Biclustering and Triclustering of Gene Expression Microarray data : AN OVERVIEW

Ananya Anindita B120007
Mitul Kumar Dayal B120033
Belagallu Shashi Sekhar B120014

Abstract

This paper reviews prior applications of Biclustering and Triclustering of Gene Expression Microarray data between 2013 and 2023.In traditional gene expression analysis, clustering methods help identify groups of genes that exhibit similar expression patterns. This is valuable for understanding gene functions and relationships with transcription factors.This paper provides an overview of biclustering and triclustering techniques applied to gene expression microarray data, a specialised approach, where a sub-matrix is created to identify genes exhibiting highly similar behaviour across specific conditions. Essentially, it aims to provide a more comprehensive and accurate analysis of microarray data, taking into account nuanced relationships among genes under different conditions. Biclustering is a method that identifies subsets of genes showing similar expression patterns only under specific conditions, while triclustering extends this concept to include a third dimension, such as time or experimental parameters. The paper reviews the current landscape of these techniques, emphasising their importance in uncovering nuanced correlations within complex gene expression datasets. It explores various biclustering and triclustering approaches, offering insights into their methodologies. The goal is to enhance understanding and awareness of these methods, fostering further development and application in the analysis of gene expression microarray data.

I. INTRODUCTION

After mapping the entire genome, DNA microarray analysis has become a highly popular method in bioinformatics for understanding how genes function. Biologists face a challenge with the massive amount of data generated by DNA microarray experiments. Clustering is a widely used method to organise and make sense of the large amount of data produced by experiments.It is presently the far most used method for gene expression analysis which provides a divide-and–conquer strategy to extract meaningful information from expression profiles. Clustering algorithms do not always give good results because most gene models are included in the

experimental set. Therefore, integration should be adapted to the method that can find local patterns from gene expression data.Binary clustering can find local patterns by searching for gene patterns by a set of experiments.Biclustering algorithms identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions, each gene and condition in a bicluster are only a subset of the gene and condition. In biclustering, if some points are similar in several dimensions they will be clustered together in that subspace proved of great value in finding the interesting patterns in the microarray expression data.

However, both clustering and biclustering are inadequate when analysing gene expression microarray data and focusing on the effects of time on gene behaviour. Such longitudinal experiments allow for a comprehensive analysis of molecular processes in which time plays an important role. Therefore, genes need to be analysed using specific tools and according to specific conditions and time points. Therefore, triclustering becomes a balancing tool.

In simpler terms, when we represent gene expression patterns in clusters, it's useful for many analyses. However, it falls short when trying to capture detailed co-expression patterns among genes within a gene module.Biclustering breaks down the co-expression matrix into clusters related to specific conditions, but it struggles when time is an additional factor. That's where triclustering comes in handy. Triclustering helps uncover genes co-expressed across both conditions and time, providing a more detailed understanding of pairwise gene relationships in a broader context.

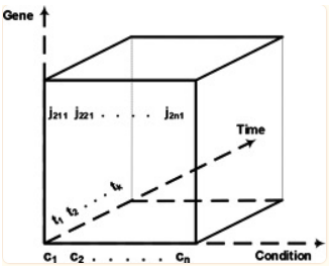
II. RELATED WORKS

J. Hartigan introduced binary clustering in his paper “Direct Clustering” as a divide-and-conquer strategy that initially divides the input matrix into submatrices until a certain number of user-defined binary clusters are obtained. This algorithm works well for evaluating numerical data. Although it is not used for gene expression data, it is still considered the first linkage algorithm. This algorithm can provide fast results. However, the disadvantage of this algorithm is that it cannot reconstruct the matrix after it has been transformed into a submatrix.Cheng and Church (CC) used two clustering methods in gene expression patterns, from the concept of concrete sequence coverage. The two-set quality test here is performed using residual square measure (MSR). In this method, the similarity between the gene and the binary cluster is evaluated based on their expression values.Improving the accuracy of CC, Mukhopadhay et al. added a new SMSR (Scaled MSR) scale based on the same concept that Church used in his algorithm. SMSR now recognizes scaling patterns. Similarly Research in Triclustering focuses on Integrating biological knowledge, handling large datasets, incorporating temporal dynamics and integrating with other omics data. Cheng and Church (1999) First proposed Biclustering algorithm based on singular value decomposition (SVD) for 2D gene expression data, later extended to 3D with Triclustering for time series gene expression data(TCS).Lazzeroni and Owen (2002) Proposed the COEX algorithm based on mutual information, one of the first

