BINARIZATION OF CONSENSUS PARTITION MATRIX FOR ENSEMBLE CLUSTERING

Basel Abu-Jamous[1], *Rui Fa*[1], *Asoke K Nandi*[1,2] *and David J. Roberts*[3]

[1]Signal Processing and Communications Research Group, Department of Electrical Engineering and Electronics,
The University of Liverpool, L69 3GJ, UK. {basel88, r.fa, a.nandi}@liverpool.ac.uk

[2] Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland.

[3] National Health Service Blood and Transplant, The University of Oxford, Oxford, UK. david.roberts@ndcls.ox.ac.uk

ABSTRACT

In this paper, a new paradigm of clustering is proposed, which is based on a new Binarization of Consensus Partition Matrix (Bi-CoPaM) technique. This method exploits the results of multiple clustering experiments over the same dataset to generate one fuzzy consensus partition. The proposed tunable techniques to binarize this partition reflect the biological reality in that it allows some genes to be assigned to multiple clusters and others not to be assigned at all. The proposed method has the ability to show the relative tightness of the clusters, to generate tight cluster or wide overlapping clusters, and to extract the special genes which bear the profiles of multiple clusters simultaneously. A synthetic periodic gene dataset is analysed by this method and the numerical results show that the method has been successful in showing different horizons in gene clustering.

Index Terms— Ensemble clustering, Consensus function, Fuzzy partition, Binarization of Consensus Partition Matrix (Bi-CoPaM).

1. INTRODUCTION

Data clustering is one of the widely used classes of unsupervised learning techniques. Many methods have been introduced in the literature to group the individual objects in a population within which the objects are more similar, such as self-organizing maps (SOMs) [1], k-means [2], hierarchical clustering [3], self-organizing oscillator networks (SOONs) [5], fuzzy clustering [6], model-based clustering [7] and others. However, robustness and reliability are always problems such that none of the methods is a perfect method which always gives the best results for all types of datasets.

In order to enhance the robustness of clustering, the idea of clustering ensembles has been proposed where the results of many clustering experiments are combined. However, many obstacles have been found while combining results from different clustering experiments [8]. One main problem is that due to the unsupervised nature of clustering, it is not a straightforward task to map a specific cluster from one of the clustering results to its corresponding cluster from another clustering result. Another problem is that different clustering results may give different numbers of clusters while the correct number of clusters is unknown.

Consensus function method has been employed as an essential step to solve aforementioned problems in clustering ensembles. There are many methods for consensus function, including relabelling and voting [9], co-association matrix [10], hypergraph methods [11], weighted kernel consensus functions [12], non-negative matrix factorization [14], greedy algorithms [13], and others. In all of the aforementioned methods, the final consensus partition assigns each object to one of the final clusters exclusively, that is, no objects are unassigned and no objects are assigned to multiple clusters. In some cases this does not give the best view of the dataset. For instance, in gene discovery through gene clustering, it is common that a single gene participates in many biological processes and thus has a close profile to different clusters at the same time [15]. Applying many clustering experiments over such set of genes may result in the assignment of this gene correctly to different clusters in different experiments. This kind of information has not been explored thoroughly up to now except in some studies such as [15]. Moreover, in many applications, the best view for the data points might not be the one that assigns all of them into distinct clusters. In some applications, one might want to focus on small subsets of points that make up the clusters' cores while accepting that some potentially important points are out of consideration. In some other applications one might like to get large clusters that contain all of the possible data points while accepting that some data points are assigned in multiple clusters.

In this paper, we propose a novel clusters ensemble method, Binarization of Consensus Partition Matrix (Bi-CoPaM) [16], which provides a new paradigm, or a view, of data clustering in order to tackle the aforementioned issues. This method exploits the information provided by different clustering results through generating consensus partitions that

The project (Ref. NIHR-RP-PG-0310-1004-AN) is supported by National Institute for Health Research (NIHR), UK.

allow single object to be assigned to multiple clusters or to be unassigned to any of them at all. This method is a tunable tool which allows the researcher to tighten or widen the clusters according to the needs of a specific research. This Bi-CoPaM method is based on relabelling and voting methods which assume that the number of clusters is known and fixed for all of the partitions. The main steps of the algorithm are: partitions generation, relabelling, fuzzy consensus partition matrix (CoPaM) generation and binarization. There are many advantages of identifying these multiply assigned or unassigned objects, for example, in some applications, such as gene discovery, this may have a direct meaning which is that these genes actually belong to these different biological groups in the case of the multiple-assigned genes, or that they really do not belong to any (irrelevant genes to the problem under consideration) in the case of the unassigned ones. In other applications, these objects may be considered uncertain and be taken into further analysis and discussions. We employ a very simple example using a synthetic gene dataset to show how the proposed method works.

