# On the Effectiveness of Constraints Sets in Clustering Genes

Erliang Zeng, Chengyong Yang, Tao Li, and Giri Narasimhan
Bioinformatics Research Group (BioRG)
School of Computing and Information Sciences
Florida International University
Miami, Florida, 33199, USA
E-mail:{ezeng001, cyang01, taoli, giri}@cs.fiu.edu
Current address for author C. Yang: Applied Biosystems Inc., Foster City, CA.
E-mail: chengyong_yang@appliedbiosystems.com

***Abstract*—In this paper, we have modified a constrained clustering algorithm to perform exploratory analysis on gene expression data using prior knowledge presented in the form of constraints. We have also studied the effectiveness of various constraints sets. To address the problem of automatically generating constraints from biological text literature, we considered two methods (cluster-based and similarity-based). We concluded that incomplete information in the form of constraints set should be generated carefully, in order to outperform the standard clustering algorithm, which works on the data source without any constraints. For sufficiently large constraints sets, the constrained clustering algorithm outperformed the MSC algorithm. The novelty of research presented here is the study of effectiveness of constraints sets and robustness of the constrained clustering algorithm using multiple sources of biological data, and incorporating biomedical text literature into constrained clustering algorithm in form of constraints sets.**

## I. Introduction

Clustering algorithms such as hierarchical clustering [1], K-means [2], and model-based clustering [3] are widely used in the analysis of gene expression data [4]. New algorithms for microarray data analysis have been incorporated into programs such as CLICK [5], EXCAVATOR [6], PIMM [7], and many more. However, the quality of clusters varies greatly, as is their ability to lead to biologically meaningful conclusions.

Some of the earliest genome-scale analysis methods for microarray data involved clustering or linear decomposition of gene expression profiles to obtain clusters of co-expressed genes, many of which were shown to be biologically meaningful [1], [8], [9], [10]. Others used these clusters for functional annotation of genes (for example, see [11]). These approaches often merely rediscovered known associations and typically did not take advantage of a vast amount of prior knowledge [12]. Much of this knowledge is buried in publications available from the biomedical literature databases. Many researchers have considered the problem of automatically extracting knowledge from the text literature and have applied it successfully to many interesting problems; see survey [13]. As shown in several papers, article abstracts can be used to successfully predict gene function [14], [15], [16] and to get functionally related gene clusters [17].

Many other sources of data are also likely to be of great assistance in the analysis of gene expression data. Such sources include protein-protein interaction data, transcription factor and regulatory elements data, comparative genomics data, protein expression data, and much more. These data provide us with a means to begin elucidating the large-scale modular organization of the cell. Conclusions drawn from more than one data source is likely to lead to improved insights. However, modifying existing exploratory analytical techniques to deal with multiple information sources is a major challenge. Efforts have been made to address this challenge. Examples include Signature algorithm [18], Multi-Source Clustering (MSC) algorithm [19], EXCAVATOR algorithm [6], IC-Clustering method [20], and many more.

The main problem with multi-source clustering is that they need "complete" data sets. In other words, every source of information must be available on every gene or protein (when a small number is missing, the values are imputed to make the dataset "complete".). However, data from sources may be incomplete, and in fact may be on a very small and select set of genes. Data resulting from experiments that are not high throughput are often not genome-wide. Such information, even if it is validated by biological experimentation, is useless for these methods. We consider the situation where partial information is presented in the form of constraints.

Recent work from the machine learning community has focused on the use of prior information in the form of instance-level constraints. Two types of pairwise constraints have been proposed: positive constraints, which specify that two instances must remain in the same cluster, and negative constraints, which specify that two instances must not be placed in the same cluster. While great efforts have been made to develop efficient constrained clustering algorithm variants [21], [22], [23], [24], [25], the role of constraints sets in constrained clustering algorithm has not been fully studied. Recently, Wagstaff et al., and Davidson et al. attempted to link the quality of constraints sets with clustering algorithm performance [26], [27]. Two properties of constraints set - inconsistency and incoherence - were shown to be strongly negative correlated with clustering algorithm performance.

In this paper, we investigate the problem of clustering genes using gene expression data with additional information in the form of constraints generated from potentially diverse sources of biological information[1]. In particular, we adapt a K-means based constrained clustering algorithm called MPCK-means originally developed by Bilenko et al. [25] and explore methods to automatically generate such constraints (both positive and negative) from multiple sources of biological data. We investigate the effectiveness of different constraints set and demonstrate that constrained clustering results in clusters that are more biologically meaningful than those using gene expression data alone when appropriate constraints set are used. For constraint set from class labels (yeast galactose dataset) [28], [29], the Rand index was used to assess the clusters. For constraints sets from biological literature (Spellman yeast dataset) [30], corrected mutual information (an improvement over the z-score measure [31]) based on gene ontology information was used.