dedicated triclustering methods for microarray data. It identified co-expressed genes under specific conditions and time points.
Later Troyanskaya et al. (2003): Developed the ISA algorithm, emphasising biclustering with informative submatrices. This paved the way for incorporating biological relevance into triclustering.Tan et al. (2005) used the technique iCluster (iterative co-clustering) based on k-means clustering and biclustering and Improved scalability and accuracy for large datasets compared to previous methods. Also Inspired further development of hybrid clustering approaches for triclustering gene expression data. Mukhopadhyay et al. (2006): Introduced the OPF triclustering method, focusing on identifying overlapping patterns in gene expression data. This addressed limitations of earlier methods that assumed disjoint clusters.
Evolutionary Approaches like Zheng et al. (2007): Proposed an evolutionary biclustering method with a fitness function based on gene correlation and condition specificity. This marked the application of evolutionary algorithms to triclustering. Zhang et al. (2008): Developed the TSPEA gene expression triclustering algorithm, utilising a travelling salesman problem-based approach. This improved efficiency and scalability compared to earlier methods. Wu et al. (2010): Presented the EGA-based triclustering approach, employing an enhanced genetic algorithm. This incorporated domain knowledge and achieved superior performance in identifying biologically relevant submatrices. Lu et al. (2012): Proposed the MPSO algorithm for triclustering, combining particle swarm optimization with mutual information for detecting differentially expressed genes under specific conditions. This improved accuracy and efficiency in identifying relevant patterns.
Recent Advancements like Yang et al. (2019) Introduced a method TriRNSC (Triclustering of RNA-Seq) with restricted neighbourhood search for triclustering. Adapted triclustering for analysing RNA-seq data, which provides higher resolution gene expression profiles
Continuation: Demonstrated the versatility of triclustering for different types of gene expression data and its continued relevance in biological research FCTriClus (2017): Developed a fuzzy co-triclustering method based on density clustering and information gain. This handled uncertainty in gene expression data and identified more informative submatrices.

III. DATA PRE-PROCESSING(Creating a GCN)

The concept of a triclustering is illustrated as in the below fig, where the x-axis represents conditions, the y-axis represents the time of observation, and the z-axis represents gene expression values in a 3D space.



In a 3D microarray dataset with n1 genes, n conditions, and k time points, a tricluster (TC) is defined as a subspace of the original space, where subsets of genes (n1), conditions (n), and time points (k) are considered. Biclustering, a related concept, is acknowledged as an

NP-hard problem, indicating that finding optimal solutions is computationally challenging and often requires heuristic approaches. This difficulty extends to triclustering, where heuristic-based algorithms are also preferred due to the high computational cost compared to biclustering algorithms.

In the data pre-processing phase, the focus is on transforming the initial 3D data matrix M into a Gene Co-expression Network (GCN). This involves establishing a graphical network that connects a set of genes based on their pairwise similarity scores. The construction of GCNs typically relies on gene expression data, calculated using the Pearson correlation coefficient (ρ) as a measure of similarity. The threshold value (th) is applied to filter out statistically significant connections in the network. In the input data, where multiple time points are defined for each condition, gene expression levels are represented in a condition-time plane. For instance, the expression levels of gene x are structured as

x = ((x11, x12, ..., x1t), (x21, x22, ..., x2t), ..., (xn1, xn2, ..., xnt)),

where n is the number of conditions, and t is the number of time points for each condition. The Pearson correlation coefficient is then computed between two genes, x and y, whose expression levels are represented in matrix form within a condition-time plane. The equation captures the correlation between gene expression profiles across multiple conditions and time points.