This paper is organized as follows: Sec. 2 details the proposed method Bi-CoPaM, Sec. 3 discusses the empirical results of applying this method on a synthetic gene dataset and Sec. 4 concludes the paper.

2. BI-COPAM METHOD

Assume that the original problem is to group M objects into K clusters. Clustering is carried out over R different experiments which generate R different partition matrices $\{\mathbf{U}^1, \cdots, \mathbf{U}^R\}$. The goal is to find the final consensus binary partition (CBP) $\mathbf{B}^*$ which relaxes the conventional partitioning constraints by allowing some objects to be assigned to multiple clusters at the same time or to be not assigned at all in a way which best reflects the information provided by the partitions. This section describes the principles of the proposed method *Binarization of Consensus Partition Matrix* (Bi-CoPaM) which has four main steps and will be detailed in the following subsections.

2.1. Partitions Generation

In this initial step, R clustering experiments are carried out to generate R partitions. Different experiments can be set up by using different clustering methods with different parameters, i.e. using the same clustering method with different parameters generates different partitions.

Each clustering result can be presented in the form of a fuzzy partition matrix. The matrix $\mathbf{U}_{K\times M}$ is a 2D matrix with K rows representing the clusters and M columns representing the objects. Each element of the matrix $u_{i,j} \in [0,1]$ represents the membership value of the *j*-th object in the *i*-th cluster. A value of 0 means that this object does not belong to this cluster at all and a value of 1 means that it fully belongs to it. The properties of the partition matrix is mathematically expressed by

$$(1) \quad u_{i,j} \in [0,1], \quad 1 \le i \le K, 1 \le j \le M,$$

$$(2) \quad \textstyle\sum_{i=1}^{K} u_{i,j} = 1, \quad 1 \le j \le M,$$

$$(3) \quad 0 < \textstyle\sum_{j=1}^{M} u_{i,j} < M, \quad 1 \le i \le K.$$

2.2. Relabelling

Due to the nature of unsupervised learning where the generated clusters do not have meaningful labels, the results of different clustering experiments over the same dataset are not guaranteed to be aligned. Relabelling a partition matrix U to be aligned with a reference partition matrix $\mathbf{U}^{ref}$ aims at finding a matrix $\bar{\mathbf{U}}$ which represents one of the permutations of the rows of $\mathbf{U}$ such that its similarity to $\mathbf{U}^{ref}$ is maximized. This optimization problem is formulated mathematically in (1):

$$\hat{\mathbf{U}} = \arg \max_{\forall \text{perm}(\mathbf{U})} \Gamma(\mathbf{U}^{ref}, \text{perm}(\mathbf{U})), \tag{1}$$

where perm$(\mathbf{U})$ is a permutation of the rows of $\mathbf{U}$ and $\Gamma(\cdot,\cdot)$ measures the similarity between any two partitions.

The size of the search space is $K!$ which makes any brute force search infeasible solution for large values of $K (K \ge 8)$. For larger values of K, different heuristic approaches can be used to tackle this problem. The approach which is adopted in this paper is the min-max approach which is detailed in the steps below:

(1) A dissimilarity distance matrix $\mathbf{S}_{K\times K}$ is constructed by calculating the pairwise Euclidean distance between the rows (clusters) of the matrix $\mathbf{U}$ and the rows of the reference matrix $\mathbf{U}^{ref}$.

(2) The minimum value in each of the columns is found.

(3) The maximum value of these minima is identified then the rows (clusters) from $\mathbf{U}$ and $\mathbf{U}^{ref}$ which correspond to this similarity value are mapped.

(4) The row and the column which show the aforementioned value are deleted from the similarity matrix.

(5) If all of the K rows from $\mathbf{U}$ and $\mathbf{U}^{ref}$ are mapped, the algorithm terminates, otherwise it goes back to step (2) with the reduced similarity matrix.