The main contributions of this paper are (a) demonstrating how to automatically extract useful knowledge in the form of constraints from biological data sources such as text literature data, (b) showing how to use these constraints in a constrained clustering algorithm, (c) resulting in a clustering algorithm that uses incomplete and heterogeneous sources of data, and (d) studying the effectiveness of constraints sets and the robustness of the clustering algorithm to noisy constraints sets.

The rest of the paper is organized as follows. In Section 2, we describe adaptive MPCK-means algorithm briefly and compare it to standard approaches. Section 3 introduces the data sources used in this paper, investigate the effectiveness of constraints sets and discuss ways to derive constraint pairs from prior biological knowledge. In Section 4, we present the experimental results. We conclude with some discussions in Section 5.

## II. Methods

In this section we provide some background on the K-means algorithm, discuss constraints and then discuss the adaptive MPCK-means algorithm briefly.

### A. K-means Clustering

K-means is a clustering algorithm to partition the input data set into $k$ (user-specified constant) groups. It attempts to minimize the vector quantization error ($VQE$):

$$VQE = \frac{1}{2}\sum_{j=1}^{k}\sum_{x_i \in Q_j}(C_j - x_i)^2 \tag{1}$$

K-means is an iterative algorithm, with each iteration consisting of two steps. Given a set of cluster centers, the first step minimizes the error by assigning the instances to their closest centers. The second step finds new cluster centers that minimize the distortion. This can be analytically solved by taking the first order partial derivative of Eq. (1) with respect to the $j^{th}$ centroid $C_j$ and setting it to zero. Its solution gives the center up-date rule, which sets the new center to be the mean of the data points in that cluster.

Iterating through these two steps decreases the distortion monotonically and the algorithm con-verges when there is no further change in assign-ment of instances to clusters. In this paper, our goal is to adapt a generalized constrained version of the K-means algorithm and to apply it to con-straints derived from prior biological knowledge or literature data.

### B. The Type of Constraints

We assume that the input consists of two types of pairwise constraints: (1) positive constraints, which specify that two genes must lie in the same cluster, and (2) negative constraints, which specify that two genes must not be placed in the same cluster. Furthermore, all constraints may be provided with a confidence measure. Positive constraints define a transitive binary relation over the instances; a transitive closure over the input constraints is computed and then presented to our modified algorithm. In general, constraints may be derived from any given data source. In Section 3, we discuss from what data sources these constraints can be generated and how.

### C. The Adaptive MPCK-means Algorithm

The MPCK-means is a K-means algorithm that integrates constraints and metric learning. This semi-supervised clustering algorithm has shown to have better performance than constraint-based learning methods and metric-based learning methods [25]. MPCK-means has the following objective function.

$$GCVQE = \frac{1}{2}\sum_{j=1}^{k} VQE_{A_j} + PM + PC \tag{2}$$

where

$$PM = \sum_{(x_i, x_j \in M)} p_{ij}^{m} \neg\Delta(y(x_i), y(x_j)),$$

and

$$PC = \sum_{(x_i, x_j \in C)} p_{ij}^{c} \Delta(y(x_i), y(x_j)).$$

Here $VQE_{A_j}$ is the Euclidean distance function parameterized by a symmetric positive-definite matrix $A_j$ that accommodates the constraints. $p_{ij}^m$ and $p_{ij}^c$ are penalty parameters for the positive and negative constraints respectively, and $y(x_i)$ returns the index of the cluster to which $x_i$ belongs. $\Delta$ is the Kronecker delta function defined by: $\Delta(x, y) = 1$, if $x = y$, and 0 otherwise, and $\neg\Delta$ denotes its negation.

There are two extreme cases for choosing penalty parameters. They could all be set to zero; the algorithm then corresponds to the standard K-means algorithm where the constraint information is used in the initialization step. Alternatively, they could all be set to positive infinity; the algorithm then

[1]We note that the prior knowledge we explore is different from Bayesian priors, which typically specify probability distributions over the possible explanations/models for computing the posterior distribution. In contrast, prior knowledge can be used for a variety of purposes such as enforcing a particular property on the model, modifying the objective criteria and optimization procedure, and learning a new data representation.

corresponds to the version of constrained K-means algorithm that disallows violation of any constraints [21] (also adopted by EXCAVATOR [6]). The Signature algorithm explores a similar strategy by selecting a set of functionally related genes and refining them without further enforcing any penalty [18]. The penalty function is based on the data items that cause violations of constraints, and is similar to that used in earlier work [25]. The penalty for a violated positive constraint is equal to the square of the distance between the two items involved. For a violated negative constraint, the penalty is the difference between the square of the distance between the maximally separated data items in that cluster and the square of the distance between the two data items themselves. The penalty function is shown in Eq. (3) & Eq. (4).

$$p_{ij}^{m} = w_{ij}(x_i - x_j)^2 \tag{3}$$

$$p_{ij}^{c} = w_{ij}((x' - x'')^2 - (x_i - x_j)^2) \tag{4}$$

Note that if a high-confidence positive constraint pair consists of instances that are not very close, then higher penalties are imposed, which will tend to eventually tend to bring them into the same cluster. The proposed penalty function has the new feature that a confidence measure ($w_{ij}$) can assigned to each constraint. For our experiments, we set the $w_{ij}$ values equal to 1.

