Available online at www.ijpab.com

DOI: http://dx.doi.org/10.18782/2320-7051.6308

ISSN: 2320 – 7051

Int. J. Pure App. Biosci. **6 (2):** 37-46 (2018)

Research Article

International Journal of Pure & Applied Bioscience

OPEN ACCESS

ExBIRCH: Scalable Non-Centroid BIRCH-like Algorithm for Clustering Gene Expression Data based on Average Correlation

Mohamed A. Mahfouz*

*PhD, Department of Computer and Systems Engineering, Faculty of Engineering, Alexandria University, Egypt

*Corresponding Author E-mail: m.a.mahfouz@gmail.com

Received: 5.03.2018 | Revised: 10.04.2018 | Accepted: 14.04.2018

ABSTRACT

Clustering is the main step in gene expression analysis. BIRCH algorithm is able to efficiently, incrementally and dynamically cluster data points. However original BIRCH algorithm is limited to the Euclidean distance measure. Euclidian distance is not suitable for gene expression clustering because it is sensitive to scaling and differences in average expression level while correlation is not. This paper proposes an extended BIRCH algorithm (ExBIRCH) based on average Pearson correlation on normalized gene expression dataset. The adaptive possibilistic clustering algorithm is directly applied to the produced sub-clusters represented by their CF vectors. The proposed algorithm inherits the ability of BIRCH to provide a compact model representation. Several clustering algorithms can be applied on leaf nodes of the output tree similar to BIRCH however the proposed possibilistic paradigm has a high rejection to outliers and is able to deal with existing overlapping between clusters. Also, the use of average correlation instead of cluster center, helps discovering non-convex shaped clusters. Experimental study shows that the proposed algorithm is able to generate higher quality clusters in terms of three assessment measures compared to existing algorithms for clustering gene expression data.

Key words: *Bioinformatics, Gene expression analysis, Hierarchical Clustering, Possibilistic*

INTRODUCTION

Clustering analysis is a major technique in exploratory data analysis. Data to be clustered usually represented as objects and a characteristic vector for each object. A measure of similarity (or dissimilarity) is defined between pairs of such vectors. The goal is to partition the objects into subsets, which are called clusters, so that two criteria are satisfied: compactness – objects in the same cluster are highly similar to each other; and separation - elements from different clusters have low similarity to each other. In the scope of gene expression data, elements are usually genes, the vector of each gene contains its expression levels under each of the monitored conditions or experiment. Thus, applying cluster analysis techniques to gene expression data may highlight groups of functionally related genes.

Cite this article: Mahfouz, M.A., ExBIRCH: Scalable Non-Centroid BIRCH-like Algorithm for Clustering Gene Expression Data based on Average Correlation, *Int. J. Pure App. Biosci.* **6(2):** 37-46 (2018). doi: http://dx.doi.org/10.18782/2320-7051.6308

Hierarchical clustering algorithms (HCA) work either in a top-down manner, by repeatedly partitioning the set of elements (divisive) such as Divisive Analysis (DIANA)[1], or in a bottom-up fashion (agglomerative) by repeatedly merging the set of elements such as Agglomerative nesting (AGNES)[1] In each iteration, a linkage strategy[2] is applied to the newly formed clusters, leading to a new dissimilarity matrix. The output of HCA is typically represented by a dendrogram. In[3] clustering software package along with visualization program based on the average-linkage hierarchical clustering algorithm has been developed. Traditional HCA suffer from the defect that they cannot recover bad decisions that were done in previous steps (i.e., the inability to reunite or split whatever it already divided or agglomerated).

Hybridized K-means[4] was heavily used in the scope of gene expression data. It is simple, fast and avoids the arbitrary choice of cluster centroids in traditional k-means. In[5] the benefit of intelligent K-means[6] and kernel K-means[7] are incorporated to develop an extension of K-means clustering algorithm termed as Intelligent Kernel K-means (IKKM). IKKM is successfully applied to gene expression clustering and was able to overcome the drawbacks of traditional K-means algorithm.

Medoid-based algorithms such as PAM and CLARA or CLARANS[1] are examples of clustering algorithms that are applicable to relational data however gene expression data is not relational data. A cluster is represented by the most centrally located object (instead of cluster centroids). They suffer from the same drawbacks of centroids-based algorithm except they are more robust.