2.3. CoPaM Generation

The objective of this step is to relabel the clusters of all of the partition matrices to be aligned, then to find the element by element mean of all of them to generate the CoPaM $\mathbf{U}^*$. In order to relabel the clusters in one of the partitions, there should be a reference partition with which the comparisons are held. One suggestion is to consider the first partition as the reference and relabel all of the others according to it. In [8], it is suggested that an intermediate fuzzy CoPaM $\mathbf{U}^{int(k)}$ is initialized with the values of the first partition $\mathbf{U}^1$, and then the other partitions are relabelled and fused with this intermediate matrix one by one while considering it as the reference at each step. This later suggestion is the one which is considered in this paper.

Mathematically, let $\hat{\mathbf{U}}^r$ be the relabelled partition matrix of the partition $\mathbf{U}^r$ and let $\mathbf{U}^{int(k)}$ be the intermediate partition matrix after the k-th stage, i.e. after relabelling and fusing the partitions $\{\mathbf{U}^1, \cdots, \mathbf{U}^k\}$. Let the function Relabel$(\mathbf{U}, \mathbf{U}^{ref})$ denote relabelling the partition matrix U by considering $\mathbf{U}^{ref}$ as the reference partition. Equation (2) shows how the intermediate partition matrix can be calculated by the normal approach and the recursive approach:

$$\mathbf{U}^{int(k)} = \frac{1}{k}\sum_{r=1}^{k}\hat{\mathbf{U}}^r = \frac{1}{k}\hat{\mathbf{U}}^k + \frac{(k-1)}{k}\mathbf{U}^{int(k-1)}. \quad (2)$$

Generating the fuzzy CoPaM is achieved by following the algorithm shown in the following steps:

```
(1) $\mathbf{U}^{int(1)} = \mathbf{U}^1$
(2) For $k = 2$ to $R$
    a. $\hat{\mathbf{U}}^k = $ Relabel$(\mathbf{U}^k, \mathbf{U}^{int(k-1)})$
    b. $\mathbf{U}^{int(k)} = \frac{1}{k}\hat{\mathbf{U}}^k + \frac{k-1}{k}\mathbf{U}^{int(k-1)}$
(3) $\mathbf{U}^* = \mathbf{U}^{int(R)}$
```

This final CoPaM $\mathbf{U}^*$ is then passed to the binarization step which is discussed in the next section.

2.4. Binarization

This step is the primary novelty of the proposed novel method. The information which is provided by the final CoPaM $\mathbf{U}^*$ is used to generate a CBP $\mathbf{B}^*$. The CBP $\mathbf{B}^*$ is a pseudo-partition matrix with K rows for the clusters and M columns for the objects, where each column contains either unique 1 or multiple 1s or no 1 at all, representing that the object belongs to either unique cluster or multiple clusters or none cluster at all, respectively. The properties of $\mathbf{B}^*$ is given by

$$\begin{aligned}
&(1) \qquad b^*_{i,j} \in \{0,1\}, \quad 1 \le i \le K, 1 \le j \le M,\\
&(2) \qquad 0 \le \textstyle\sum_{i=1}^{K} b^*_{i,j} \le K, \quad 1 \le j \le M,\\
&(3) \qquad 0 \le \textstyle\sum_{j=1}^{M} b^*_{i,j} \le M, \quad 1 \le i \le K.
\end{aligned}$$

Different binarization techniques are proposed in the following part to consider different objects of view of the clustering results. There are two measurements monitoring the status of the binarization techniques: one is M^{multi} indicating the number of objects belonging to more than one cluster, and the other is M^{un} indicating the number of objects belonging to none of the clusters.

2.4.1. Binarization Techniques

There will be five binarization techniques presented as follows:

Intersection Binarization (IB): This is the strictest binarization technique where an object is assigned to a cluster if all of the partitions map this object to that cluster. Mathematically, this is formulated as:

$$b^*_{i,j} = \begin{cases} 1, & u^*_{i,j} = 1 \\ 0, & otherwise \end{cases} \quad (3)$$

This technique results in $M^{multi} = 0$ and $M^{un} \ge 0$.

Maximum Value Binarization (MVB): Each object is assigned to the cluster to which its maximum membership value objects. If more than one cluster share the same maximum value, it is assigned to all of them. Its mathematical formula is:

$$b^*_{i,j} = \begin{cases} 1, & u^*_{i,j} = \max_{1\le k\le K} u^*_{k,j} \\ 0, & otherwise \end{cases} \quad (4)$$

This technique results in $M^{multi} \ge 0$ and $M^{un} = 0$. The value of M^{multi} is usually very small and might reach 0.

