Prioritization of Clusters for Post-genomic Analysis

Malay Bhattacharyya\textsuperscript{a}, Sanghamitra Bandyopadhyay\textsuperscript{b}

\textsuperscript{a}Department of Computer Science & Engineering, University of Kalyani, Nadia – 741235, India. E-mail: malaybhattacharyya@klyuniv.ac.in.
\textsuperscript{b}Machine Intelligence Unit, Indian Statistical Institute, Kolkata – 700108, India. E-mail: sanghami@isical.ac.in.

Abstract

Clustering algorithms are occasionally used on biological datasets to obtain statistically coherent groups of biomolecules. The biological coherence of such a group (e.g., a gene or protein cluster) might have different interpretations. Functional similarity is the most widely used form of biological coherence of a cluster. We often require to assign priorities, which would signify the betterness of biological coherence, to such clusters for post-genomic analysis. In this paper, we propose a novel approach of prioritizing gene clusters. We introduce a new measure of compactness for quantifying the coherence of the clusters based on their strength of associativity. Employing this, a post processing subroutine is introduced to fit with the standard clustering algorithms to obtain a set of improved prioritized clusters. We test the results upon applying the method on several microarray datasets. Statistical and biological studies on the derived clusters depict the effectiveness of the proposed methodology in the better selection of functionally enriched gene clusters. The proposed methodology helps to select significant clusters and also filter out the noisy ones.

1. Introduction

Clustering of a set of objects brings together the objects having similar pattern and separates those having dissimilar pattern [16]. Given a set of data points $D \in \mathbb{R}^n$, formally, a $k$-clustering solution of the points in $D$ is defined to be a set of clusters $\{D_1, D_2, \ldots, D_k\}$ such that (i) $D_i \neq \phi, \forall i \in \{1, 2, \ldots, k\}$, (ii) $\bigcup_{i=1}^{k} D_i = D$ and (iii) $D_i \cap D_j = \phi, i \neq j, \forall i, j \in \{1, 2, \ldots, k\}$. Conventionally, this formalism is defined for a static clustering scenario. Some real-life applications like social network analysis [10], web mining [7], gene expression study [26], etc. demand the inclusion of dynamics in clustering. While clustering a set of objects, the concept of dynamism becomes applicable in two ways – either the set of objects is dynamic [1] or the approach of clustering is itself dynamic [19]. For several applications, we require to prioritize (rank) the clusters obtained from both static and dynamic clustering. One such example is
the post-genomic analysis, where we have the need of assigning priorities to clusters. This helps to recognize the important clusters (of genes or proteins) for further analysis, and as well as reject the irrelevant ones.

Genes are molecules of an organism that participate in various biological activities. Study of biomolecules like genes, proteins or microRNAs can elucidate many functional principles at the system level of an organism [23]. Targeting this, various high-throughput technologies have evolved over time in the recent decades. This has enabled the analysis of a large amount of data for extracting significant information. Microarray analysis is one of such tools that helps to measure the expression levels (amount of mRNAs generated) of thousands of genes in parallel [27]. Such experiments are generally conducted on multiple gene probes over certain conditions (time points, tissue types, etc.). Generally, genes are clustered based on the resemblance in their expressibility/repressibility patterns reflected in the microarray analysis. A cluster of genes are therefore statistically coherent in nature. The genes in the same clusters are anticipated to have similar functional activity or functional cohesiveness. As a matter of fact, the formation of clusters through a clustering solution is merely the partitioning of the data space. But it does not provide any information about the strength of the partitions, thereby quantifying the cluster priorities.

In this paper, we propose a concept of prioritization aiming the post-clustering gene analysis. We introduce a post processing subroutine to be integrated with any clustering algorithm for obtaining a set of prioritized clusters. A higher priority assigned to a cluster of genes denotes a superior biological cohesiveness. Due to the scale-free nature of the gene interaction networks, a large number of genes falls under the irrelevant (and tiny) clusters [6]. Managing this large set of outliers is an important concern in post-genomic analysis.