CURE clustering algorithm[8] adopts a compromise between centroid-based and all-point extreme approaches by using multi-representatives instead of using single representative as in CLARANS for example. It was applied successfully to gene expression data. It is less sensitive to outliers but it can deal with large datasets efficiently using random sampling or partitioning.

CLICK is grid-based clustering algorithm that seeks to identify highly connected components in the proximity graph as clusters. The existence of noise and overlapping in gene expression data may force CLICK to merge or split clusters unnecessary.

Prototype-Based Modified DBSCAN (MDB-SCAN)[9] extends the traditional density-based algorithm DBSCAN using partitioning to reduce its complexity. Both artificial and biological datasets were used to evaluate the performance of MDBSCAN. The results show that MDB-SCAN is more efficient than DBSCAN, insensitive to the selection of initial prototypes or noise, and it is still able to produce the clusters of arbitrary shapes as DBSCAN however the quality of produced clusters degrade with large number of clusters. Self-Organizing Maps (SOM) is widely used as a clustering technique for gene expression data[10]. The major drawback of SOM is that the number of clusters should be known in advance. HDP is a model-based algorithm[11] that ensures the robustness to the problem of different choices of the number of clusters by using the infinite mixture model along with HCA. In[12] multi-objectives genetic clustering is used to deal with multiple overlaps among clusters in gene expression data. Fuzzy clustering by Local Approximation of Membership (FLAME) in[13] outperforms K-means, HCA, fuzzy C-means, and fuzzy SOMs. It shows high rejection capability to outliers. Dual-rooted MST along with spectral clustering is used in[14] to allow discovering non-convex shaped clusters.

In this research study we try to tackle several problems that exist in clustering algorithms for gene expression data such as inefficiency, sensitivity to outliers, bias to spherically shaped clusters and the inability to deal with existing highly overlapped clusters. This paper proposes a new version of BIRCH algorithm termed as ExBIRCH that is based on average correlation. The possibilistic paradigm is applied to the leaf entries of the output tree of ExBIRCH. The proposed approach inherits

the efficiency of BIRCH in terms of memory and computational complexity. Also, it reduces its biasedness to spherically shaped clusters by avoiding the use of mean square objective function. Furthermore, the possibilistic paradigm helps dealing with expected high overlap between highly connected clusters, removing outliers. Also, unlike Euclidian distance, the Pearson correlation measure is directly applicable to high-dimensional datasets.

The remainder of this paper is organized as follows. Section 2 presents related work and necessary background for the proposed technique. Section 3 presents the proposed technique. In section 4, experimental results are discussed. Finally Section 5 concludes the paper and highlights future research directions.

Related Work

BIRCH Algorithm[15]

BIRCH algorithm[15] introduced two concepts for briefly and efficiently summarizing data objects: clustering feature (CF) and clustering feature tree (CF Tree). Instead of storing all Metadata of multiple data objects a CF tree is able to summarizes the valuable information of incoming data objects by using fixed and much smaller space. In BIRCH tree a CF is a node element that summarizes data objects which are close enough and should appear in one group in the final clustering. CF is stored as a vector of three values: CF = (N; $\vec{LS}$; SS), where N is the number of data objects it represents, $\vec{LS}$ and SS are the linear sum and the square sum of the enclosed data objects, as follows:

$$CF = (N; \vec{LS}; SS) = (N, \sum_{i=1}^{N} \vec{X}_i, \sum_{i=1}^{N} \vec{X}_i^2) \quad (1)$$

Using the CF vectors of produced sub clusters, it has been shown that common quality metrics can be easily calculated. In addition, the cluster of two disjoint clusters could be easily calculated by merging the two CF vectors following the Additivity Theorem, as shown below:

$$CF_1 + CF_2 = (N_1 + N_2; \vec{LS}_1 + \vec{LS}_2; SS_1 + SS_2) \quad (2)$$

The input to the algorithm is a set of N data objects and the output is hierarchy of the input data. Root node represents the whole data as one cluster while nodes in next levels correspond to smaller sub clusters. Incoming data objects are checked against existing nodes starting from root to leaf nodes and inserted in the proper CF node incrementally according to two thresholds. It has four phases, the first and third phase are the mandatory phases while the others are optional. In the first phase, data objects are inserted into the CF tree, which is characterized by two thresholds: branching factor B, which indicates the maximum number of nodes represented by a non-leaf node, and threshold T, which controls the size of the leaf-node. The insertion operation is done in a three steps:

1. Starting from the root, recursively descend the CF-tree to find the appropriate leaf node.
2. If the insertion on closest CF leaf violates the constraint on the size of a leaf node T, a new CF entry will be made. If there is no room for a new leaf to be established, the parent node will be split.
3. Apply the insertion on the path from the selected leaf node back to the root.