Value Thresholding Binarization (α-cut) (VTB-α): Each object is assigned to all of the clusters in which its membership values are not less than a threshold α. The mathematical formula is:

$$b^*_{i,j} = \begin{cases} 1, & u^*_{i,j} \ge \alpha \\ 0, & otherwise \end{cases} \quad (5)$$

This technique results in $M^{multi} = 0$ for $\alpha > 0.5$ and $M^{multi} \ge 0$ for $\alpha \le 0.5$. In general, $M^{un} \ge 0$ and its value increases when the value of α increases.

Top Binarization (TB): This is a relaxed version of the maximum value binarization technique such that each object is assigned to the maximum membership value cluster and to all of the clusters in which its membership values are within a certain difference (δ) bellow the maximum. The mathematical formula is:

$$b^*_{i,j} = \begin{cases} 1, & u^*_{i,j} \ge \max_{1\le k\le K}\left[u^*_{i,j}\right] - \delta \\ 0, & otherwise \end{cases} \quad (6)$$

This technique results in $M^{multi} \ge 0$ and $M^{multi} = 0$. The value of M^{multi} is larger than that of the maximum binarization technique and increases as the value of δ increases.

Difference Thresholding Binarization (DTB): This is a stricter version of the maximum value binarization technique such that each object is assigned to the maximum membership value cluster only if the value of the closest competitor cluster is at least as far from the maximum as a predefined difference (δ). The mathematical formula is:

$$b^*_{i,j} = \begin{cases} 1, & u^*_{i,j} - u^*_{k,j} \ge \delta, 1 \le k \le K, k \ne i \\ 0, & otherwise \end{cases} \quad (7)$$

For $\delta > 0$, this technique results in $M^{multi} = 0$ and $M^{un} \ge 0$. The value M^{un} increases as the value of δ increases.

2.4.2. *Binarization Techniques Discussion*

Different binarization techniques may suite different applications, and even for the same application, different binarization techniques may reveal different pieces of information by viewing the data from different points of view. For example, using the IB, the DTB or the VTB with high values of α can find the tightest clusters. This gives an idea about the objects that are relatively clearly assigned to one cluster. Using the VTB with small values of α or the TB with high values of δ may result in plump clusters which contain these objects that are somehow likely to be in. This can be useful in identifying the objects that play rules in different clusters and different applications may find different meanings for this. For instance, in gene clustering, this mean that these genes are participating in different biological functional groups. The MVB and TB with small values of δ are good to find the most reasonable final clustering results with a few multi assignments. The value of δ can be tuned for the best results.

3. DATA, RESULTS AND ANALYSIS

3.1. Data

An artificial microarray dataset is studied in this paper which is a synthetic periodic gene expression dataset. It consists of 450 synthetic gene expressions over 24 time points. The expressions were synthesized to show periodic sinusoidal patterns that cover two complete cycles over the time given time points. The 450 genes belong to 5 different groups that are characterized by phase shift values of their patterns. The method of the synthesis was proposed in [17] and used in many researches [18]. For this paper, the synthetic parameters that control the noise in the data were tuned to result in a noisy dataset with an SNR value of about 2.9 dB.

3.2. Experiment Setup

Four clustering methods with different configurations were applied over the dataset to generate sets of clustering partitions. Then the CoPaM was generated which was then binarized using different binarization techniques to generate the final results which were analyzed. Table 1 lists the details of the applied clustering experiments.

No.	Method	Parameters
1	k-means	Empty clusters were dropped and Kaufman deterministic initialization [2] was used.
2	SOMs	Batch mode learning, 2D hexagonal grids and bubble neighbourhood (Repeated 20 times then combined)
3	SOMs	Batch mode learning, 2D hexagonal grids and Gaussian neighbourhood. (Repeated 20 times then combined)
4	HC	Single linkage
5	HC	Complete linkage
6	HC	Average linkage
7	HC	Centroid linkage
8	HC	Ward linkage
9	HC	Median linkage
10	SOON	$b \in \{0.1, 0.5, 1, 10, 20, 40, 60, 80, 100, 120\}$ $C_E \in \{0.1, 0.12, 0.14, 0.16, 0.18, 0.2, 0.3, 0.4\}$ $d0 \in [3.6: 0.2: 6.0]$ where b is the constant which controls the concavity of the mapping function, C_E is the constant of excitation and d0 is the radius of the clusters. For each one of the datasets, the results of using all of the possible parameters' combinations were combined into one partition which was given ten times the weight of the other experiments in the final fuzzy CoPaM.