The current paper is arranged as follows. Section 2 includes some motivating examples that inspires the current work. The related state-of-the-art is described in section 3. The proposed methodology is provided in the sections 4 and 5, with complexity analysis in section 6. Section 7 provides the empirical study, and finally section 8 concludes the paper.

2. Motivation

In this section, we highlight several example scenarios (Fig. 1) to illustrate how tiny clusters and outliers might affect the clustering solution. Consider the orientation of some data points as shown in Fig. 1(a). A 2-clustering solution provided by any conventional clustering algorithm on the data points will segregate \{1, 2, 3, 4, 5, 6, 7, 8, 9\} in one cluster and \{10\} in the other. The motivation with which such a clustering solution is achieved is minimizing the intra-cluster distance and, as well, maximizing the inter-cluster variance. With this motivation, it may so happen that the data points are clustered into unbalanced groups, especially when the number of clusters is hard to estimate. This is what causes the generation of tiny clusters (say, the clusters of size less than three), when the data points are scattered. Unfortunately in many applications, like post-genomic analysis, the number of clusters is unknown and the tiny clusters are considered as irrelevant. Additionally, the presence of tiny clusters obstructs the separation of the compact clusters (e.g., the sets of data points \{1, 2, 3, 4\} and \{5, 6, 7, 8, 9\} in Fig. 1(a)). Suppose, we include some additional data points and obtain the orientation shown in Fig. 1(b). With this, a clustering solution will assign most of the single data points into separate clusters. In that case, the separation of the close and compact clusters becomes harder. Therefore, a motivation of clustering like this might not be suitable for post-genomic analysis where tiny clusters frequently occur [6]. It is also understandable that when the data points are clustered into balanced groups no problem occurs due to the presence of tiny clusters.

Let us consider the data points shown in Fig. 1(a) from another perspective. A 2-cluster solution, that produces the clusters \{1, 2, 3, 4, 5, 6, 7, 8, 9\} and \{10\} might alter if the point 10 is an outlier. Therefore, the priorities are closely dependent on the size and compactness of a cluster in a noisy dataset. But, due to the lack of such information the aforesaid solution will be produced. In post-genomic analysis, such prior information is also unavailable. So, prioritization of clusters to decide about their utility becomes necessary for post-genomic analysis.

In presence of more outliers than the number of data points, clustering becomes a problem. Sometimes we also have to approach the clustering on datasets with tiny candidates and outliers in it. This is an evident case for gene interaction networks where many genes remain segregated from the shorter sized population of core clusters. In fact, it has already been established that various clustering algorithms addresses a choice of non-redundant conflicting objectives and it is almost impossible to unify them [18]. Therefore, the development of application oriented clustering and quantification of priorities of clusters has been in demand.
3. Related Works

Ranking of clusters obtained from a clustering solution is a recent concept. One of the very first approaches toward this is related to a content-based image retrieval application [22]. However, the basic task behind such ranking is giving priorities to a cluster, i.e. quantifying the quality of a cluster. Indeed, this is a goal of many earlier clustering studies. Unfortunately, there is no common agreement on measuring the quality of a clustering solution or of a single cluster [17]. We study some of the related works and their limitations hereunder.

The literature on clustering and community detection consists of numerous measures of quantifying the statistical quality of a group of objects. Some of them are based on distance-based metrics such as minimum diameter, sum-of-squares, k-medians and k-means [15]. Minimum cut and maximum flow is a common measure [12]. There are some others on graph-theoretic measures like normalized cut that takes care of the limitation of minimum cut measure of favoring to partition out small sets of nodes from the graph [24]. It simply normalizes the cut strength by the sizes of the two partitions produced by the cutting procedure. However, deciding the normalized cut of a graph is again an NP-complete problem [24]. In a more recent study, the limitation of minimum cut has provoked the proposal of a new measure, called conductance, that also accounts the expansion of the cut partitions generated [17]. Recently, some topology based measures like k-cliques [21] and others [14] have also emerged. Modularity is another measure of cohesiveness. Consider a particular division of a network into $k$ communities. Let us define a $k \times k$ symmetric matrix $e$. A element $e_{ij} \in e$ denotes the fraction of all those edges in the network that connect vertices in community $i$ to vertices in community $j$. The modularity of a network can be defined as given below.