$$x = ((x_{11}, x_{12}, \ldots, x_{1t}), (x_{21}, x_{22}, \ldots, x_{2t}), \ldots, (x_{n1}, x_{n2}, \ldots, x_{nt})) \text{ and}$$

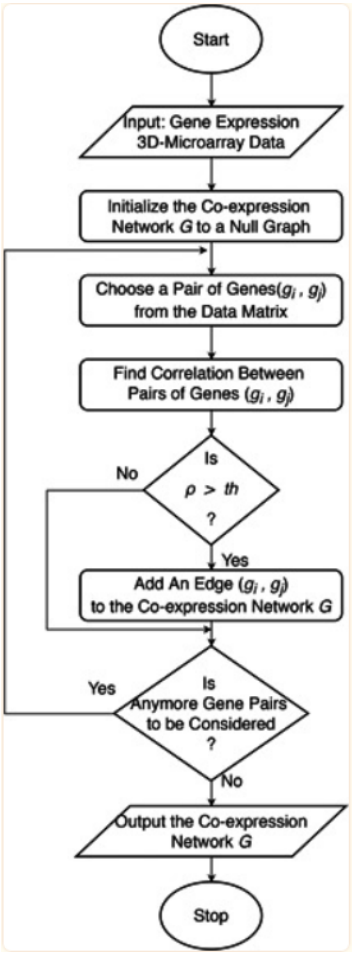
$$y = ((y_{11}, y_{12}, \ldots, y_{1t}), (y_{21}, y_{22}, \ldots, y_{2t}), \ldots, (y_{n1}, y_{n2}, \ldots, y_{nt}))$$

computed over *n* conditions in *t* time points can be defined as

$$\rho(x, y) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{t} (x_{ij} - \bar{x})(y_{ij} - \bar{y})}{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{t} (x_{ij} - \bar{x})^2 \sum_{i=1}^{n} \sum_{j=1}^{t} (y_{ij} - \bar{y})^2}}$$

Here, xij and yij denote the expression levels of genes x and y for the ith condition in the jth time point, while $\bar{x}$ and $\bar{y}$ represent the average expression levels of genes x and y, respectively. In summary, this phase involves the transformation of the original data matrix into a GCN, emphasising the computation of Pearson correlation coefficients to establish meaningful connections between genes based on their expression patterns across conditions and time points.

And the workflow is as follows



and the pseudocode of this phase of work is described in Algorithm below

Input: Expression Matrix(M) and Threshold(th)
Output: Co-expression network (G)

```
Initialize the co-expression network G(E, V) = ϕ
Set R = NumRows(M)
for i = 1 to R-1 do
    for j = i+1 to R do
        ρ = PearsonCorrelation(g_i, g_j)
        if (ρ ≥ th) then
            Add (g_i, g_j) to graph G
        end
    end
end
return G
```

IV. APPLYING RNSC Algorithm

RNSC is a local search clustering algorithm. A single experiment of the method as applied to a graph G consists of several basic stages: 1. Either read or randomly generate an initial clustering Co € C[G]. 2. Apply the naive cost function to the clustering and data structures. Attempt to minimise the naive cost by modifying the clustering one move at a time, reaching a best naive clustering C,. 3. Do the same for the scaled cost: Starting with the naive clustering C,, apply the scaled cost function to the clustering and data structures and attempt to minimise the scaled cost by making one move at a time. The best scaled clustering, the output of the experiment, is denoted C,. This is a very high-level description of a single experiment. Generally, more than one experiment will be run,

generating a set {ce }XZ | where My is the number of experiments. The output clustering will then be Cp, the element of this set with the lowest scaled cost.