Like K-means, the adaptive MPCK-means algorithm is an iterative algorithm, with each iteration consisting of two steps. The first step seeks to minimize the generalized constrained vector quantization error (Eq. (2)). This is achieved by assigning instances so as to minimize the proposed error term. For pairs of instances in the constraint set, the $GCVQE$ is calculated for each possible combination of cluster assignments, and the instances are assigned to the clusters so that $GCVQE$ is minimized. The second step is to update the cluster centroids. As in K-means, the first order derivative of $GCVQE$ is set to zero and solved. Note that for our choice of penalty function, the centroid update is the same as that in the standard K-means algorithm and is the mean of the instances associated with that cluster.

## III. Data sources and Constraints Generation

To experiment with the adaptive MPCK-means algorithm, we chose to analyze the yeast gene expression datasets available from public databases. There is an enormous volume of biomedical literature containing useful knowledge that can be utilized to improve the analysis of the gene expression data [19], [32], [33]. Additional sources of biological data include genome sequence data, transcription factor regulation data, protein-protein interaction database, protein expression data, and much more. In what follows we will specify the data sources used in this work and discuss different ways of deriving constraints sets.

### A. The Data Sources

Two gene expression data sets were used in this research. The first one is the yeast galactose dataset [28]. From that study Yeung et al. compiled a subset of 205 genes reflecting four functional categories [29]. In the original experiment, microarrays were used to measure the mRNA expression profiles of yeast growing under 20 differ perturbations to the GAL pathway. Four replicates were performed for each condition.

The second gene expression data set used was generated from cultures synchronized in cell cycle by four independent methods and consisted of measurements of 6206 genes over 77 experimental conditions [30]. Standard R routines (based on the K nearest neighbor method) were used to impute missing values [34] with default parameters (15 nearest neighbors). For each gene, normalization was achieved by subtracting the median expression value and then dividing by the standard deviation. The normalized expression data was then used for the clustering.

For text information, 31924 yeast-related MEDLINE abstracts were downloaded using Entrez/Pubmed search engine based on text matching [35]. The relationship between the abstracts and the genes were represented by a "abstract-gene" relation and was constructed from the curated literature references available from the Saccharomyces Genome Database (SGD) [ftp://genome-ftp.stanford.edu/pub/yeast/data_download/literature_curation/], which provides a list of genes associated with each abstract [36]. After removing genes having no literature references, the remaining 5473 genes were retained for further analysis. It is worth mentioning that the above filtering process is unnecessary since the adaptive MPCK-means algorithm does not require all the data sources to contain the same set of genes. It only requires a single (complete) data set (in this case, gene expression data). However, this filtering step was retained since it allowed us to compare in a fair manner the results of our new algorithm and the previous Multi-Source Clustering algorithm [19].

### B. Constraints Generation

As mentioned earlier, the yeast galactose dataset contains a list of genes with known functional categories. For any chosen pair of genes, if they were from the same functional category, then a positive constraint can be generated. Otherwise a negative constraint can be generated. For our experiments, constraints were chosen by randomly choosing sets of pairs of genes.

For the yeast cell cycle data set, constraints were generated from text literature data. A naive approach is as follows. Generate positive constraint pairs from genes associated with the same document. After merging all positive constraint pairs into connected components, neighborhood sets could be produced and genes belonging to different neighborhood sets could be used to build the negative constraint set. It is clear that this approach ignores contextual information and natural language issues and could potentially generate erroneous constraints. Although this approach is efficient, we used the "gene-term" matrix as described in an earlier study of [19].

We explored two ways to generate gene pair constraints from the gene-term matrix. We refer to the two approaches as *similarity-based* and *cluster-based* constraint generation methods. In the first approach, cosine similarity was calculated between all pairs of genes. Positive constraint pairs were generated by picking gene pairs having high similarity. Negative constraint pairs may be generated by picking gene pairs with low similarity values. However, for the text literature data source, this is not appropriate since it is difficult to differentiate between "negative information" (i.e., two genes should not be in the same cluster) and "lack of information". For our experiments with this data set, we did not use any negative constraints. In the second approach, the spherical K-means algorithm, which is a K-means algorithm using cosine-based distance, was applied to the gene-term matrix T times (we used T=100). Frequencies of gene pairs appearing in the same cluster were counted and sorted into T + 1 bins using the bucket sort algorithm. Gene pairs with high frequency were listed as positive constraint pairs. In other words, the constraints can be generated from the consensus matrix $M$, where $M_{ij}$ gives the probability that the K-means algorithm placed gene $i$ and $j$ in the same cluster. Two sets of constraint pairs were generated: constraint pairs from text similarity, and constraint pairs from text clustering. We selected 20,000 pairs for each set and the two resulting sets were used as the constraint pool for our experiments.