$$\sum_i \left( e_{ii} - \left( \sum_j e_{ij} \right) \right)^2,$$

where $e_{ij}$ represents the fraction of all edges in the network that connect vertices in community $i$ to vertices in community $j$ [20]. Edge betweenness is sometimes used for quantifying the strength of a community [13]. The betweenness of an edge is defined as the number of shortest paths, between the vertex pairs, that includes the edge considered. Some statistical methods (e.g., see [4]) are also in literature. However, there is no single, widely acceptable, definition. Moreover, many of the above notions are application specific or suited for restricted scenarios, e.g. hard partitional clustering, clustering of metric space data points, or model-based clustering [15, 18].

A new measure for computing the strength of clusters, integrated cohesion [5], which can be applied to any arbitrary weighted network has recently been proposed. In the same study, a new cluster ranking algorithm C-Rank with applications in mining mailbox networks is also presented. It describes the limitations of edge separators and proposes the use of vertex separators for modeling cohesiveness of a cluster. In this, cohesion is defined as [5]

$$\min_{(S,S_1,S_2)} \frac{|S|}{\min{|S_1|,|S_2|} + |S|},$$

where $|S|$ represents the order (number of vertices) of the subgraph $S$, and $S_1, S_2 \subset S$ denote its partitions. However, the conception of vertex separators is also very restrictive to the structure of the graph.

Consider the two unweighted graphs shown in Fig. 2(a). For both the graphs $G_1$ and $G_2$ shown in Fig. 2(a), the value of cohesion $= 1/(4+1) = 0.2$. Unfortunately, removal of the common vertices shared between the two cliques
and two cycles in the graphs $G_1$ and $G_2$, respectively, results in the formation of a pair of subgraphs of same orders. Therefore, the cohesion values become same. However, it is evident that the graph $G_1$ is more cohesive than $G_2$. The problem is in defining the cohesiveness using the notion of separability of the subgraph itself, not considering its topological properties.

The computation of integrated cohesion for weighted graphs depends on the cohesion of their unweighted versions, which we already discussed to bear a limitation. Again, the distribution of the edge weights can result into varying integrated cohesion values. E.g., see Fig. 2(b), where two weighted graphs having different weight distributions are shown. Due to the maximality constraint in case of weighted graphs, noisy edge weights may contribute to erroneous results. There are some recent approaches that consider minimum participation of each vertex in a cluster to quantify its strength in terms of association density [2]. Extending this, a relaxed version of association density has also been proposed on fuzzy graphs [3,9].

Unfortunately, most of the existing methods are not oriented towards the application of post-genomic studies. Recently, Liang and Wang have proposed a dynamic clustering approach for expression studies without dealing with the prioritization issues [19]. However, their approach shows a solution of managing large outliers. They kept the scattered, singleton and mini-cluster genes (probable outliers) all in a separate cluster while applying the dynamic clustering. So, this might be problematic for special orientation of data points. Sun et al. have recently proposed another clustering framework that integrates ranking, namely RankClus, for the analysis of heterogeneous information networks [25]. But it has limited applicability to clustering citation information in DBLP. Moreover, the theoretical framework of RankClus is based on bipartite networks [25]. We describe an approach to manage the existing limitations of prioritizing clusters in the subsequent sections of the current paper and highlight its utilities in post-genomic analyses.

