Journal of Data Mining and Knowledge Discovery
ISSN: 2229–6662 & ISSN: 2229–6670, Vol. 2, Issue 1, 2011, pp-30-35
Available online at http://www.bioinfo.in/contents.php?id=42

IMPROVED MOUNTAIN CLUSTERING ALGORITHM FOR GENE EXPRESSION DATA ANALYSIS

NISHCHAL K. VERMA[1*], ABHISHEK ROY[2], YAN CUI[3]
[1]Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, India - 208016
[2]Department of Electrical and Electronics, National Institute of Technology Karnataka, Surathkal, Mangalore, India - 575025
[3]Department of Molecular Sciences, Center for Integrative and Translational Genomics, University of Tennessee Health Science Center, Memphis, TN, USA - 38163
*Corresponding Author: Email- nishchal.iitk@gmail.com

Received: July 24, 2011; Accepted: August 26, 2011

Abstract- With the advent of expression microarray and other high-throughput technologies, it has been made possible to perform transcriptome-wide analysis of gene expression. Clustering analysis becomes essential for interpreting huge volume of gene expression data and for deriving useful biological information they contain. Five clustering algorithms were systematically evaluated upon three microarray datasets in a recent study [1] to compare the performances of these algorithms on complex human tissues. In this paper, we propose Improved Mountain Clustering (IMC), a new algorithm for gene expression data analysis. We used the Gene Ontology enrichment index and the microarray datasets in [1] to evaluate the performance of IMC. We extend the comparison to another two clustering methods, Fuzzy c-means (FCM) and self-organizing maps (SOM). K-means clustering was also included in the comparison to calibrate our results with that of [1]. The performance of K-means, FCM, SOM, IMC-1 (the original IMC) and IMC-2 (a modified version of IMC) were evaluated on the three datasets at four p-value thresholds for GO enrichment. In 66.67% cases, IMC-1 and IMC-2 outperformed other algorithms, whereas in 33.33% cases, tied with the best performer of other algorithms. IMC-1 and IMC-2 were the fastest among the tested algorithms due to their low computational complexity. The IMC algorithms outperformed K-means, FCM and SOM in the tests on the expression microarray datasets. As K-means outperformed other clustering algorithms tested in [1] and same criteria and datasets are used in this work, IMC should be the most effective among all the seven algorithms (i.e. IMC, K-means, FCM, SOM, CRC, ISA and memISA).
Key words – Expression microarray, Improved Mountain Clustering, Gene Ontology enrichment, Hypergeometric distribution, Mountain function, Euclidean distance, Transcriptome data, Computational complexity.

BACKGROUND

Clustering methods have been widely used in large-scale gene expression data analysis. Genes with similar expression pattern are grouped into clusters. Thus, genes belonging to same cluster(s) may respond to experimental conditions concertedly. They are also likely to share similar functions and involve in same biological processes [1-3].

Improved mountain clustering (IMC) is a newly developed algorithm and has been applied to the colour segmentation problem in image analysis [4]. IMC outperformed other clustering algorithms including K-means [5, 6], FCM [7], EM [8] and MMC [9] in image segmentation [4]. Its capability of extracting high quality clusters from high volume data and its low computational complexity [4] makes IMC a promising method for large-scale gene expression data analysis.

In a recent comparative study of clustering methods for expression microarray data analysis, Gene Ontology enrichment indexes were used to evaluate the performance of four clustering algorithms, i.e. K-means [5, 6], Chinese Restaurant Clustering (CRC) [10], the Iterative Signature Algorithm (ISA) [11] and a new, progressive variant of ISA called memISA [12], on three brain expression microarray datasets [1]. Here we used the same criteria and datasets to test the performance of IMC and three other algorithms, K-means, FCM and SOM [13, 14]. SOM is a widely used clustering method for microarray data analysis. FCM algorithm is a well-known technique for pattern classification and has also been used for expression microarray data analysis [15]. K-means outperformed other clustering methods in [1], so we repeated the test on K-means to calibrate our results with theirs. Two versions of IMC, IMC-1 and IMC-2, were tested. IMC-1 refers to the original IMC whereas IMC-2 refers to a modified version of IMC which is optimized heuristically by modifying its threshold function.

METHODS

Datasets

Gene clustering was performed on three brain expression microarray datasets: Perrone–Bizzozero (PB) [16], McLean 66 (MC66) [17] and Dobrin [18]. Table 1 in [1] summarized the information about the datasets. The clustering was performed on exactly the same pre-processed datasets as what used in [1].