## C. Evaluation Methods

To evaluate clustering, two different measures were used. Evaluation was performed using the $Rand\ index$ [37] for the first data set (with known labels). Rand index allows for a measure of agreement between the clustering results and their true labels. Let $n$ be the size of data set $D$. The clustering result is viewed as a collection of $n \times (n-1)/2$ pairwise decisions. For each pair of items $d_i$ and $d_j$ in $D$, the algorithm either assigns them to the same cluster or to different clusters. Let $p_1$ ($p_2$, respectively) be the number of decisions where the algorithm correctly assigns the same (different, respectively) label to items $d_i$ and $d_j$. Then

$$RandIndex = \frac{p_1 + p_2}{n \times (n-1)/2}$$

For the results from the second data set (without known labels), Rand index cannot be used. The clusters were assessed using the knowledge from the Gene Ontology (GO) database [http://www.geneontology.org]. The GO database provides a controlled vocabulary to describe genes and gene product attributes in different organisms. It thus provides a list of genes that can be associated with a specific term in the vocabulary. It represents all GO terms as a directed acyclic graph [38]. The SGD gene association file was downloaded directly from the Gene Ontology website (Revision: 1.1230). A table of 6470 genes and 4032 GO terms was produced in which a 1 in position (i,j) meant that gene i is associated with GO term j, and a 0 indicates a lack of knowledge about their association. Each GO term is seen as an attribute and the gene-attribute table is the knowledgebase used to assess the quality of gene clusters. A figure of merit called corrected mutual information ($CMI$) was used to measure the quality of cluster. The definition of the $CMI$ measure is shown below.

$$CMI = MI_{Cluster} - \overline{MI}_{random},$$

where

$$\overline{MI}_{random} = \sum_{j=1}^{m} MI_{random(j)}/m,$$

and

$$MI_{Cluster} = MI(C, A_1, \ldots, A_{N_A}) = \sum_i MI(C, A_i)$$

$$= N_A H(C) + \sum_i H(A_i, C).$$

Here $MI_{Cluster}$ corresponds to the mutual information between the cluster assignment produced by a clustering algorithm and the GO database; $MI_{random}$ refers to the mutual information between a random cluster assignment and the GO database. $MI(C, A_1, \ldots, A_{N_A})$, the mutual information between a cluster assignment ($C$) and all attributes (GO terms, $A_1, \ldots, A_{N_A}$), is defined as the sum of the mutual information between the cluster assignment and the genes associated with each individual attribute, using the standard definition of mutual information. More details refers to [31]. Thus, higher $CMI$ values suggest that the clustering results are more significantly related to gene function. Our GO-based $CMI$ measure is similar to the z-score measure [31]. The $CMI$ measure was chosen over the z-score because the z-score were erratic and unstable[2].

## IV. Experimental Results

The motivation for designing the constrained clustering algorithm is based on the hypothesis that constraints based on prior correct knowledge can assist a clustering algorithm to improve its performance. Our experimental results confirm the hypothesis, but also show situations where it is not true.

## A. Effectiveness of Constraints Set

We used the yeast galactose data set (with all replicates) to investigate the effectiveness of constraints sets. The first observation was the performance of clustering with randomly generated constraints can be deteriorated even if the constraints conformed to the known functional categories (i.e., labels). This was observed earlier in a different context [26]. In fact, many of the errors in constrained clustering had been correctly classified when no constraints were provided.

As shown in Figure 1, constraints sets with small size can be detrimental to performance of the algorithm. However, when

[2]The z-score is implemented as a web service called ClusterJudge [http://llama.med.harvard.edu/cgi/ClusterJudge/cluster_judge.pl]. However different runs of ClusterJudge produce very different values for the same input and were thus not used for this work. This can easily be seen by comparing our graphs in Figure 3 (using $CMI$ measure) with those obtained by the z-score measure (see Figure 1 in (Yang, et al., 2005); also see Figure 3 in (Gibbons and Roth, 2002)).

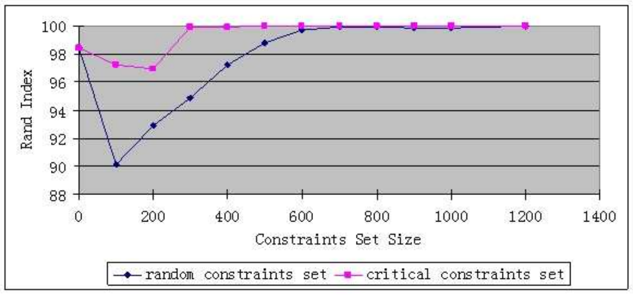

Fig. 1. Performance of constrained clustering algorithm when using different constraints set size and different types of constraints sets. The experiments involved computed the average Rand index on 10 runs for each constraints set size.

the size of the constraints set is sufficiently large, then accurate clustering is achieved with both types of constraints sets. Therefore, we investigated the alternative option of generating constraints from relationships which the clustering algorithm could not learn when no constraints were provided. A second constraints set was generated by the following procedure. First a clustering was obtained by running standard clustering algorithm (without any constraint); then constraint pairs were chosen at random from all pairs of genes that included at least one misclassified gene. The pairs were then placed in the positive constraints set or negative constraints set using the known labels. We refer to the resulting set of constraints as the critical constraints set.