4. Theoretical Insights

In this study, our main focus is on post-genomic analysis and for this we are aiming toward clustering of genes. To account for a region in a Euclidean space we must have at least three data points. In fact, a cluster containing at most two data points is not able to encompass a boundary in the Euclidean space. We define such clusters as tiny clusters. In contrast, a cluster having at least three data points are said to be non-tiny. Based on this understanding, here we treat tiny clusters as outliers in case of the gene expression analysis. In fact, the size and compactness are the main features for quantifying the strength or priority of a cluster. We formally characterize the size and compactness of a cluster with the following terminologies.

1. **Order**: Order of a cluster defines the number of data points in the cluster.
2. **Cohesiveness**: Cohesiveness of a cluster defines the Euclidean space surrounded by the data points within the cluster.
Let us assume that $o(s_i)$ and $c(s_i)$ denote the order and cohesiveness of any arbitrary cluster $s_i$, respectively. Then, for a given a set of $n$ clusters $s = \{s_1, s_2, \ldots, s_n\}$, the objective of our clustering approach is to minimize the function

$$Z = \sum_{i=1}^{n} f(o(s_i), c(s_i)),$$

where $f(o, c)$ denotes the strength of a cluster, given its order and cohesiveness $o$ and $c$, respectively. It is easily realizable from a general intuition that it should hold

$$f(o(s_i), c(s_i)) \propto \frac{1}{o(s_i)},$$

and additionally,

$$f(o(s_i), c(s_i)) \propto o(c_i).$$

Combining (4) and (5), we obtain the relation

$$f(o(s_i), c(s_i)) = k \cdot \frac{o(c_i)}{o(s_i)},$$

where $k$ is a constant value.

We have to ensure this constraint as far as possible. It is important to note that, adopting such a criteria may generate bias towards large clusters amongst a clustering solution. The two sets of data points shown in Fig. 3 have contradictory priorities based on size and intra-cluster variance (a form of compactness). Thus, combining these two features without imparting any bias to the final result is a challenging task. We propose a novel measure of compactness towards this goal in the following section.

5. Methodology

In this section, we describe the proposed method of prioritization and its utility in further improvement of a clustering result. We assume that a clustering result is readily available on which we will be working on. We introduce a priority-based rearrangement that will make outliers separated automatically from a set of strong clusters. This is achieved by integrating the tiny clusters in a separate cluster. However, the concept of keeping outliers in a separate cluster is not a new one [19]. The entire methodology of prioritization and cluster improvement is shown in Algorithm 2.

5.1. Basic Definitions

Before going into the description of the algorithm, we first define the bonding strength of a data point in a cluster as given below. This definition is inspired from a recently proposed measure of association density of a vertex within a set of vertices (vertexlet) [2].
Definition 5.1 (Bonding strength of a data point). Given a cluster $C$, the bonding strength, $B_{d_i/C}$, of a data point $d_i \in C$ is defined as the ratio of the sum of the similarities between $d_i$ and each of the data points belonging to $C$, and the order of the cluster $C$. Thus, the bonding strength of a data point $d_i$ with respect to the cluster $C$ is computed as

$$B_{d_i/C} = \frac{\sum_{d_j \in C, d_i \neq d_j} Sim(d_i, d_j)}{\sigma(C) - 1}.$$

In Eqn. (7), $\sigma(C)$ represents the order of the cluster $C$ and $Sim(d_i, d_j)$ denotes the similarity between the data points $d_i$ and $d_j$. In fact, this is the complementary to the distance measure. We assume the similarity values are normalized within the range $[0,1]$. For a graph, $Sim(i, j)$ represents the normalized weight (denoting resemblance) associated with the edge $(i, j)$. Based on this definition, we define the bonding strength of a cluster as follows.

Definition 5.2 (Bonding strength of a cluster). The bonding strength of a cluster is defined as the weighted difference between the mean and standard deviation of all the data points belonging in, with respect to the cluster. So, the bonding strength of a cluster $C$ is given by

$$B_C = \alpha(C) \ast (\mu(B_{d_i/C}) - \sigma(B_{d_i/C})),$$

where $\mu(B_{d_i/C})$ and $\sigma(B_{d_i/C})$ denotes the mean and standard deviation of bonding strengths of all the data points in the cluster, respectively.