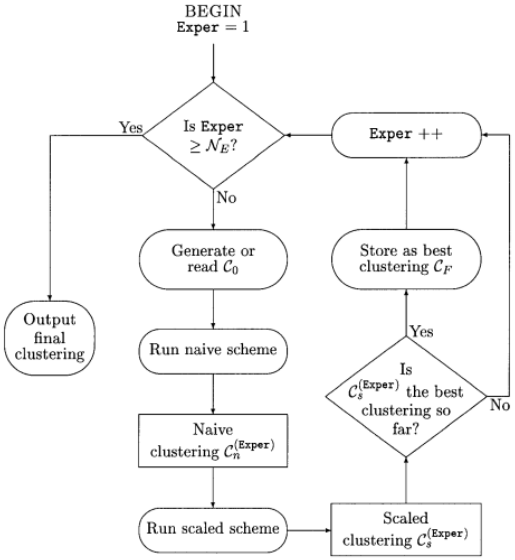


Figure: The RNSC algorithm

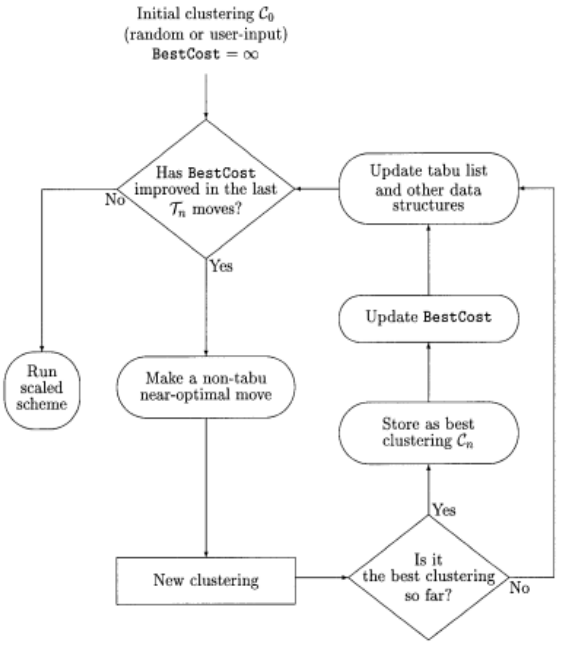


Figure : The RNSC naive cost scheme

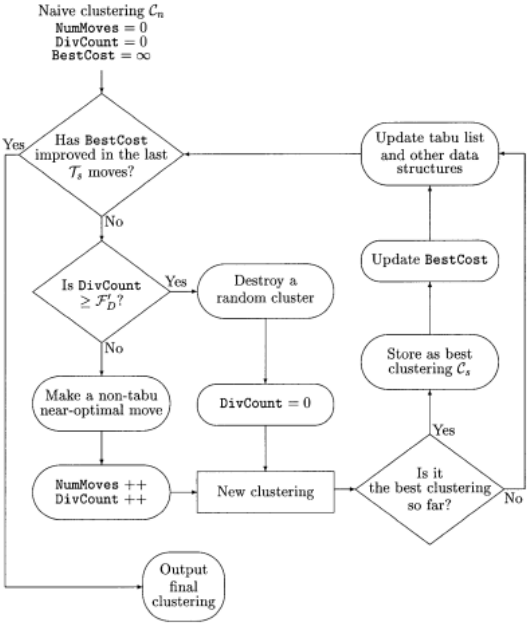


Figure : The RNSC scaled cost scheme

The naive cost function would be:

$$C_{\mathrm{n}}(G,P) = \frac{1}{2}\sum_{v}(c_{\mathrm{p}}(V) + l_{\mathrm{p}}(V))$$

The scaled cost function would be:

$$C_{\mathrm{p}}(G,P)=\frac{(n-1)}{3}\sum_{v\epsilon V}\frac{(c_{\mathrm{p}}(V)+l_{\mathrm{p}}(V))}{\mid R(v)\cup p_{v}\mid}$$

Types of Moves in RNSC algorithm:
Move Types Every move type that RNSC uses involves moving a single vertex from one cluster to another, possibly emptying a cluster or creating a singleton cluster in the process. As detailed in Section 3.3, RNSC uses a constant number of clusters, and the clusters may be empty. Therefore in a graph G = (V, FE), the neighbourhood of a clustering C, denoted N(C), consists of |V| - (Mc — 1) moves, where NG is the number of clusters. This is because there are |V| vertices, each of which can be moved to any cluster that it does not occupy, i.e. one of Ng — 1 clusters.