Estimation of Number of Clusters

In general, performance of a clustering method is affected by varying parameters such as the number of clusters [19]. In spite of availability of various methods in the literature for estimation of number of clusters M [20], its estimation becomes close to impossible in a real microarray data while dealing with gene clustering. Moreover, in an organism, with reference to the complexity of its genetic interaction, intuitively speaking deciding a particular value for M becomes almost impractical [19]. Here, analysis being done on the same microarray datasets used in [1], the optimum number of clusters used is referred from [1]. However, the performance of techniques is examined over an entire range of nearby or practical values of M .

Gene Ontology Enrichment

The Gene Ontology (GO) enrichment index can be defined as the percentage of significantly enriched clusters (below a pre-defined p-value cut-off), with genes from one or more gene ontology categories (from the goa_human database) at different significance levels, using Fisher's exact test and Benjamini false discovery rate multiple testing correction [21]. For all GO biological process terms, clusters were examined for enrichment with minimum third order in GO hierarchy. To avoid affecting results adversely due to chance appearance of 1 or 2 genes from a GO category with few members, at least 3 genes from the input cluster had to match a GO category for the cluster to be called enriched for that category [1]. The percentage of clusters that are found enriched will give a measure of the biological, functional relevance of the clusters.

GO enrichment is calculated with the help of web-based service, GOstat [22]. This accepts group IDs, of clustered genes which are to be annotated and of the total genes in the microarray data as input. The enrichment p-value is calculated using hypergeometric distribution [23].

K-means Clustering

K-means [5, 6] is one of the most widely used clustering techniques. In a recent comparative study of four clustering methods for brain expression microarray data, K-means outperformed other clustering methods [1]. In this algorithm it is required to specify the number of clusters, M initially. This technique is a hard clustering technique, i.e. each gene can belong to only one cluster. In this technique M centroids (quasi data points representing the centers of the clusters) are distributed at random among the data points. Each data point is assigned to the group that has the closest centroid. When all objects have been assigned, centroids are moved to minimize its distance with the associated data points. This process is repeated until centroids stop moving. This is performed for the given number of iterations. Among the various obtained configurations, the one with the smallest distance between points and their associated centroids is considered as the outcome for the clustering solution. K-means was performed using an inbuilt function 'kmeans' in MATLAB 7.5.0

Fuzzy C-Means (FCM)

Fuzzy c-means (FCM) [7] is a method of clustering which allows one object to belong to two or more clusters. This method, developed by Dunn in 1973 [24] and improved by Bezdek in 1981[25], is frequently used in pattern recognition. In gene expression data analysis, it links each gene to all clusters via a real-valued vector of indices. It assigns degree of membership to each gene. The degree of membership varies between zero to one. Genes with higher degrees of membership for a cluster are more strongly associated with that cluster. FCM was performed using an inbuilt function 'fcm' in MATLAB 7.5.0

Self-Organising Maps (SOM)

SOM [13, 14] clustering starts with a set of nodes with randomly selected geometry (e.g., a 3 x 2 grid) in the k -dimensional gene expression space. The positions of the nodes are adjusted iteratively. Each iteration involves random selection of a gene x and the nodes are moved in the direction of x . The closest node N_x is moved the most and other nodes are moved by smaller amounts. The further away the node is from N_x , the less it is moved. The process is repeated for required number of iterations. SOM was performed using a web-based software package, GenePattern [26].

Improved Mountain Clustering (IMC)

IMC was first introduced by Nishchal K. Verma and M Hanmandlu in [4], where it was applied to color segmentation and outperformed other clustering algorithms for providing high quality clusters and low computational complexity [4]. Here we used IMC to analyze the complex brain expression microarray data.

The Algorithm

Step 1: Normalize each dimension of hyper-space, so that the data points are bounded by the unit hypercube.

We define the j^{th} data in $\mathbf{x}$ hyperspace as:

$$\mathbf{x}^j = \left\{ x_1^j, x_2^j, ..., x_D^j \right\} \quad (1)$$

where, D is the total number of dimensions of hyperspace.