Evidently, higher the $B_C$ value, i.e. more the bonding strength, denotes a stronger cluster. We consider the variability of the bonding strength to prioritize the clusters. For example, the bonding strength of a cluster with bonding strength vector of the data points as $[0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5]$ is 0.027, whereas with bonding strength vector of the data points as $[0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.3]$ is 0.05. Thus, the latter one is highly compact and, therefore, less separable. The size of the cluster is also accounted in this definition for giving priorities to compact and additionally larger clusters. The complexity of computing the bonding strength of a data point and a cluster are both $O(n^2)$, where $n$ denotes the number of data points in the cluster.

5.2. Proposed Method

With these precursory definitions, now let us have an overview of the prioritization process as formalized in Algorithm 1. Initially, the clusters of order 1 or 2 are integrated to produce a single cluster of outliers. Finally, the evolved clusters are ordered by their bonding strength to produce the prioritized clusters in the order from higher to lower. In this way, the complete process is accomplished.

**Algorithm 1** An algorithm for prioritization of clusters

**Input:** A set of clusters $C$.

**Output:** A set of clusters $C_{new}$ ordered by priority.

**Algorithmic Steps:**

1. Integrate the clusters having order less than three in a single cluster.
2. For each cluster $c_j \in C_{new}$ do
3. Find the bonding strength $B_{c_j}$.
4. End for
5. Return the clusters in $C_{new}$ ordered decreasingly by their bonding strength.

Algorithm 2 describes a cluster refinement process employing the prioritization. Given a clustering outcome $C$, we rearrange the clusters for the improvement of clustering solution. For this purpose, a temporary variable $C_{new}$ is used to keep the clustering solution undergoing rearrangement. At the beginning (Step 1), $C_{new}$ is initialized with $C$, the original clustering solution as input. Now, the complete process of rearrangement (Steps 2-16) is iterated until we get a new improved clustering solution or reach the maximum number of iterations (tracked by the variable $MAX$-ITER). Within this, for each data point the clusters are arranged by shifting the data points which yield a better bonding strength. Ties might appear while selecting the cluster ($c_{min}$) with minimum bonding strength (Step 7). If (eventually) the refinement process produces the same set of original clusters $C$, the iterative subroutine terminates (Steps 13-15) indicating the failure of the scope for improvement.
Algorithm 2: An algorithm for prioritization-based cluster refinement

**Input:** A set of clusters $C$ and the maximum iteration number $MAX-ITER$.

**Output:** A set of refined clusters $C_{new}$.

**Algorithmic Steps:**

1. $C_{new} \leftarrow C$.
2. **while** $MAX-ITER$ not reached **do**
   3. **for** each data point $d_i \in C_{new}$ **do**
   4. **for** each cluster $c_i \in C_{new}$ **do**
   5. Find the bonding strength $B_{d_i/c_i}$.
   6. **end for**
   7. Select the cluster $c_{min}$ for which the $B_{d_i/c_i}$ value is minimum. Ties are resolved arbitrarily.
   8. Remove $d_i$ from the cluster it was belonging in (say $c_{pre}$) and assign it to the new cluster $c_{min}$.
   9. **if** $(B_{c_{pre}} + B_{c_{min}})$ does not improve with this cluster rearrangement **then**
   10. Assign $d_i$ back to the cluster $c_{pre}$ and remove from $c_{min}$.
   11. **end if**
   12. **end for**
   13. **if** $C_{new} = C$ **then**
   15. **end if**
   16. **end while**

5.3. Complexity Analysis

The worst case computational complexity of computing the bonding strength of a cluster (the routine shown in Algorithm 1) is $O(k^2)$, where $k$ denotes the number of data points in the cluster. This computational cost gets involved in the cluster reformation process shown in Algorithm 2. Therefore, the computational complexity of this routine becomes $O(m^2n)$, where $m$ denotes the total number of data points and $n$ denotes the number of clusters. This is however a bit computationally expensive. We describe the results obtained and studied on gene expression data using this methodology in the following section.

