Table 1 (Panel B) demonstrates that the clustering result of DWCN is most consistent with GO terms partitions. For each GO term, we also evaluate whether it has significant overlap with each of the clusters by a hypergeometric P-value, which is adjusted for multiple comparison by the Bonferroni correction. The smallest P-value across the r clusters is derived for each GO term. DWCN gives the most significant hypergeometric P-values across the majority of the 27 GO terms compared to the other methods (Supplementary Table 4).

As shown in Table 1 (Panel B), if the GO terms are randomly generated (R-DWCN), the overlap between the resulting clusters and the GO terms decreases.

Figure 3a shows distributions of the prediction accuracy rates from the cross-validation study where 20% of the GO term prior annotations are removed. Across 10 validation sets, DWCN achieves significantly higher accuracy rates than PW-Kmeans.

Finally, we measure the 'tightness' of the clusters produced by different methods using the within-cluster mean squared distances. As shown in Supplementary Figure 3d, the sizes of the DWCN clusters are not smaller than those of clusters produced by tight clustering and PW-Kmeans, yet the DWCN clusters are 'tighter' (Supplementary Fig. 3a).

Table 1. Rand indices comparisons on simulation

Panel A			
	Rand index 1	Rand index 2	Weighted rand
DWCN	0.53	0.72	0.59
K-means	0.10	0.47	0.29
Tight	0.25	0.35	0.29
PWK-means	0.37	0.23	0.32
Panel B			
	Rand index 3	Total entropy	No. of over-represented GO terms ($P < 0.01$)
DWCN	0.136	57.72	27
R-DWCN	0.026	83.47	7
K-means	0.06	69.57	18
Tight	9.5×10^{-5}	109.95	12
PWK-means	0.004	85.3	1
Panel C			
	Rand index 3	Total entropy	No. of over-represented GO terms ($P < 0.01$)
DWCN	0.23	71.33	32
K-means	0.026	87.46	21
Tight	0.01	76.7	18
PWK-means	0.0019	101.83	0
Panel D			
	Rand index 3	Total entropy	No. of over-represented GO terms ($P < 0.01$)
DWCN	0.14	47.4	27
K-means	0.049	55.63	19
Tight	0.013	53.56	14
PWK-means	0.008	85.05	2

Panel A: Comparisons made between DWCN, K-means, tight clustering and PWK-means algorithms. Higher indices values imply better consistency between the identified clusters and the underlying true clusters. Panels B and C: Evaluation of the clusters identified from yeast cell-cycle data. Rand index 3, total entropy and total number of over-represented GO terms comparisons between DWCN, K-means, tight clustering and PWK-means by using (Panel B) 27 GO slim terms and (Panel C) 214 GO terms as functional categories correspondingly. Panel D: Same evaluation scheme used as (Panels B and C) for the clusters identified from yeast segregants data using 27 GO slim terms.

3.2.2 Large set of GO classes We also used a large set of GO classes, 214 GO terms, as functional categories to demonstrate how our approach could be used with GO terms from the bottom level of the GO hierarchy. We defined the distance between two GO terms as $\text{mean}_{x\in GO1, y\in GO2}[1\text{-corr}(x,y)^2]$. Other distance measures can also be applied. We cluster GO terms by hierarchical clustering, and choose the cutoff of the tree by Dynamic Tree Cut method (Langfelder *et al.*, 2008). This technique detects clusters in a dendrogram depending on their shape, and it is capable of identifying nested clusters, and automatically detects the optimal number of clusters. With this set of GO terms as prior, the number of clusters is chosen to be 35 (32 GO related and three tight clusters) with a minimum of 24 and a maximum of 357 genes in the merged GO categories. A total of 3169 genes were annotated. DWCN results were superior to those obtained using other methods tested on the 214 GO Term dataset as shown in Table 1 (Panel C) and Supplementary Figure 3b.

3.3 Yeast segregant data

We tested our method on another large-scale gene expression data coming from 112 yeast segregants in a cross between two parental strains BY and RM (Brem and Kruglyak, 2005; Brem *et al.*, 2005). We use this dataset to test the robustness of DWCN. The same 27 GO slim terms is used as functional categories. The number of clusters is chosen to be 29 (27 GO related and two tight clusters). We applied the all mentioned clustering methods on this dataset. Again, we are able to demonstrate the superiority of DWCN. The results are shown in Table 1 (Panel D), Supplementary Figure 3b and c.