Table 1. Clustering experiments setup.

No.	Technique	Param. (α or δ)	Clusters					M^{un}	M^{multi}
			C1	C2	C3	C4	C5		
1	IB	-	0	0	0	0	0	450	0
2	UB	-	329	371	385	292	372	0	450
3	MVB	-	90	98	90	81	91	0	0
4	TB	0.05	90	100	90	83	91	0	4
5	TB	0.1	90	101	90	87	92	0	10
6	TB	0.2	92	103	90	90	92	0	17
7	TB	0.4	106	118	92	91	99	0	50
8	VTB	0.3	102	109	91	91	98	0	41
9	VTB	0.5	89	96	90	78	91	6	0
10	VTB	0.7	76	89	87	50	77	71	0
11	VTB	0.8	12	85	81	1	61	210	0
12	VTB	0.9	0	79	0	0	15	356	0
13	DTB	0.1	89	93	89	78	91	10	0
14	DTB	0.3	86	90	88	70	80	36	0
15	DTB	0.6	19	86	82	15	67	181	0
16	DTB	0.8	0	80	0	0	36	334	0

Table 2. Synthetic data set Bi-CoPaM results.

3.3. Results and Analysis

Table 2 shows the results of the binarization techniques when applied over the synthetic dataset. The first column of each of the tables shows the sequential index of the binarization experiment, the second and the third columns show the name and the parameter value of the employed binarization technique, the next set of columns show the numbers of the genes assigned to each of the clusters and the last two columns show the numbers of the unassigned and the multi-assigned genes respectively. As this data set was synthesized, it should have five groups of genes, each of them consists of 90 genes that show periodic patterns with the same phase shift, i.e. peaking at the same time points.

The MVB technique results in the closest partitioning result to crisp clustering with the minimum numbers of multi-assigned and unassigned genes. Its results here are close to the aforementioned suggestion. From the figures in Table 2, it can be seen how the clusters shrink in the DTB technique's results while they are widened and overlap in the TB technique's results. The researcher might be interested in getting the core genes of the cluster (1) by getting the 12 genes that are assigned to it in the VTB technique with $\alpha = 0.8$, or he might be interested in the extended version of cluster (1) to study all of the genes that have some potential in being in this cluster by getting the 106 genes assigned to this cluster by TB with $\delta = 0.8$. Another observation is that the second cluster is clearly purer than the other four ones as it maintains most of its genes even in the extreme tightening conditions except

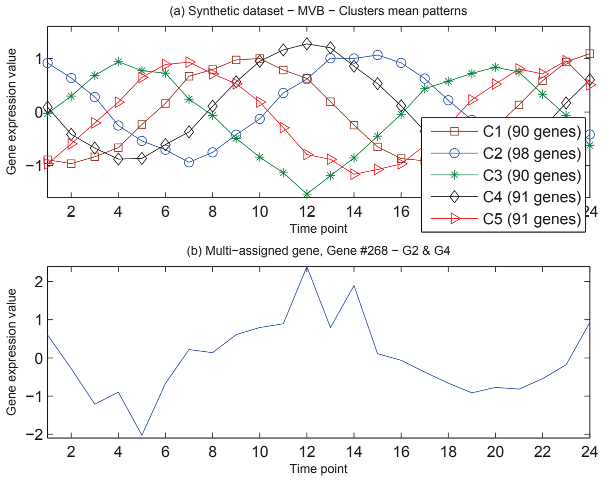

Fig. 1. Clustering results of synthetic dataset. (a) Synthetic clusters mean patterns (b) Multiple assigned gene: the gene ♯268 which is assigned to C2 and C4 simultaneously in TB with δ=0.05.

for the IB. The fourth cluster is the noisiest as it tends to lose its genes faster than the others.

Fig. 1 (a) shows the mean patterns for the five clusters from the MVB technique. It is clear that all of them show periodic patterns with different phase shifts for different clusters as expected. By slightly widening the clusters using TB with $\delta = 0.05$, four genes are multi-assigned. One of these genes, namely, the 268-th gene out of 450, is assigned to both clusters (2) and (4). This gene is plotted in Fig. 1 (b) and its high peak at the time point twelve and the slightly lower peak at the time point nine clearly justify its membership in both cluster one and cluster four when compared with the mean patterns in Fig. 1 (a) and it is clear that it has two peaks at the 12-th and the 14-th time points which correspond to the peaking time of the mean expressions of these two clusters.