In the rebuilding phase (the second phase), the leaf entries in the initial CF tree are scanned and a smaller CF tree will be rebuilt by removing outliers and clustering crowded sub-clusters into a large cluster, which will need to generate a larger threshold T. In the third

phase, any clustering algorithm can be directly applied to the produced sub-clusters represented by their CF vectors to produce the required number of clusters. As shown above, the original data need to be scanned only once in the first phase, which means the whole dataset does not have to reside in memory to perform BIRCH clustering and this feature makes BIRCH efficiently applicable to gene expression clustering. In the fourth phase, data objects are redistributed to its closest cluster of the clusters produced in previous phase, so that a set of new clusters is obtained.

Possibilistic Clustering

The objective function to be minimized in possibilistic c-means (PCM) [16] is as follows:

$$J_m(U) = \sum_{p=1}^{k}\sum_{i=1}^{n} u_{pi}\, d(x_i, \theta_p) + \beta_p \sum_{i=1}^{n}(u_{pi}\ln u_{pi} - u_{pi}) \qquad (3)$$

Where u_{pi} and θ_p are the possibilistic membership of an object x_i in cluster p and the center of cluster p respectively. β_p is an input parameter estimated using the average size of the *p-th* initial cluster and stay constant in all iteration of PCM. *k* is the required number of clusters. *n* is the number of objects to be clustered.

The possibilistic C-means algorithm (PCM) was proposed to tackle the major drawbacks associated with the constrained memberships used in algorithms such as the fuzzy C-means (FCM). In Possibilistic clustering, the constraints on the elements of the membership matrix U are relaxed to:

1) $u_{pi} \in [0,1] \quad \forall\, p,I,$ 2) $0 < \sum_{i=1}^{n} u_{pi} < n \quad \forall\, p$ and 3) $\bigvee_{p} u_{pi} > 0 \quad \forall i$

Recent approach in[17] uses an adaptive approach to dynamically update β_p. It starts with overestimated number of clusters and adaptively reduce the number of clusters by removing a cluster if it doesn't have an object that has a maximum membership in it. This algorithm is modified to allow applying the possibilistic clustering using the average correlation instead of centers as explained in section 2.4.

MATERIAL AND METHODS

In order to develop an accurate clustering algorithm for large gene expression datasets, an extended BIRCH algorithm termed ExBIRCH is proposed with a modified CF vector and distance measure other than Euclidian distance. Meanwhile, to address the problem of computing the average correlation between a gene and a set of genes, the gene expression data matrix needs to be normalized. Also the CF vector is modified to fit the extended BIRCH algorithm (ExBIRCH). Finally, we discuss the clustering phase in Section 3.4.

Similarity Measure

The most common similarity measures in the field of bioinformatics are similarity measures related to correlation coefficient. Most of them have corresponding dissimilarity measures. The Pearson's correlation coefficient of two random variables *x* and *y* is formally defined as follows:

$$s(x,y) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right) \qquad (4)$$

Where $\bar{x}, \bar{y}$ are the sample mean of *x* and *y* respectively, while σ_x, σ_y are the sample standard deviation of *x* and *y*. It is a measure of how well a straight line can be fitted to a scatter plot of *x* and *y*. Pearson correlation coefficient falls in [-1, 1].

The corresponding dissimilarity measure d(*x*,*y*) equals 0.5(1- s(*x*,*y*)) and it fall in[1]. The threshold T for ExBIRCH can be set on the s or d.