The normalized data point $\bar{\mathbf{x}}^j$ is defined as:

$$\bar{\mathbf{x}}^j = \frac{\mathbf{x}^j - (\mathbf{x})_{\min}}{(\mathbf{x})_{\max} - (\mathbf{x})_{\min}}; \quad \forall \ j = 1, 2, ..., n \quad (2)$$

where,

$$(\mathbf{x})_{\min} = \left\{ \min_{j=1}^{n} x_1^j, \min_{j=1}^{n} x_2^j, ..., \min_{j=1}^{n} x_D^j \right\} \quad (3)$$

$$(\mathbf{x})_{\max} = \left\{ \max_{j=1}^{n} x_1^j, \max_{j=1}^{n} x_2^j, ..., \max_{j=1}^{n} x_D^j \right\} \quad (4)$$

and n is the total number of data points.

Step 2: Determine the threshold value d_1 for each window. d_1 is the positive constant defining the

neighbourhood of the data point. We compute these from the heuristics:

$$d_1 = \frac{1}{2n}\sum_{j=1}^{n}\left(\frac{\min\left(\mathbf{x}^j\right)}{\sum_{i=1}^{D} x_i^j}\right).(\alpha)\ ; \qquad (5)$$

where,

$$\alpha = \frac{M}{M+1}$$

and, M is the number of clusters.

Step 3: Calculate the potential value of each point P_1^r using mountain function, which is a function of distance $d^2\left(\overline{\mathbf{x}}^r, \overline{\mathbf{x}}^j\right) = \left(\overline{\mathbf{x}}^r - \overline{\mathbf{x}}^j\right) Q \left(\overline{\mathbf{x}}^r - \overline{\mathbf{x}}^j\right)'$ between $\overline{\mathbf{x}}^r$ and all other data points.

$$P_1^r = \sum_{j=1}^{n} \exp\left[-\left(\frac{d^2\left(\overline{\mathbf{x}}^r, \overline{\mathbf{x}}^j\right)}{d_1^2}\right)\right]\ ;\ \forall\ r = 1,2,...,n \quad (6)$$

Step 4: Select the first cluster center according to the highest value of P_1^r as,

$$\overline{c}_1 = \overline{\mathbf{x}}^* \Leftarrow P_1^* = \max_{r=1}^{n}\left(P_1^r\right) \qquad (7)$$

Here, value of $*$ is that value of r at which the value of P_1^r is found to be the highest.

Step 5: Assign those data points to the first cluster whose Euclidean distance from the first cluster center is less than a threshold, d_1 i.e.

If

$$d^2\left(\overline{\mathbf{x}}^r, \overline{c}_1\right) \le d_1\ ; \qquad \forall\ r = 1,2,...,n \qquad (8)$$

then $\overline{\mathbf{x}}^r$ is assigned to the first cluster.

Step 6: Remove all those data points from the total dataset which are assigned to the cluster formed.

Step 7: Repeat Steps 2 to 6 for the remaining data to make successive clusters. Similarly for selection of m^{th} cluster center, revision of potential value is done for the reduced dataset and m^{th} cluster center is selected with the highest value of P_m^r as,

$$\overline{c}_m = \overline{\mathbf{x}}^* \Leftarrow P_m{}^* = \max_{r=1}^{n}\left(P_m^r\right) \qquad (9)$$

Step 8: Form required number of clusters M, using the Steps 2 to 7 and then separate out these clusters from the whole dataset. Rest of the data points are distributed among the formed clusters depending upon their Euclidean distances, i.e. nearness to the respective cluster centers.

The above algorithm corresponds to IMC-2 and the difference it has with IMC-1 is the factor α in (5) with which earlier expression for threshold function is multiplied heuristically to provide improved threshold function value in IMC-2. The codes for IMC-1 and IMC-2 clustering algorithms were developed and implemented on MATLAB 7.5.0.

RESULTS AND DISCUSSION

Performance comparison of the clustering algorithms

We have applied all the four clustering techniques to three different microarray datasets. The clusters obtained were tested for enrichment in GO biological process categories using GOstat for four different p-value cut-offs i.e. p <0.0001, p <0.001, p <0.01 and p <0.1.