4 DISCUSSION

In this article, we proposed a novel clustering method, DWCN, to identify clusters of genes with both coherent expression profiles and similar biological functions. Our method exploits the known biological functions when evaluating cluster-specific weights in the loss function. The weights are updated according to the cluster memberships so that if the genes within one cluster share similar biological functions, the cluster is assigned a smaller weight in the loss function. This construction allows the weights to refine clusters under more meaningful biological contexts. At the same time, the un-annotated or ungrouped genes can be partitioned into one of the existing clusters, or simply left un-clustered.

In the initialization of our method while constructing tight cluster within each GO term, one gene is allowed to be assigned to multiple clusters. However, during the iterative updates of our algorithm, one gene is only assigned to one GO term. This restriction is necessary; otherwise, the loss function will increase. A possible extension is to adjust for the number of clusters it belongs. This is a direction worth further research.

As demonstrated in the 'Results' section, our method can generate clusters that are both biologically coherent and statistically tight. Importantly, we have shown that the clusters obtained from our method are tighter and no smaller than the clusters obtained from other methods. In other words, our method produces better clusters in terms of their sizes and tightness, which is unrelated with GO term information. Several parameters used by the algorithm i.e. the number of clusters, and penalization constant for noise genes, can be fine tuned to meet specific needs in application. We have derived some guidelines for parameter setting. Our results verified that the clusters generated by DWCN are more likely to overlap with biological categories as compared with other clustering methods. Several criteria of evaluating these comparisons were employed.

Our method requires an initial cluster assignment to start the iteration. We employ the tight clustering method together with biological annotation to obtain the initial clusters. If no gene function annotations are available, DWCN is equivalent to the tight clustering method. In cases where prior functional categories are small and with significant overlaps (e.g. GO terms from the bottom of the GO

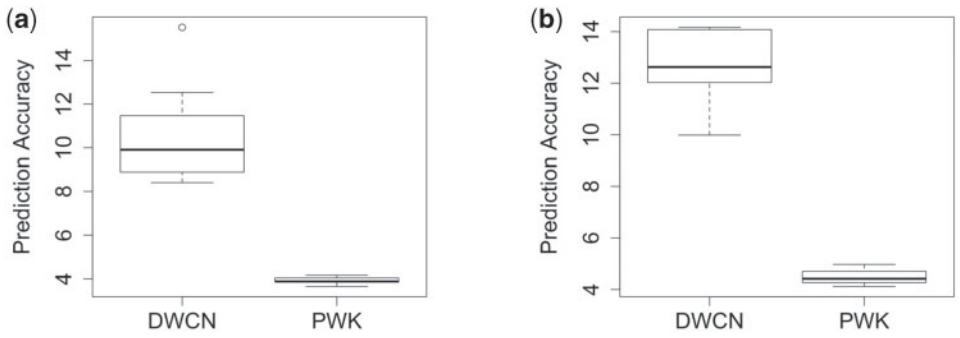

Fig. 3. Prediction accuracy percentage distribution for (**a**) yeast cell cycle and (**b**) yeast segregants data with 27 GO slim terms.

hierarchy), we propose to merge them by hierarchical clustering followed by cutting the tree at an appropriate height.

The recurrent issue about the optimal number of clusters is a topic we have not fully explored in this report. We used hierarchical clustering followed by dynamic tree cut to select a certain number of GO terms, which guides the clustering procedure. However, as one referee suggested, sometimes rigorous and data-driven criteria may be preferable. Methods such as rrediction strength (Tibshirani *et al*., 2001) or prediction based re-sampling method (Dudoit and Fridlyand, 2002) could be tested. In addition, methods based on stability criteria (Bertoni and Valentini, 2007, 2008; Ho, 1998; Smolkin and Ghosh, 2003) could also be explored.

The proposed clustering method is not restricted to clustering genes. The same idea can be extended for clustering conditions such as clustering tumor samples. Gene expression profiles have been widely used to discriminate different sub-types, or clinical outcomes of human cancers (Furey *et al*., 2000; Ghosh, 2002; Golub *et al*., 1999; Shipp *et al*., 2002; Singh *et al*., 2002) In a dataset where the class labels for some cancer patients are missing or incomplete, we can use the known labels to generate 'function categories', and apply our clustering algorithm so that patients having similar expression profiles can be clustered consistently with their class labels.