Figure 1 shows that accurate clustering can be achieved with much smaller critical constraints sets (roughly, 300 is enough in comparison to the roughly 700 that is needed in the case of random constraints sets). In practice, it is very difficult to estimate the size of the critical constraints set required to achieve accurate clustering.

Table I & Table II show more detailed information on these experiments. The columns include the following quantities on 10 runs: minimum Rand index (Min), maximum Rand index (Max), Average Rand index (Ave), standard deviation of Rand index (SD), and the number of runs on which the Rand index was higher than the average Rand index of the standard K-means algorithm which is used as the baseline (R). Clearly, the algorithm with the critical constraints set outperformed the version with a random set. The last column also indicates that a great percentage of the runs actually resulted in a better performance than the "baseline".

### B. Results with Inconsistent Constraints

The results shown in Section 4.1 showed that small constraints set may hurt the ability to infer accurate clusters. However, all constraints used in Section 4.2 were known to be true and were based on known functional categories. Next we investigated how "robust" the algorithm was, i.e., how does it perform when a small number of constraints are incorrect or inconsistent with known functional categories. Once again, we consider two ways of generating constraints sets. As before pairs were either picked at random or picked from among pairs

TABLE I
CLUSTERING PERFORMANCE (RAND INDEX) WITH RANDOM CONSTRAINTS SET, THE COLUMNS INCLUDE THE QUANTITIES ON 10 RUNS.

| Constraints Set Size | Min | Max | Ave | SD | R[a] |
|---|---|---|---|---|---|
| 0 | 98.46 | 98.47 | 98.469 | 0.0032 | 0 |
| 100 | 85.86 | 98.48 | 90.158 | 4.5926 | 1 |
| 200 | 88.85 | 100 | 92.897 | 3.46083 | 1 |
| 300 | 86.09 | 100 | 94.882 | 4.5318 | 3 |
| 400 | 93.14 | 99.49 | 97.233 | 1.9473 | 3 |
| 500 | 96.01 | 100 | 98.768 | 1.379 | 7 |
| 600 | 98.24 | 100 | 99.709 | 0.5566 | 9 |
| 700 | 99.49 | 100 | 99.897 | 0.1685 | 10 |
| 800 | 99.74 | 100 | 99.948 | 0.0909 | 10 |
| 900 | 99.52 | 100 | 99.874 | 0.1516 | 10 |
| 1000 | 99.52 | 100 | 99.889 | 0.1757 | 10 |
| 1200 | 99.87 | 100 | 99.98 | 0.0445 | 10 |

[a] the number of runs on which the Rand index was higher than the average Rand index of the standard K-means algorithm (used as the baseline)

TABLE II
CLUSTERING PERFORMANCE (RAND INDEX) WITH CRITICAL CONSTRAINTS SET, THE COLUMNS INCLUDE THE QUANTITIES ON 10 RUNS.

| Constraints Set Size | Min | Max | Ave | SD | R[a] |
|---|---|---|---|---|---|
| 0 | 98.46 | 98.47 | 98.469 | 0.0032 | 0 |
| 100 | 85.64 | 100 | 97.219 | 5.0443 | 8 |
| 200 | 82.71 | 100 | 96.925 | 6.1087 | 8 |
| 300 | 99.49 | 100 | 99.898 | 0.215 | 10 |
| 400 | 99.49 | 100 | 99.898 | 0.215 | 10 |
| 500 | 100 | 100 | 100 | 0 | 10 |
| 600 | 100 | 100 | 100 | 0 | 10 |
| 700 | 100 | 100 | 100 | 0 | 10 |
| 800 | 100 | 100 | 100 | 0 | 10 |
| 900 | 100 | 100 | 100 | 0 | 10 |
| 1000 | 100 | 100 | 100 | 0 | 10 |
| 1200 | 100 | 100 | 100 | 0 | 10 |

[a] the number of runs on which the Rand index was higher than the average Rand index of the standard K-means algorithm (used as the baseline)

with at least one misclassified gene. A small number of the generated constraints were then made inconsistent (i.e., put into positive constraints when they should have been put into negative constraints, vice versa).

Figure 2a and Figure 2b show that a small number of inconsistent constraints does not affect clustering algorithm performance a lot when both consistent and inconsistent pairs of constraints were generated from the $critical constraints set$. Otherwise, the performance of the algorithm decreases dramatically. However, Figure 2c shows that algorithm performance decreases considerably if the fraction of inconsistent constraints rises above some threshold even when the consistent and inconsistent pairs of constraints were generated from $critical constraints set$. Overall, the results confirm the hypothesis that inconsistent constraints will diminish clustering performance and should be avoided.