When Euclidian distance needs to be used as dissimilarity measure for gene expression data the input matrix need to be normalized in order to reduce its sensitivity to scaling and differences in average expression level. The normalization is done by subtracting the mean of the values of each row representing a gene expression values (feature vector) from the values of this row then the resulting new entries are divided by the standard deviation of its corresponding row. For input matrix X, each entry x_{ij} is replaced by $(x_{ij} - \overline{x_i}) / \sigma_{x_i}$

When input gene expression matrix is normalized, the computations required for computing gene-gene correlations is reduced as follows:

$$s(x, y) = \frac{1}{n} \sum_{i=1}^{n} x_i \, y_i \tag{5}$$

Proposed Extended BIRCH (ExBIRCH)

Several research studies showed that the use of Pearson correlation coefficient as a similarity measure gives a higher performance compared to Euclidian distance. A very interesting property for normalized matrix is that the computation of average correlation between a feature vector *x* and a group of feature vectors is reduced to computing the correlation between *x* and the vector representing the average of this group. For example, the correlation between a feature vector *x* and another two feature vectors *y* and *z* can be computed as follows:

$$average\ correllation = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n} x_i \, y_i + \frac{1}{n}\sum_{i=1}^{n} x_i \, z_i\right) = \sum_{i=1}^{n} x_i \left(\frac{1}{2}(y_i + z_i)\right) \tag{6}$$

Using this property, the CF vector in BIRCH clustering algorithm can be to two values:

CF = (*N*, $\vec{LS}$), where *N* means the number of data objects, $\vec{LS}$ represent the linear sum of these data objects.

We will show how the distance between two sub-clusters of two CF vectors represented this way. Also other common quality metrics can be computed using the stored CF vectors.

$$CF = (N; \vec{LS}) = (N, \sum_{i=1}^{N} \vec{X}_i) \tag{7}$$

$$CF_1 + CF_2 = (N_1 + N_2; \vec{LS}_1 + \vec{LS}_2) \tag{8}$$

The average correlation between a feature vector of a gene represented by $\vec{X}_j$ and a sub cluster represented by a vector CF is computed as follows:

$$s(\vec{X}_j, CF) = \frac{1}{N} \sum_{i=1}^{N} \vec{X}_j . \vec{X}_i . = \frac{1}{N} \vec{X}_j . \sum_{i=1}^{N} \vec{X}_i . = \frac{1}{N} \vec{X}_j \, \vec{LS} \tag{9}$$

Using Eq. (9), the average correlation between two sub clusters represented by CF_1 and CF_2 is computed as follows:

$$s(CF_1, CF_2) = \frac{1}{N_1} \sum_{X_j \in CF_1} s(\vec{X}_j . CF_2) = \frac{1}{N_1 N_2} \sum_{X_j \in CF_1}^{N} \vec{X}_j \, \vec{LS}_2 = \frac{1}{N_1 N_2} \vec{LS}_1 . \vec{LS}_2 \tag{10}$$

Estimating appropriate value for the Threshold T

Appropriate value for the threshold T is estimated by analysing the pair-wise correlation between genes. For each gene, the average correlation between it and a suitable number of nearest genes (close to the accepted number of objects in a leaf node) is computed. A cutting point for estimating the threshold T is identified by sorting the genes according to its average correlation with their nearest neighbours. Then, the gene which has much lower average correlation compared to its predecessor is identified. The average correlation of the predecessor of such gene is selected as the minimum average similarity between genes in a sub cluster. When T corresponds to average similarity then it is a lower bound for the average similarity of objects (gene) within a leaf node.

Clustering Leaf Nodes

Using Eq.[10] algorithm such as Hierarchical clustering based on average linkage strategy can be directly applied to leaf nodes. However iterative approach is more efficient and can recover bad decisions iteratively. We chose the possibilistic approach as it is more robust. In this section we show how the adaptive approach for PCM[17] can be modified to be applicable for our non-centroid case. In order to use the adaptive approach in[17] the objective function in Eq[3]. should be modified to use the average similarity instead of distance to a center. An artificial center $\vec{Z}_p$ representing a cluster C_p of a set of leaf nodes (sub clusters) is computed as follows:

$$\vec{Z}_p = \frac{\sum_{i=1}^{n} u_{pi}(\vec{LS}_i / N_i)}{\sum_{i=1}^{n} u_{pi}} \quad (11)$$

Where *n* is the number of leaf nodes to be clustered and N_i is the number of objects represented by a leaf node number *i*. The memberships can be computed based on Eq.[11]. The following is a Possibilistic c-means (PCM) like updating equations. The memberships are updated in iteration *t* as follows:

$$u_{pi}(t) = e^{-\alpha E_{pi}(t)/(\beta_p(t)\beta_{\min})} \qquad \forall\ p,i \quad (12)$$