1. Global Moves In RNSC,
2. Random Moves (Diversification)
3. Restricted Neighbourhood Moves: (Intensification)
4. Forbidden (Tabu) Moves.

V. VALIDATING RESULTS

The research findings are validated through (GO) term analysis and (KEGG) pathway analysis.

a. GO term analysis:

The aim of the GO term analysis is to validate the tricluster solutions by standardising gene characteristics and product attributes across different species and databases.

The GO project constructs a framework for defining gene product characteristics, facilitating the conversion of machine-readable outputs into biologically significant information. Through the application of the GO Term Finder tool, gene grouping hypotheses undergo independent validation, classifying co-expressed genes according to cellular component, molecular function, or biological process. The analysis underscores elemental gene product activities, cellular components, extracellular environments, and essential molecular events governing the functionality of biological units. This methodology ensures a holistic comprehension of gene expression patterns grounded in biological significance.

b. KEGG pathway analysis:

The KEGG pathway analysis aims to interpret biological systems by evaluating multi-level functions and utilities of organisms and cells.

KEGG, serving as an extensive resource database, orchestrates the mapping of molecular entities such as genes, proteins, and small molecules onto an array of structures like molecular interaction networks, pathways, hierarchies, and modules. This mapping procedure involves a set operation, creating organism-specific pathways by autonomously aligning manually annotated genome data with established pathway maps. The versatile KEGG Mapper tool excels in multiple functions, encompassing MODULE mapping, KEGG

pathway mapping, and BRITE mapping, thereby enriching the exploration of molecular interactions and biological pathways.

VI. CONCLUSIONS

Clustering methods are rather effortless to implement and have a reasonable computational complexity yet fail to represent the genuine clustering of data. The performance of every clustering algorithm may vary significantly with diverse data sets, and there is no absolute finest algorithm among the clustering algorithms. The significant disadvantage of clustering algorithms are the fact that time variation is not considered in its calculations, variations of densities in the data space resulting in overlapping clusters, cluster validation, presence of irrelevant attributes, high level of background noise, no prior knowledge and the dimensionality curse.So far Biclustering and Triclustering algorithms have proven significant improvement to the weaknesses and inadequacy of clustering algorithms.A novel framework based on RNSC is introduced for tricluster identification in gene expression profiles. The process begins with constructing a Gene Co-expression Network (GCN) in the data preprocessing phase. The proposed TriRNSC framework employs both naive and scaled cost functions for gene expression analysis, demonstrating comparable performance to state-of-the-art methodologies. Results, particularly considering gene size, are impressive. Biological validations through GO and pathway analysis confirm TriRNSC's coherence and significance. This suggests that the RNSC algorithm is emerging as a promising graph-based approach for extracting triclusters from 3D gene expression microarray data. However, challenges persist, and **opportunities for future improvement include addressing volume, execution time, and complexity issues.**

How we intend to improve results:

1. ***Volume of Data -: the more the data the better***
2. ***Gene Ontology(GO) and KEGG pathway analysis : To validate the results biologically***

REFERENCES

[1] Shreya Mishra,Swati Vipsita “Triclustering of Gene Expression Microarray data using Evolutionary Approach”

[2] Bhawani Sankar Biswal, Sabyasachi Patra, Anjali Mohapatra,and Swati Vipsita , “TriRNSC: triclustering of gene expression microarray data using restricted neighbourhood search”

[3] P. O. Brown and D. Botstein, “Exploring the new world of the genome with dna microarrays,” Nature genetics, vol. 21, pp. 33–37, 1999.

[4] C. Rubio-Escudero, F. Mart´ınez-Alvarez, R. Romero-Zaliz, and I. Zwir, ´ “Classification of gene expression profiles: comparison of k-means and expectation maximisation algorithms,” in Hybrid Intelligent Systems, 2008. HIS’08. Eighth International Conference on. IEEE, 2008, pp. 831–836.