The IMC algorithms clearly outperformed K-means, FCM and SOM (Figure 1-3). SOM was found to form average quality clusters whereas, FCM gave the least enriched clusters. Detailed performance data can be found in Additional file 1. The performance of K-means, FCM, SOM, IMC-1 (the original IMC) and IMC-2 (a modified version of IMC) were evaluated at optimum number of clusters for the three datasets at four p-value thresholds for GO enrichment. In 66.67% cases, IMC-1 and IMC-2 outperformed, whereas in 33.33% cases tied with the best performer of K-means, FCM and SOM. The analysis done for a range of values of M to measure the effectiveness of clustering techniques is shown by pooling the results. The range is chosen around the estimated optimum number of clusters for each dataset, i.e. $M \in [2,12]$, so that the ambiguity over fixing a particular number of clusters for a microarray data with thousands of genes in each cluster (Additional file 2) would not bias the result. IMC-2 even further improved the results of IMC-1 in terms of GO enrichment in PB dataset. Analysing the pooled results in PB dataset, there are 66% cases with IMC-1 and IMC-2 outperforming the best performer of K-means, FCM and SOM, while being second to none in most of the remaining cases (Additional file 1).

Computational Complexity

Computational complexity refers to the number of steps involved in the algorithm of a clustering method. Computational complexity of various clustering methods is compared to obtain their relative efficiency in terms of time complexity i.e. $O(.)$. The number of steps involved in clustering via IMC is less than K-means, FCM and SOM. Clearly, this reduces the time involved in clustering microarray data, which is important, given the volume of transcriptome data. Computational complexity of all the methods compared here is shown in Table 1.

CONCLUSION

Though all the clustering methods discussed here are well-proven for producing quality clusters, still when it comes to the complex transcriptome data, IMC-2 produces clusters with higher level of GO enrichment than the other clustering methods compared. This new clustering method provides an excellent choice for transcriptome data analysis for its capability of

discovering biologically meaningful clusters while having additional advantage of lower computational complexity.

Authors' Contributions

Nishchal K. Verma and Yan Cui conceived the idea of the work. Abhishek Roy wrote the source codes and analyzed the data. All the authors together wrote the manuscript.

Acknowledgements

We thank Alexander L Richards for providing the three brain expression microarray datasets and for his help with the use of related analysis tools.

References

[1] Richards A.L., Holmans P., O'Donovan M.C., Owen M.J. and Jones L. (2008) *BMC Bioinf*., 9, 490.

[2] Azuaje F. and Bolshakova N. (2002) *In: Berrar D., Dubitzky W. and Granzow M. (eds.), A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, Boston/Dordrecht/London*, 230-245.

[3] Wolfe C.J., Kohane I.S. and Butte A.J. (2005) *BMC Bioinf*., 6(227).

[4] Verma N.K. and Hanmandlu, M. (2007) *International Journal of Image and Graphics*, 7(2), 407-426.

[5] MacQueen J.B. (1967) *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, California*, 1, 281-297.

[6] Hartigan J.A. and Wong M.A. (1979) *Appl. Stat*., 28(1), 100-108.

[7] Razaz M. (1993) *IEEE International Symposium on Circuits and Systems, ISCAS*, 3, 2051-2054.

[8] Dempster A.P., Laird N.M. and Rubin D.B. (1977) *J. Roy. Statist. Soc. Ser. B*, 39(1), 1-38.

[9] Azeem M.F., Hanmandlu M. and Ahmad N. (1999) *2nd International Conference on Information Technology*, 63-68.

[10] Qin Z.S. (2006) *Bioinformatics* 22(16), 1988-1997.

[11] Bergmann S., Ihmels J. and Barkai N. (2003) *Phys. Rev. E Stat. Nonlin. Soft Matter Phys*., 67(3 Pt 1).

[12] Kloster M., Tang C. and Wingreen N.S. (2005) *Bioinformatics*, 21(7), 1172-1179.

[13] Kohonen T. (1990) *Proceedings of the IEEE*, 78(9), 1464-1480.

[14] Tamayo P., Slonim D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S. and Golub T.R. (1999) *Proc. Natl. Acad. Sci. USA*, 96(6), 2907-2912.

[15] Dembele D. and Kastner P. (2003) *Bioinformatics*, 19(8), 973-980.

[16] Barrett T., Troup D.B., Wilhite S.E., Ledoux P., Rudney D., Evangelista C., Kim I.F., Soboleva A., Tomashevsky M. and Edgar R. (2006) *Nucleic Acids Res*., 35, D760-D765.

[17] *National Brain Databank, Brain Tissue Gene Expression Repository. [http://national_databank.mclean.harvard.edu/brainbank/Main].*

[18] Higgs B.W., Elashoff M., Richman S. and Barci, B. (2006) *BMC Genomics*, 7(70).

[19] Thalamuthu A., Mukhopadhyay I., Zheng X. and Tseng G.C. (2006) *Bioinformatics*, 22(19), 2405-2412.