### C. Results with Constraints from Text Data

For the yeast cell cycle data set, since the functional categories are not available, the constraints set cannot be generated as described before. But the constraints set can be generated from other knowledgebases such as biological text literature.

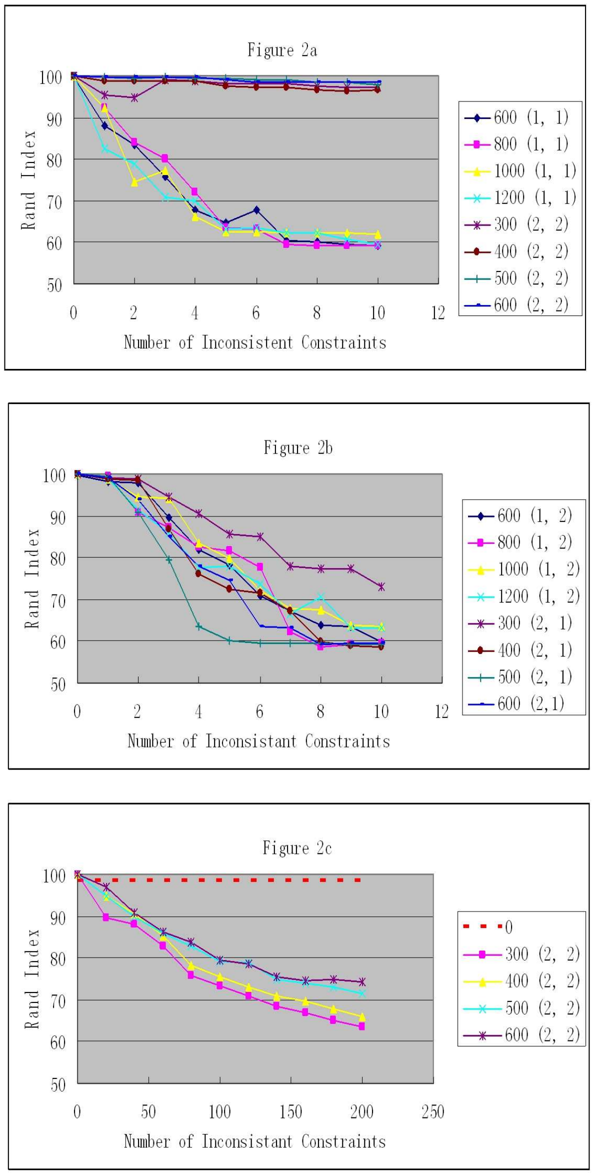

Fig. 2. Performance of constrained clustering algorithm when using inconsistent constraints. $N(m_1, m_2)$ means that experiment was performed for constraint set size $N$ generated by approach $m_1$, and the inconsistent constraints pair were from set $m_2$. $m_1 = 1$ means that constraint set was randomly generated, $m_1 = 2$ means that constraint set was critical constraint set. $m_2 = 1$ means inconsistent constraints pair were from randomly generated constraint set, $m_2 = 2$ means inconsistent constraints pair were from critical constraint set.

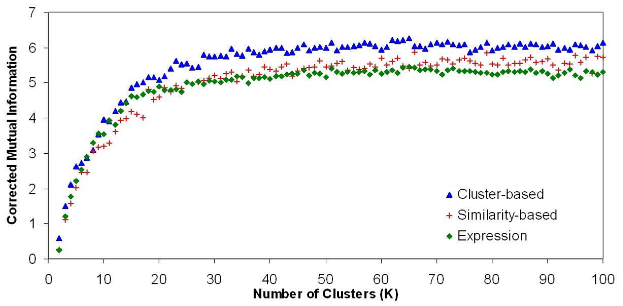

Fig. 3. Clustering results from clustering gene expression data alone along with constrained clustering using positive constraints from both cluster-based and similarity-based methods. The horizontal axis shows the number of clusters desired, and the vertical axis shows corrected mutual information measure.

We discussed two approaches to generate constraints sets from biological text literature: cluster-based and similarity-based. In this subsection, we compare the effectiveness of constraint generation by the two methods. 1000 pairs of constraints were randomly picked from the positive constraint set. These were then provided as inputs to the adaptive MPCK-means algorithm. The expression data consisted of 5473 yeast genes under 77 experimental conditions.

The experiments were performed in 10 runs for each parameter setting (same k, same constraint type, if any) and the averaged $CMI$ was plotted against the number of clusters, k, for all values of k from 2 to 100.

As mentioned earlier, no negative constraints were used in this set of experiments. Figure 3 shows the $CMI$ measures results using three approaches. All three used gene expression data and differed in the constraints generation method used. The first performed clustering using no constraints. The second performed clustering using constraints generated by the cluster-based method while the third used constraints generated by the similarity-based approach. The version with constraints performed better than the one with no constraints. The constraints generated from the cluster-based method performed better than those from the similarity-based approach.