6. Experimental Study

There are a large number of static and dynamic clustering algorithms available in the literature. It is really a matter of argument that how proper candidates for comparing a clustering result should be selected. But as our method incorporates a post-processing subroutine to be applied on the data points available beforehand, we validate the results in comparison with a widely-used static clustering algorithm. All the experiments have been carried out in a machine with Intel(R) Core(TM) i5-2410M processor running at 2.30 GHz speed and having 4 GB primary memory. We study two microarray datasets containing gene expression profiles of yeast and rat for this purpose.

6.1. Results on Yeast Sporulation Dataset

We have analyzed a microarray dataset prepared to study the sporulation of *Saccharomyces cerevisiae* (baker’s yeast) [11]. It originally contained the expression profiles of 6118 genes measured across 7 time points (0, 0.5, 2, 5, 7, 9 and 11.5h) during the sporulation process of budding yeast. On log-transforming the entire dataset, the genes whose expression levels did not change significantly have been discarded from further analysis. This is determined with a threshold level of 1.6 for the root mean squares of the log2-transformed ratios. The final remaining set contains 474 genes.

To verify the effectiveness of our proposed method of cluster refinement, we have first clustered the dataset using standard k-means clustering algorithm [15]. For this, the number of clusters is set to be 20. After obtaining the clusters, the proposed clustering process (Algorithm 2) is applied onto it. It gets iterated for four times and results
Different statistics & Prioritized k-means

<table>
<thead>
<tr>
<th>Different statistics</th>
<th>k-means</th>
<th>Prioritized k-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>min $\alpha(C)$</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>max $\alpha(C)$</td>
<td>48</td>
<td>50</td>
</tr>
<tr>
<td>min $B_C$</td>
<td>3.612055</td>
<td>3.612055</td>
</tr>
<tr>
<td>max $B_C$</td>
<td>47.03953</td>
<td>49.08866</td>
</tr>
<tr>
<td>$\mu_B$</td>
<td>22.39794</td>
<td>22.44397</td>
</tr>
<tr>
<td>$\sigma_B$</td>
<td>12.96152</td>
<td>14.10063</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GO term</th>
<th>Analysis of the cluster obtained before prioritization</th>
<th>Analysis of the cluster obtained after prioritization</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO ID</td>
<td>#Genes</td>
<td>GO ID</td>
</tr>
<tr>
<td>Cellular catabolic process</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Carboxylic acid metabolic process</td>
<td>0019752</td>
<td>13</td>
</tr>
<tr>
<td>Oxoacid metabolic process</td>
<td>0043436</td>
<td>13</td>
</tr>
<tr>
<td>Cellular ketone metabolic process</td>
<td>0042180</td>
<td>13</td>
</tr>
<tr>
<td>Organic acid metabolic process</td>
<td>0006082</td>
<td>13</td>
</tr>
<tr>
<td>Glucose catabolic process</td>
<td>0006007</td>
<td>11</td>
</tr>
<tr>
<td>Hexose catabolic process</td>
<td>0019320</td>
<td>11</td>
</tr>
<tr>
<td>Monosaccharide catabolic process</td>
<td>0046365</td>
<td>11</td>
</tr>
<tr>
<td>Alcohol catabolic process</td>
<td>0046164</td>
<td>11</td>
</tr>
<tr>
<td>Glucose metabolic process</td>
<td>0006006</td>
<td>11</td>
</tr>
<tr>
<td>Carbohydrate catabolic process</td>
<td>0016052</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 1. Comparative study of the clustering results without and with prioritization on yeast gene expression dataset.