Where

$$E_{pi} = 0.5(1 - s(x_i, .\vec{Z}_p)) = 0.5(1 - \frac{1}{N_i}\vec{LS}_i .\vec{Z}_p)$$

And β_p is computed for each initial cluster *p* from the initial hard membership matrix U using an initialization procedure in step 1 of Table 1, as follows:

$$\beta_p = \frac{\sum_{x_i \in C_p} E_{pi}}{n_p} \quad for\ p = 1,2,..k \quad (13)$$

where n_p is the size of the hard cluster *p* produced in the initialization step.
In each iteration t, the values of $\beta_p(t)$ are updated as follows:

$$\beta_p(t) = \frac{\sum_{x_i : u_{pi} = \max_{r=1}^{k} u_{ri}} u_{pi} E_{pi}}{\sum_{x_i : u_{pi} = \max_{r=1}^{k} u_{ri}} u_{pi}} \quad for\ p = 1,2,..k \quad (14)$$

In Eq. (14), the memberships are included without defuzzification. The dominator represents the number of clusters which have at least one object with its greatest membership in it. The parameter α falls in]0,∞[and should be fine-tuned.

Table 1: Adaptive possibilistic clustering for the leaf nodes of ExBIRCH

Input:

α */* parameter in eq. (16) for computing memberships */*

ε */* the threshold controlling the convergence*/*

k */* the number of clusters to be found*/*

Output:

U */* matrix of possibilistic memberships*/*

Begin

1. *Compute initial hard membership matrix U using initialization procedure*

2. initialize β_p for *p*=1,2..*k* using Eq. (12)

3. Compute artificial cluster center Z_p for *p*=1,2..*k* using Eq. (11)

4. set $\beta_{min} = min\ \beta_p$

5. Repeat

Store memberships *U* in U'

Update U using Eq. (12)

Update artificial cluster center Z_p using Eq. (11)

Update β_p using Eq. (14)

until $(\| U - U' \| < \varepsilon)$

6. Output *U*

End

RESULTS AND DISCUSSION

Datasets

The following two datasets are subjected to a set of algorithms in[18] including k-means, Self Organizing Maps (SOM), Hierarchical average-link clustering, CLICK, Self Organizing Trees (SOTA) [19] and αCORR. The reported results in[18] along with the results of the proposed clustering algorithm are presented in this paper. For each of the chosen data sets, 25 runs were executed to get a reliable average measure of the validation indices for the proposed algorithms at various numbers of clusters. Several experiments are done for fine tuning the input parameter T for each dataset as explained in section 3.3. When the number of clusters *k* is not very larger than the actual number of clusters, appropriate values for the parameter α are around 1[17].

Table 1: List of Datasets used in our experimental study

Dataset	Ref.	no. objects	no.	True Clusters
Yeast Cell Cycle (YCC)	[3]	698	72	5
Peripheral Blood Monocytes (PBM)	[20]	2329	139	18

Performance measures

In the following performance analysis of the proposed algorithm, the Dunn Index[21] Silhouette width[22] and Adjusted Rand Index[23] and are used as assessment measures. Dunn Index is a non-linear measure that combines the ratio between the largest cluster similarity and the smallest intra-cluster similarity in a cluster. The Dun Index lies on [0, +∞] and should be maximized.

The Silhouette Width for a cluster is computed as the average silhouette value over all input data. The silhouette value of an object *i* (s_i) combines the average distance between an object *i* and all data items in the same cluster and the average distance between *i* and all data items in the closest other cluster. s_i lies on[1]. When s_i is close to 1, it indicates that the *i*-th sample x_i has been well clustered.

Adjusted Rand Index is a normalized external measure that takes a single clustering result as an input, and compares it with a known set of class labels to assess the degree of similarity between them. It is limited to the interval[1] and should be maximized.

Experimental Results

Table 2 shows the values of each validity index over all the algorithms under comparison on YCC dataset. As we can notice, the proposed algorithm outperformed all other algorithms for the adjusted rand index, scoring more than 69% matching with the true classification of the YCC dataset. For Dunn's Index, the proposed algorithm is slightly less than α*CORR*, the place for top performance under this measure. Finally, the silhouette width shows that ExBIRCH was the best performing algorithm. Its higher silhouette width may account to the match of its objective function to the definition of silhouette width.