4. CONCLUSIONS

A novel method of ensemble clustering, Bi-CoPaM, was proposed in this paper to tackle the clustering problem in a new sense. This new method exploits the information provided by many partitions generated by different clustering experiments. Bi-CoPaM combines these partitions to generate one CoPaM which is then binarized using one of the six proposed binarization techniques to generate final binary partitions. The proposed binarization techniques relax the conventional clustering conditions by allowing genes to be assigned to multiple clusters at the same time or not to be assigned to any of the clusters at all. Different binarization techniques reveal different pieces of information from the CoPaM such as the relative tightness of the clusters, the clusters' tight cores, the wide overlapping clusters, and the special genes that are either multi-assigned or unassigned. A synthetic periodic gene dataset was analyzed by this method and the numerical results showed that the method has been successful in showing different horizons in gene clustering.

5. REFERENCES

[1] T. E. Kohonen, *Self-Organizing Maps,*, New York: Springer-Verlag, 1997.

[2] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods fro the k-means algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

[3] 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. (PNAS)*, vol. 95, pp. 14863–14868, 1998.

[4] M. B. H. Rhouma and H. Frigui, "Self-organization of pulse-coupled oscillators with application to clustering," *IEEE Trans. Pattern Anal. Mach. Intell.,*, vol. 23, no. 2, pp. 1–16, 2001.

[5] Sameh A. Salem, Lindsay B. Jack, and Asoke K. Nandi, "Investigation of self-organizing oscillator networks for use in clustering microarray data," *IEEE Trans. on NanoBio.*, vol. 7, no. 1, pp. 65–79, 2008.

[6] R. Baumgartner, C. Windischberger, and E. Moser, "Quantification in functional magnetic resonance imaging: fuzzy clustering vs. correlation analysis," *Magnetic Resonance Imaging*, vol. 16, no. 2, pp. 115–125, 1998.

[7] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.

[8] A. Weingessel, E. Dimitriadou, and K. Hornik, "An ensemble method for clustering," in *DSC 2003 Working Papers*, 2003.

[9] H. G. Ayad and M. S. Kamel, "On voting-based consensus of cluster ensembles," *Pattern Recognition*, vol. 43, pp. 1943–1953, 2010.

[10] A. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Proceedings of the Sixteenth International Conference on Pattern Recognition (ICPR)*, 2002, pp. 276–280.

[11] A. Strehl and J. Ghosh, "Cluster ensembles C a knowledge reuse framework for combining multiple partitions',," *Journal of Machine Learning Research,*, vol. 3, pp. 583–617, 2002.

[12] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted cluster ensemble using a kernel consensus function," *Lecture Notes in Computer Science, Heidelberg: Springer.*, vol. 5197, pp. 195–202, 2008.

[13] V. Filkov and S. Skiena, "Integrating microarray data by consensus clustering," in *Proc. of International Conference on Tools with Artificial Intelligence*, 2003.

[14] W. Wang, "An improved non-negative matrix factorization algorithm for combining multiple clusterings," in *2010 International Conference on Machine Vision and Human-machine Interface*, 2010, pp. 604–607.

[15] R. Avogadri and G. Valentini, "Ensemble clustering with a fuzzy approach," Tech. Rep., in Supervised and Unsupervised Ensemble Methods and their Applications Studies in Computational Intelligence, 2008.

[16] Basel Abu-Jamous, "The Analysis of Microarray Data,", M.Sc. thesis (supervised br Professor Asoke K. Nandi), Department of Electrical Engineering and Electronics, The University of Liverpool, 2010-2011. http://liverpool.academia.edu/BaselAbuJamous/Papers/955138/Microarray_Data_Analysis_-_MSc_Thesis

[17] L. P. Zhao, R. Presntice, and L. Breeden, "Statistical modelling of large microarray data sets to identify stimulus-response profiles," *Proc. Natl. Acad. Sci. (PNAS)*, vol. 98, no. 10, pp. 5631–5636, 2001.

[18] R. Fa and A. K. Nandi, "Comparisons of validation criteria for clustering algorithms in microarray gene expression data analysis," in *Proceedings of the second International Workshop on Genomic Signal Processing (GSP2011)*, 2011.