## *D. Results Comparison with MSC*

Finally we compare the results of our experiments with adaptive MPCK-means to that of the MSC algorithm [19], which currently has the best performance among the algorithms that analyze gene expression data with text literature data. As in section 4.3, k was set to 50. In all the previous experiments, only 1000 constraints were used. Here different constraint sizes were tried to investigate the effect of the size of the constraint on the overall clustering performance. Since the total size of constraint pool was 20,000, constraint set sizes were selected starting from 2000 with a step of 2000 along with a constraint set of size of 1000, resulting in 11 different constraint set sizes for each type of constraint set.

Figure 4 presents the $CMI$ measures results for constrained clusters obtained by cluster-based positive, and similarity-based positive constraints and MSC results in dashed line. As constraint set size increases, the $CMI$ measures for each of three types of constraints steadily improved and flattened out after 16000. MSC performed better than the cluster-based adaptive MPCK-means version when constraint set size was less than 10000, and was outperformed for larger set sizes. However, MSC performed significantly better when constraints were generated by text similarity.

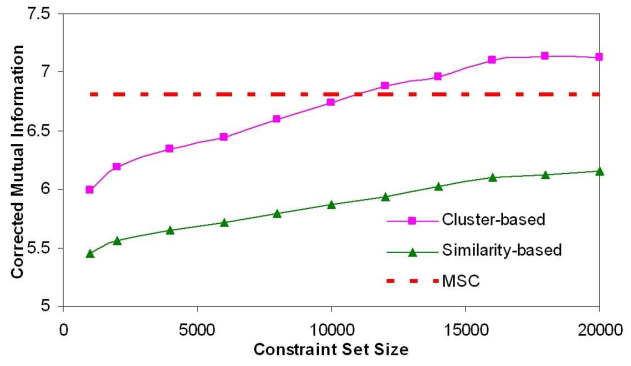

Fig. 4. Comparison of clustering generation methods as a function of constraint set size.

## V. Discussion and Conclusions

We have adapted MPCK-means algorithm to gene expression data with constraints derived from prior known functional categories and from a subset of the literature data related to the Saccharomyces genome.

We concluded that constraints set may not improve the performance of a clustering algorithm if it is not carefully chosen or not large enough. We also made conclusion that inconsistent constraints should be avoid since small number of them will hurt the clustering. We explored several ways of deriving constraints from prior knowledge about the data and from external biological text literature. Our results suggested that clustering using constraints generated from a cluster-based approach outperformed the case when constraints were generated using a similarity-based approach. Comparisons with MSC, another multi-source clustering algorithm that relies on complete data sets, showed that constraint-based methods performed better assuming that the constraints were of sufficiently high quality. Based on our experience, negative constraints are not appropriate for text literature data source. However, we conjecture that it is appropriate for other data sources such as protein-protein interactions, and to a limited extent, regulatory element information.

Several open questions remain unanswered. First, how does the adaptive MPCK-means algorithm perform when more sources of biological data are included? Second, since there is no need to have every gene put into some cluster, how should the objective function be revised to account for a "noise" cluster? Third, is there any theoretic reason to explain why some constraints set decrease the performance of a clustering algorithm? Fourth, is it possible to generate high quality constraints? Finally, can all clustering algorithms, with MSC in particular, be adapted to utilize constraints profitably?

## SUPPLEMENTAL WEBSITE

http://biorg.cs.fiu.edu/GCC

## Acknowledgments

GN was supported by NIH Grant P01 DA15027-01 and NIH/NIGMS S06 GM008205, TL was supported by NSF CAREER Award IIS-0546280, and IBM Faculty Award, and EZ was supported by FIU Presidential Fellowship.

## References

[1] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc Natl Acad Sci U S A*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[2] R. Herwig, A. Poustka, C. Muller, C. Bull, H. Lehrach, and J. O'Brien, "Large-scale clustering of cdna-fingerprinting data," *Genome Research*, vol. 9, no. 11, pp. 1093–1105, 1999.

[3] K. Yeung, C. Fraley, A. Murua, A. Raftery, and W. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, pp. 977–987, 2001.

[4] J. Quackenbush, "Computational analysis of microarray data," *Nat Rev Genet*, vol. 2, pp. 418–427.

[5] R. Shamir and R. Sharan, "Click: A clustering algorithm for gene expression analysis," in *Proc Int Conf Intell Syst Mol Biol*, vol. 8, 2000, pp. 307–316.

[6] D. Xu, V. Olman, L. Wang, and Y. Xu, "Excavator: a computer program for efficiently mining gene expression data," *Nucleic Acids Res.*, vol. 31, no. 19, pp. 5582–5589, 2003.