Table 2: Experimental results of the studied algorithms at the true number of clusters (Five) of the true solution of YCC dataset

Algorithm	*Adjusted rand Index*	*Dunn Index*	*Silhouette Width*
k-means	0.499	0.212	0.280
Average Link	0.496	0.255	0.308
CLICK	0.537	0.246	0.280
SOM	0.484	0.202	0.239
SOTA	0.496	0.131	0.253
αCORR	0.621	0.302	0.413
ExBIRCH	0.692	0.269	0.472

Table 3 shows the values of each validity index over all the algorithms under comparison on PBM dataset. As we can notice, the proposed algorithm outperformed all other algorithms for both the adjusted rand index and silhouette width. Also for Dunn's Index, the proposed algorithm is slightly less than α*CORR*, the place for top performance under this measure. The silhouette width results are relatively much higher than the case for YCC dataset.

Table 3: Experimental results of the studied algorithms at the true number of clusters (Eighteen) of the true solution of PBM dataset

Algorithm	*Adjusted rand Index*	*Dunn Index*	*Silhouette Width*
k-means	0.431	0.180	0.132
Average Link	0.449	0.117	0.184
CLICK	0.448	0.117	0.125
SOM	0.386	0.092	0.077
SOTA	0.438	0.152	0.190
aCORR	0.469	0.212	0.202
ExBIRCH	0.471	0.201	0.388

Finally, Table 4 shows the performance of the proposed algorithm compared to the other algorithms on PBM Dataset for different number of clusters. The proposed algorithm has higher and more stable performance over large number of clusters ranging from 9 to 24. This may account to that ExBIRCH breaks the dataset into smaller, more coherent units of correlated functionality within each leaf node, without any distortion or changing gene membership across the boundaries of the clusters constituting of the true solution.

Table 4: Silhouette Width for different number of clusters on PBM Dataset

Algorithm	2	3	4	6	9	12	15	18	21	24
k-means	0.3	0.32	0.321	0.199	0.175	0.161	0.140	0.132	0.130	0.137
Average Link	0.3	0.265	0.288	0.310	0.261	0.216	0.191	0.184	0.168	0.183
CLICK	0.3	0.295	0.285	0.189	0.174	0.165	0.143	0.125	0.108	0.101
SOM	0.3	0.295	0.285	0.201	0.137	0.085	0.09	0.077	0.076	0.062
SOTA	0.3	0.285	0.285	0.265	0.217	0.205	0.195	0.190	0.171	0.145
α-correlation	0.3	0.305	0.334	0.225	0.217	0.180	0.209	0.202	0173	0.172
ExBIRCH	0.3	0.279	0.298	0.299	0.270	0.247	0.294	0.388	0.323	0.289

Biological Discussion

By applying the online CGI web-based application provided by Gibbons[24] on the clusters obtained by using T= 0.90 for ExBIRCH and $\alpha = 1$, $k = 5$ for the adaptive possibilistic clustering on YCC dataset. The proposed algorithm achieves z-score equals 6.6 which is much better performance compared to k-means, Average Link, CLICK, SOM and SOTA which achieve 0.9, 1.8, 2.2, 0.6 and 1.15 respectively. This indicates that the proposed algorithm was successfully able to capture most of the similarity in biological functionalities of different genes in YCC dataset.

Conclusion and Future Works

Clustering gene expression data has several challenges. Gene expression data usually large, high dimensional, noisy and has missing values. Also, existing clusters are highly overlapped and not necessary have convex shape. This paper presented a scalable BIRCH-like algorithm based on average correlation for clustering gene expression data along with its experimental results and analysis. The proposed algorithm inherits the efficiency from its parent algorithm BIRCH. Also the use of average correlation instead of centroids allows dealing with non-convex shaped clusters and avoiding drawback of using Euclidian distance. The adaptive possibilistic paradigm is applied to leaf nodes that allow dealing with overlap and give a rejection capability to noise.

As a future research, the parallel version of BIRCH presented in[25] can be adapted to the proposed algorithm to allow dealing with very large datasets. Also, the proposed algorithm can be used in developing a biclustering algorithm[26]. Finally, the proposed algorithm can be adapted to other application areas such as the sub-sequences clustering.