Table 2. Comparative study on a single cluster with 37 genes found from the yeast sporulation dataset based on the largest ten functional enrichment coverage found from GO. The ID and gene count corresponding to GO terms not obtained in the enrichment analysis are marked with ‘NA’.

into the same number of clusters. Different statistics of the clusters obtained, in original result and after prioritizing, are shown in Table 1. On comparing both of them, the original clusters seem to be improved (showing more bonding strength within the clusters) by prioritization. Again advantageously, the result obtained after prioritizing the clusters is in the form of an ordered list of gene clusters. These clusters are arranged by their statistical coherence (in terms of bonding strength) and we hypothesize that biologically the superior ones are likely to be the promising gene clusters for further study.

The prioritization subroutine is simply a rearrangement process applied on the clusters to make them biologically coherent. Therefore, after reorganizing the clusters we have considered a single cluster for further analyzing its biological significance. Our aim is to verify whether a cluster becomes more coherent by our proposed method of cluster refinement. We mapped the clusters from the original set to the revised ones obtained after prioritization and the only cluster whose size remained unaltered (but bonding strength changes) is taken for analysis. We stick to a same-sized cluster because changes in size may bias the comparative results. We have carried out functional enrichment analysis on this cluster and its corresponding reorganized cluster (obtained by applying Algorithm 1) for post-genomic analysis. To do this, a sophisticated web tool Funcassociate is used to extract functional enrichment information based on gene ontology (GO) [8]. The web server version Funcassociate 2.0 is freely available at: http://llama.med.harvard.edu/funcassociate. The annotation results of the top ten GO terms that are associated to majority of the genes contained in this cluster are shown in Table 2. In all the cases shown, we have obtained more biologically significant clusters (as the numbers of genes having similar functions get increased) by the prioritization. This is indeed an indication of clustering improvement.

To further study the significance of the clusters, we have computed the $p$-values obtained for the annotations of the single cluster under observation. The functional annotations in GO are used for this purpose. The $p$-value, as decreases for a specific function annotated, indicate that the annotation result has a lower probability to be obtained by chance. The best ten functional enrichment $p$-values found from GO for the single cluster, in the original set and after prioritization, are listed in Table 3. As can be seen from the table, for the same biological function of glucose catabolic process we obtained the $p$-values 1.09E-15 and 1.40E-17 before and after prioritization, respectively. Certainly, this reduction of $p$-value by the order $10^{-2}$ denotes the superiority of the revised cluster. The proposed priority based reorganization of the clusters provides a better co-functional gene cluster. Similarly, the $p$-values for
Table 3. Comparative study on a single cluster with 37 genes based on the best ten functional enrichment p-values found from GO. The ID and p-value corresponding to GO terms not obtained in the enrichment analysis are marked with ‘NA’.

<table>
<thead>
<tr>
<th>GO term</th>
<th>Analysis of the cluster obtained before prioritization</th>
<th>Analysis of the cluster obtained after prioritization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyruvate metabolic process</td>
<td>0006090 3.60E-16</td>
<td>0006090 3.60E-16</td>
</tr>
<tr>
<td>Glucose catabolic process</td>
<td>0006007 1.09E-15</td>
<td>0006007 1.40E-17</td>
</tr>
<tr>
<td>Hexose catabolic process</td>
<td>0019320 4.52E-15</td>
<td>0019320 6.75E-17</td>
</tr>
<tr>
<td>Monosaccharide catabolic process</td>
<td>0046365 1.06E-14</td>
<td>0046365 1.73E-16</td>
</tr>
<tr>
<td>Alcohol catabolic process</td>
<td>0046164 2.81E-14</td>
<td>0046164 5.08E-16</td>
</tr>
<tr>
<td>Gluconeogenesis</td>
<td>0006094 4.56E-14</td>
<td>0006094 6.75E-14</td>
</tr>
<tr>
<td>Glycolysis</td>
<td>0006096 5.37E-14</td>
<td>0006096 5.37E-14</td>
</tr>
<tr>
<td>Glucose metabolic process</td>
<td>0006006 1.33E-13</td>
<td>0006006 2.83E-15</td>
</tr>
<tr>
<td>Carbohydrate catabolic process</td>
<td>NA</td>
<td>0016052 1.47E-14</td>
</tr>
<tr>
<td>Hexose metabolic process</td>
<td>NA</td>
<td>0019318 8.11E-14</td>
</tr>
<tr>
<td>Hexose biosynthetic process</td>
<td>0019319 1.90E-13</td>
<td>NA</td>
</tr>
<tr>
<td>Alcohol biosynthetic process</td>
<td>0046165 2.13E-13</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 4. Comparative study of the clustering results without and with prioritization on rat CNS expression dataset.