[20] Milligan G.W. and Cooper M.C. (1985) *Psychometrika*, 50(2), 159-179.

[21] Khatri P., Draghici S. (2005) *Bioinformatics*, 21(18), 3587-3595.

[22] Beißbarth T. and Speed T.P. (2004) *Bioinformatics*, 20(9), 1464-1465.

[23] Tavazoie S., Hughes J.D., Campbell M.J., Cho R.J. and Church G.M. (1999) *Nat. Genet*., 22(3), 281-285.

[24] Dunn J.C. (1973) *Journal of Cybernetics*, 3(3), 32-57.

[25] Bezdek J.C. (1981) *Plenum Press, New York, USA.*

[26] Reich M., Liefeld T., Gould J., Lerner J., Tamayo P. and Mesirov J.P. (2006) *Nat. Genet*., 38(5), 500-501.

Table-1 - Computational Complexity of various clustering techniques.

Computational Complexity	K-Means	FCM	SOM	IMC-1	IMC-2
Time- Complexity	$O(N.i)$	$O(N.i)$	$O(N.i)$	$O\left(\left(N-\sum_{r=0}^{M-1}N_r\right)^2\right)$	$O\left(\left(N-\sum_{r=0}^{M-1}N_r\right)^2\right)$

Here, N is the number of genes, i is the number of iterations and M is the number of clusters.

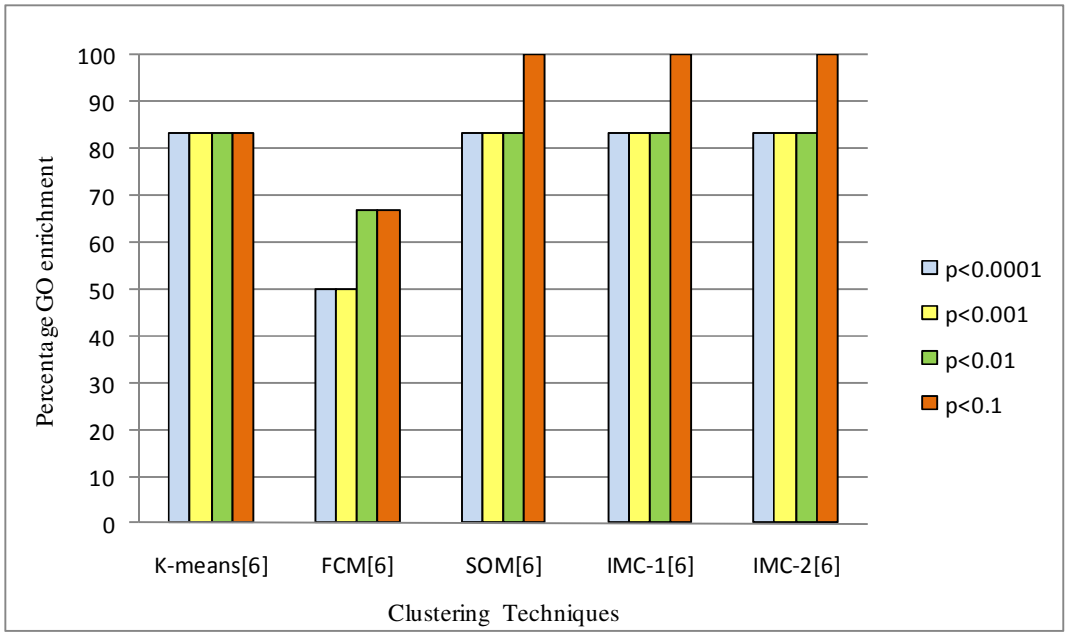

Fig. 1- GO enrichment of clusters for all methods - Dobrin dataset. Light blue, yellow, green and orange bars are the percentage of clusters that are significantly enriched for one or more GO categories at p <0.0001, 0.001, 0.01 and 0.1 respectively. Numbers in square brackets are the optimum number of clusters for the analyzed dataset.