REFERENCES

1. Kaufman, L. and Rousseeuw, P.J., *Finding groups in data: an introduction to cluster analysis*: John Wiley & Sons (2009).
2. Sokal, R.R., A statistical method for evaluating systematic relationship, *University of Kansas science bulletin,* **28:** 1409-1438 (1958).
3. Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D., Cluster analysis and display of genome-wide expression patterns, *Proceedings of the National Academy of Sciences,* **95(25):** 14863-14868 (1998).
4. Khan, S.S. and Ahmad, A., Cluster center initialization algorithm for K-means clustering, *Pattern recognition letters,* **25(11):** 1293-1302 (2004).
5. Handhayani, T. and Hiryanto, L., Intelligent kernel k-means for clustering gene expression, *Procedia Computer Science*, **59:** 171-177 (2015).
6. Ma'sum, M.A., Wasito, I. and Nurhadiyatna, A., Intelligent K-Means clustering for expressed genes identification linked to malignancy of human colorectal carcinoma. pp. 437-443.
7. Dhillon, I.S., Guan, Y. and Kulis, B., Kernel k-means: spectral clustering and normalized cuts. pp. 551-556.
8. Guha, S., Rastogi, R. and Shim, K.,

CURE: an efficient clustering algorithm for large databases. pp. 73-84.

9. Edla, D.R., and Jana, P.K., A prototype-based modified DBSCAN for gene clustering, *Procedia Technology,* **6:** 485-492 (2012).
10. Sathishkumar, K., Balamurugan, E. and Narendran, P., An efficient artificial bee colony and fuzzy c means based co-regulated biclustering from gene expression data, *Mining intelligence and knowledge exploration*, pp. 120-129: Springer (2013).
11. Wang, L. and Wang, X., Hierarchical Dirichlet process model for gene expression clustering, *EURASIP Journal on Bioinformatics and Systems Biology,* **1:** 5, (2013).
12. Bandyopadhyay, S., Mukhopadhyay, A. and Maulik, U., An improved algorithm for clustering gene expression data, *Bioinformatics*, **23(21):** 2859-2865 (2007).
13. Fu, L. and Medico, E., FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data, *BMC bioinformatics,* **8(1):** 3 (2007).
14. Galluccio, L., Michel, O., Comon, P., Kliger, M. and Hero, A.O., Clustering with a new distance measure based on a dual-rooted tree, *Information Sciences,*. **251:** 96-113 (2013).
15. Zhang, T., Ramakrishnan, R. and Livny, M., BIRCH: an efficient data clustering method for very large databases. pp. 103-114.
16. Pal, N.R., Pal, K., Keller, J.M. and Bezdek, J.C., A possibilistic fuzzy c-means clustering algorithm, *IEEE transactions on fuzzy systems,* **13(4):** 517-530 (2005).
17. Xenaki, S.D., Koutroumbas, K.D. and Rontogiannis, A.A., A novel adaptive possibilistic clustering algorithm, *IEEE Transactions on Fuzzy Systems,* **24(4):** 791-810 (2016).
18. Sharara, S.H. and Ismail, M.A., αCORR: A novel algorithm for clustering gene expression data. 974-981 .
19. Herrero, J., Valencia, A. and Dopazo, J., A hierarchical unsupervised growing neural network for clustering gene expression patterns, *Bioinformatics,* **17(2):** pp. 126-136 (2001).
20. Sharan, R. and Shamir, R., CLICK: a clustering algorithm with applications to gene expression analysis. 16 .
21. Dunn, J.C., Well-separated clusters and optimal fuzzy partitions, *Journal of cybernetics,*. **4(1):** 95-104 (1974).
22. Rousseeuw, P.J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics,* **20:** 53-65 (1987).
23. Hubert, L. and Arabie, P., Comparing partitions, *Journal of classification,* **2(1):** 193-218 (1985).
24. Gibbons, F.D. and Roth, F.P., Judging the quality of gene expression-based clustering methods using gene annotation, *Genome research,* **12(10):** 1574-1581 (2002).
25. Garg, A., Mangla, A., Gupta, N. and Bhatnagar, V., PBIRCH: A scalable parallel clustering algorithm for incremental data. pp. 315-316.
26. Madeira, S.C. and Oliveira, A.L., Biclustering algorithms for biological data analysis: a survey, *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* **1(1):** pp. 24-45 (2004).