Funding: NSF grants of USA DMS0406091 and DMS-0707160 and NSC grants of Taiwan NSC95-3114-P-002-005-Y and NSC97-2627-P-001-003 (to K.-C.L.).

Conflict of interest: none declared.

REFERENCES

Basu,S. *et al*. (2004) A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Seattle, WA, pp. 59–68.

Bertoni,A. and Valentini,G. (2007) Model order selection for bio-molecular data clustering. *BMC Bioinformatics*, **8**(Suppl. 2), S7.

Bertoni,A. and Valentini,G. (2008) Discovering multi-level structures in bio-molecular data through the Bernstein inequality. *BMC Bioinformatics*, **9**(Suppl. 2), S4.

Brem,R.B. and Kruglyak,L. (2005) The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proc. Natl Acad. Sci. USA*, **102**, 1572–1577.

Brem,R.B. *et al*. (2005) Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, **436**, 701–703.

Bryan,K. *et al*. (2006) Application of simulated annealing to the biclustering of gene expression data. *IEEE Trans. Inf. Technol. Biomed.*, **10**, 519–525.

Chakraborty,A. (2005) Biclustering of gene expression data by simulated annealing. In *Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region*. IEEE, Washington DC.

Cheng,J. *et al*. (2004) A knowledge-based clustering algorithm driven by Gene Ontology. *J. Biopharmaceut. Statist.*, **14**, 687–700.

Conrads,T.P. *et al*. (2003) Cancer diagnosis using proteomic patterns. *Expert Rev. Mol. Diagnost.*, **3**, 411–420.

Dudoit,S. and Fridlyand,J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol.*, **3**, 1–21.

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

Furey,T.S. *et al*. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Ghosh,D. (2002) Singular value decomposition regression models for classification of tumors from microarray experiments. *Pac. Symp. Biocomput.*, **98**, 18–29.

Ghosh,J. and Zhong,S. (2003) A unified framework for model-based clustering. *J. Machine Learn. Res.*, **4**, 1001–1037.

Golub,T.R. *et al*. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Hanisch,D. *et al*. (2002) Co-clustering of biological networks and gene expression data. *Bioinformatics*, **18**(Suppl. 1), S145–S154.

Hastie,T. *et al*. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol.*, **1**, RESEARCH0003.

Ho,T. (1998) The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Machine Intell.*, **20**, 832–844.

Hubert,J. and Arabie,P. (1985) Comparing partitions. *J. Classif.*, **2**, 193–218.

Jain,A.K. *et al*. (1999) Data clustering: a review. *ACM Comput. Surveys*, **31**, 264–323.

Langfelder,P. *et al*. (2008) Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics*, **24**, 719–720.

Liu,C.T. *et al*. (2008) Patterns of co-expression for protein complexes by size in *Saccharomyces cerevisiae*. *Nucleic Acids Res.*, **37**, 526–532.

MacLachlan,G. and Basford,K. (1988) *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

Marcotte,E.M. *et al*. (1999) A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83–86.

Pan,W. (2006) Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, **22**, 795–801.

Rand,W.M. (1971) Objective criteria for the evaluation of clustering methods. *J. Am. Statist. Assoc.*, **66**, 846–856.

Segal,E. *et al*. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**, 166–176.

Shipp,M.A. *et al*. (2002) Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.

Singh,D. *et al*. (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**, 203–209.

Smolkin,M. and Ghosh,D. (2003) Cluster stability scores for microarray data in cancer studies. *BMC Bioinformatics*, **4**, 36.

Spellman,P.T. *et al*. (1998) Comprehensive identification of cell cycle—regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridizationh. *Mol. Biol. Cell*, **9**, 3273–3297.

Tamayo,P. *et al*. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tavazoie,S. *et al*. (1999) Systematic determination of genetic network architecture. *Nature Genet.*, **22**, 281–285.

Thalamuthu,A. *et al*. (2006) Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, **22**, 2405–2412.

Tibshirani,R. *et al*. (2001) Cluster validation by prediction strength. *Technical Report*. Department of Statistics, Stanford Univeristy.

Troyanskaya,O. *et al*. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Tseng,G.C. (2007) Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, **23**, 2247–2255.

Tseng,G.C. and Wong,W.H. (2005) Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, **61**, 10–16.

Yeung,K.Y. *et al*. (2001) Model-based clustring and data transformations for gene expression data. *Bioinformatics*, **17**, 997–987.