[7] M. Medvedovic, K. Yeung, and R. Bumgarner, "Bayesian mixture model based clustering of replicated microarray data," *Bioinformatics*, vol. 20, no. 8, pp. 1222–1232, 2004.

[8] J. DeRisi, V. Iyer, and P. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale." *Science*, vol. 278, no. 5338, pp. 680–686, 1997.

[9] S. Raychaudhuri, J. M. Stuart, X. Liu, P. M. Small, and R. B. Altman, "Pattern recognition of genomic features with microarrays: Site typing of mycobacterium tuberculosis strains," in *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 2000, pp. 286–295.

[10] G. Sherlock, "Analysis of large-scale gene expression data," *Brief Bioinform*, vol. 2, no. 4, pp. 350–362, 2001.

[11] M. P. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *PNAS*, vol. 97, no. 1, pp. 262–267, 2000.

[12] R. Altman and S. Raychaudhuri, "Whole-genome expression analysis: challenges beyond clustering," *Current Opinion in Structural Biology*, vol. 11, no. 3, pp. 340–347, 2001.

[13] M. Yandell and W. Majoros, "Genomics and natural language processing," *Nature Reviews Genetics*, vol. 3, pp. 601–610, 2002.

[14] W. Fleischmann, S. Möller, A. Gateau, and R. Apweiler, "A novel method for automatic and reliable functional annotation of proteins," in *German Conference on Bioinformatics*, 1998.

[15] S. Raychaudhuri, J. Chang, P. Sutphin, and R. Altman, "Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature," *Genome Res*, vol. 12, pp. 203–214, 2002.

[16] J. Tamames, C. A. Ouzounis, G. Casari, C. Sander, and A. Valencia, "Euclid: automatic classification of proteins in functional classes by their database annotations," *Bioinformatics*, vol. 14, no. 6, pp. 542–543, 1998.

[17] D. Chaussabel and A. Sher, "Mining microarray expression data by literature profiling," *Genome Biology*, vol. 3, no. 10, p. RESEARCH0055, 2002.

[18] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing modular organization in the yeast transcriptional network," *Nature Genetics*, no. 31, pp. 370 – 377, 2002.

[19] C. Yang, E. Zeng, T. Li, and G. Narasimhan, "Clustering genes using gene expression and text literature data," in *CSB '05: Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 329–340.

[20] J. Sese, Y. Kurokawa, M. Monden, K. Kato, and S. Morishita, "Constrained clusters of gene expression profiles with pathological features," *Bioinformatics*, vol. 20, no. 17, pp. 3137–3145, 2004.

[21] K. Wagsta, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proceedings of 18th International Conference on Machine Learning (ICML-01)*, 2001, pp. 577–584.

[22] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of 19th International Conference on Machine Learning (ICML-02)*, 2002, pp. 19–26.

[23] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering." in *ICML*, 2002, pp. 307–314.

[24] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Advances in Neural Information Processing Systems*, vol. 15. MIT Press, 2003.

[25] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM Press, 2004, p. 11.

[26] I. Davidson, K. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms." in *PKDD*, ser. Lecture Notes in Computer Science, vol. 4213. Springer, 2006, pp. 115–126.

[27] K. Wagstaff, S. Basu, and I. Davidson, "When is constrained clustering beneficial, and why?" in *AAAI*, 2006.

[28] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood, "Integrated genomic and proteomic analyses of a systematically perturbed metabolic network." *Science*, vol. 292, no. 5518, pp. 929–934, 2001.

[29] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner, "Clustering gene-expression data with repeated measurements," *Genome Biol*, vol. 4, p. R34, 2003.

[30] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization." *Mol Biol Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[31] F. Gibbons and F. Roth, "Judging the quality of gene expression-based clustering methods using gene annotation," *Genome Research*, vol. 12, no. 10, pp. 574–581, 2002.

[32] P. Glenisson, J. Mathys, and B. D. Moor, "Meta-clustering of gene expression data and literature-based information," *SIGKDD Explor. Newsl.*, vol. 5, no. 2, pp. 101–112, 2003.

[33] D. Masys, "Linking microarray data to the literature," *Nat Genet*, vol. 28, pp. 9–10, 2001.

[34] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for dna microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[35] R. J. Roberts, "Pubmed central: The genbank of the published literature," *PNAS*, vol. 98, pp. 381–382, 2001.

[36] E. Hong, R. Balakrishnan, K. Christie, M. Costanzo, S. Dwight, S. Engel, D. Fisk, J. Hirschman, M. Livstone, R. Nash, R. Oughtred, J. Park, M. Skrzypek, B. Starr, R. Andrada, G. Binkley, Q. Dong, B. Hitz, S. Miyasato, M. Schroeder, S. Weng, E. Wong, K. Zhu, K. Dolinski, D. Botstein, and J. Cherry, "Saccharomyces genome database, http://www.yeastgenome.org/."

[37] W. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[38] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. the gene ontology consortium." *Nat Genet*, vol. 25, no. 1, pp. 25–9, 2000.