<table>
<thead>
<tr>
<th>Different statistics</th>
<th>k-means</th>
<th>Prioritized k-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \min \alpha(C) )</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>( \max \alpha(C) )</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>( \min \beta_C )</td>
<td>4.670836</td>
<td>4.699011</td>
</tr>
<tr>
<td>( \max \beta_C )</td>
<td>16.804409</td>
<td>15.937217</td>
</tr>
<tr>
<td>( \mu_{\beta_C} )</td>
<td>9.17661</td>
<td>9.2149847</td>
</tr>
<tr>
<td>( \sigma_{\beta_C} )</td>
<td>3.974585</td>
<td>3.957703</td>
</tr>
</tbody>
</table>

the biological functions like hexose catabolic process, monosaccharide catabolic process, alcohol catabolic process and glucose metabolic process from the best ten studied annotations (as shown in Table 3) get improved with this prioritization. This not only shows the importance of the prioritization of the clusters but also demonstrates its biological effectiveness for the improvement of clustering as a post-processing task.

6.2. Results on Rat CNS dataset

This experimental data is collected from a reverse transcription-coupled PCR experiment producing mRNA expression profiles of 112 genes during rat Central Nervous System (Rat CNS) development [27]. The original dataset contains expression values (\( \log_2(\frac{R}{G}) \)) of 111 genes over 9 time points. The expression data of one gene containing missing value has been omitted from the dataset and remaining portion has been used in this study. We have computed the Pearson correlation coefficient between each gene pair and taking this value as the proximity measure k-means algorithm is run on the data with the number of clusters set as 10. The number of iterations is considered as 100. The different statistics on the clusters, in original result and after prioritizing, are shown in Table 4. After prioritization that lasted for only 4 iterations, we obtained same number of clusters.

The results from Funcassociate 2.0 is used to interpret the functional enrichment of the clusters based on GO [8]. We obtained the same three biologically significant involvements – regulation of peptidyl-tyrosine phosphorylation, growth factor activity and transmembrane receptor protein tyrosine kinase signaling pathway however the sizes of the significant clusters got reduced by the prioritization. This is indeed an indication of clustering improvement. The cluster size reduces keeping the same significant genes involved therein. Similar to the results on the previous dataset, this analysis also establishes the significance of the prioritization procedure.

7. Conclusion

This paper describes an important problem of quantifying the priorities of a set of clusters. It is helpful for the effective clustering of genes aiming post-genomic analysis. However, the methodology is not explicit towards this
specific application. The proposed methodology could be used as an improvement to any existing clustering algorithm applicable to gene expression data or other real-world networks reflecting scale-freeness [6]. This is also applicable to both static and dynamic clustering. We highlight twofold advantages of our method. First, the prioritization of clusters to find co-functional genes and the prediction of gene functions. Second, providing an improvement to the clustering result for post-genomic analysis. The empirical studies establishes the efficacy of the proposed algorithm. It can also be extended for any type of biomolecules with expression data readily available.

Acknowledgments

M. Bhattacharyya gratefully acknowledges the financial support from the DST-PURSE grant. The work of S. Bandyopadhyay was supported by the grant no.- DST/SJF/ET-02/2006-07 under the Swarnajayanti Fellowship scheme of the Department of Science and Technology, Government of India. Additionally, the authors are thankful to the anonymous reviewers for their helpful suggestions.

References