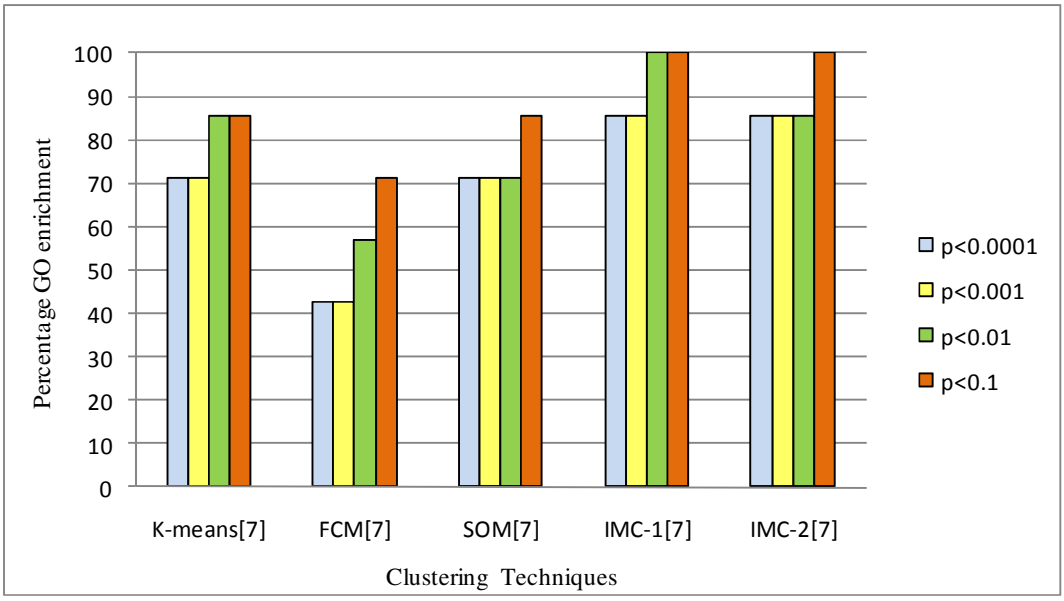

Fig. 2- GO enrichment of clusters for all methods – MC66 dataset. Light blue, yellow, green and orange bars are the percentage of clusters that are significantly enriched for one or more GO categories at p <0.0001, 0.001, 0.01 and 0.1 respectively. Numbers in square brackets are the optimum number of clusters for the analyzed dataset.

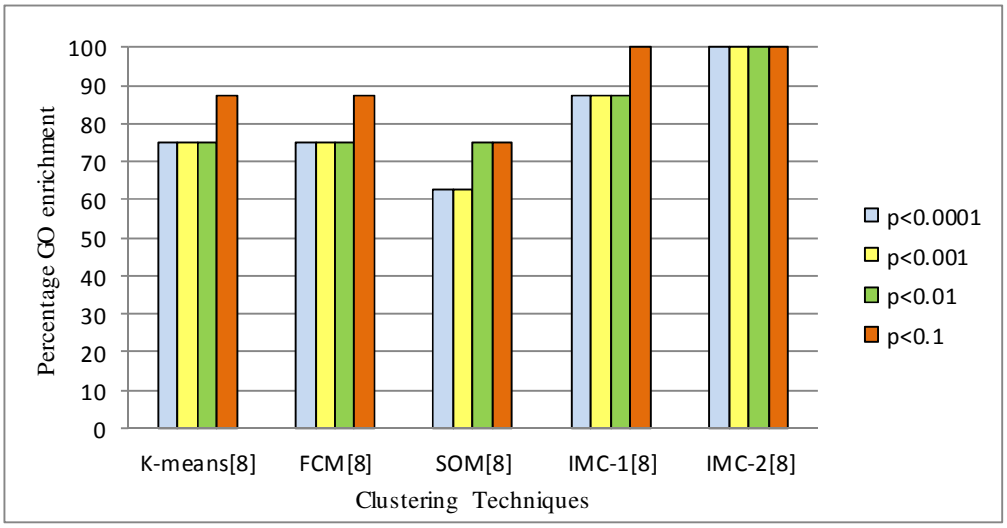

Fig. 3- GO enrichment of clusters for all methods – PB dataset. Light blue, yellow, green and orange bars are the percentage of clusters that are significantly enriched for one or more GO categories at p <0.0001, 0.001, 0.01 and 0.1 respectively. Numbers in square brackets are the optimum number of clusters for the analyzed dataset.

ADDITIONAL FILES

Additional file 1 – Pooling Results for various clustering techniques
Spreadsheet showing the pooling results for various clustering techniques at significant p cut-off values for the datasets used – Perrone Bizzozero Dataset (PB), McLean66 (MC66) Dataset and Dobrin Dataset.

Additional file 2 – Distribution of cluster sizes
Spreadsheet showing number of genes present (cluster size) in each cluster for each method across all datasets.