Identifying Potential Biomarkers for Diseases Diagnosis Through Co-Expression Analysis: An Optimization Approach

Billel Kenidra[1], Mohamed Benmohammed[2]
[1] Ecole nationale Supérieure d'Informatique (ESI), BP, 68M Oued-Smar,16270 Alger, Algeria
[2] Laboratoire d'Informatique REpartie (LIRE), University of Constantine 2, Constantine, Algeria
E-mail: b_kenidra@esi.dz[1], mohamed.benmohammed@univ-constantine2.dz[2]

Keywords: cancer-diagnosis biomarkers, gene-function inference, co-expressed analysis, high-dimensional big datasets, optimization

Received: Mai 16, 2024

While gene function is dysregulated in cancer, detecting these abnormalities will assist in diagnosis. DNA microarray technology is a significant tool for conducting research in functional genomics. This technology has been developed to assess gene expression levels across different samples. It has been used extensively in cancer research, where mutations may switch off or increase gene expression level in malignant cells. Identifying clusters of co-expressed genes has emerged as a pivotal stage in comprehending functional genomics, as it aligns with the notion that genes with related functions often exhibit similar expression patterns across varied samples. The biologist starts by analyzing the known functions of genes within each cluster in order to infer the function of the entire cluster, this inferred function is then ascribed to all unknown genes within the respective cluster. High-dimensional clustering has proven to be a fruitful pursuit for identifying co-expressed genes. This optimization problem, which is non-convex in nature, has been demonstrated to be NP-hard. DNA microarray provides large amount of gene expression datasets, resulting in millions of measurements. Practically, when there is a greater quantity of datasets to cluster and a larger number of clusters to consider, the potential number of partitions increases significantly. Consequently, this presents a computationally intensive and time-consuming combinatorial challenge, exacerbated by the high-dimensional nature of the gene expression datasets. Despite the availability of numerous high-dimensional clustering algorithms, there remains room for improving quality and reducing running-time. Indeed, the selection of a clustering algorithm is contingent upon the specific attributes of the dataset. To that end, we have proposed an algorithm specifically tailored to deal with big and high-dimensional datasets that optimizes the computational complexity. By applying this algorithm several times, a set of clusters including genes that are grouped together across multiple runs, will emerge. The centroid of each emerged cluster will be used to identify the optimal partition. Empirical studies unequivocally demonstrate an average 48% improvement in quality and an average 60% reduction in running-time compared to the approaches outlined in the related-work section.

Povzetek: Razvit je optimiziran algoritem za analizo soizraznosti genov, kar omogoča hitrejšo in natančnejšo identifikacijo biomarkerjev za diagnozo bolezni.

1 Introduction

Gene expression is the process by which information from a gene is used to synthesize a functional gene product, typically a protein [1]. Genes, which are segments of DNA, contain instructions for building molecules called proteins. These proteins perform a wide variety of functions in the body, such as serving as enzymes, structural components of cells, or signaling molecules. The process of gene expression involves two main stages [2]:

1- Transcription: This is the first step in gene expression, where the information encoded in a gene's DNA is transcribed into a molecule called messenger RNA (mRNA). The enzyme RNA polymerase reads the DNA sequence of a gene and synthesizes a complementary RNA molecule. The resulting mRNA carries the genetic code from the nucleus (where DNA is located) to the cytoplasm (where protein synthesis occurs).

2- Translation: In this step, the information in the mRNA is used to build a specific protein. Ribosomes, the cellular machinery responsible for protein synthesis, read the sequence of codons in the mRNA (each codon represents a specific amino acid) and link together the corresponding amino acids to form a polypeptide chain, which then folds into a functional protein.

The regulation of gene expression is crucial for the proper functioning of cells and organisms. It allows cells to respond to environmental signals, developmental cues, and physiological needs [3]. On the other hand, aberrations in gene expression can lead to various diseases, including cancer and genetic disorders. Understanding gene expression is fundamental to fields such as genetics, molecular biology, and medicine, as it provides insights into the mechanisms underlying cellular functions and the development of diseases.

Epigenetic modifications are changes to the DNA molecule or to the proteins with which it interacts that do not involve alterations to the underlying DNA sequence. These modifications can influence gene expression, playing a crucial role in various cellular processes and developmental events. They are also implicated in various diseases, including cancer [4]. There are several types of epigenetic modifications, and the most well-studied one is DNA methylation, it involves the addition of a methyl group (CH3) to a cytosine base, typically occurring in the context of a cytosine-guanine (CpG) dinucleotide. Methylation of DNA is often associated with gene silencing. When a gene's promoter region is heavily methylated, it can prevent the binding of transcription factors and RNA polymerase, leading to reduced gene expression [5].

Gene expression dysregulation refers to abnormal or irregular patterns of gene activity, where genes are either over-expressed or under-expressed. Dysregulation of gene expression is associated with various diseases, including cancer. In cancer, the normal control mechanisms that regulate gene expression are disrupted. This leads to abnormal patterns of gene expression, contributing to the development and progression of tumors. Tumors often exhibit irregular patterns of gene expression compared to normal tissues. This dysregulation can involve both upregulation (increased expression) and downregulation (decreased expression) of genes, contributing to the hallmarks of cancer.

DNA microarrays are powerful tools used in molecular biology and genomics to measure the expression levels of thousands of genes simultaneously. They provide a comprehensive snapshot of gene activity within a cell or tissue. DNA microarrays have a wide range of applications in molecular biology and genomics, providing a powerful tool for studying gene expression and genomic variations on a large scale. Here are some key applications of DNA microarrays:

- Gene Expression Profiling: Identifying genes that are upregulated or downregulated in response to different conditions.
- Cancer Research: Identifying genes associated with cancer development, progression, and response to treatment.
- Pharmacogenomics: Studying how genes influence responses to drugs.
- Functional Genomics: Investigating gene function on a large scale, in order to identify potential biomarkers for diseases diagnosis. (This paper focuses on this particular subject).

The workflow of DNA microarrays involves several key steps, and here is a broad overview of the process:

1) Sample Preparation: RNA or DNA is extracted from cells and converted into labeled cDNA (complementary DNA) using fluorescent dyes.
2) Hybridization: The labeled cDNA is applied to the microarray, allowing it to hybridize with the immobilized DNA probes.
3) Detection: The microarray is scanned to detect the fluorescence signals, indicating the abundance of specific genes in the sample.
4) Data Analysis: The data generated from microarrays are analyzed using bioinformatics tools to identify differentially expressed genes.

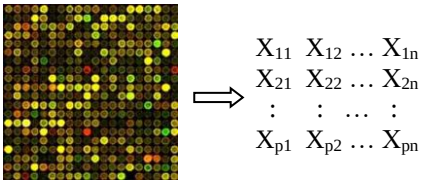

Figure 1: Matrix of expression levels resulting millions of measurements

Figure 1 shows the matrix of expression levels, if the gene shows high expression in the experimental sample but low expression in the control/reference sample, the corresponding spot on the array will mainly appear in red. Conversely, if the gene is expressed at a low level in the experimental sample but a high level in the control/reference sample, the array spot will predominantly be green. A yellow spot on the array indicates similar levels of gene expression between the experimental sample and the control/reference sample. The laser assesses the hybridization level of labeled cDNA with each probe by scanning the array for fluorescence at the Cy5 and Cy3 wavelengths. This scanning process is employed to quantify the intensity of light emitted at each spot on the screen. Gene expression levels are estimated based on the amount of hybridization intensity. Subsequently, each gene (represented as a spot) is isolated, and its relative fluorescence intensity is employed to generate a numerical value, describing the level of expression [6].

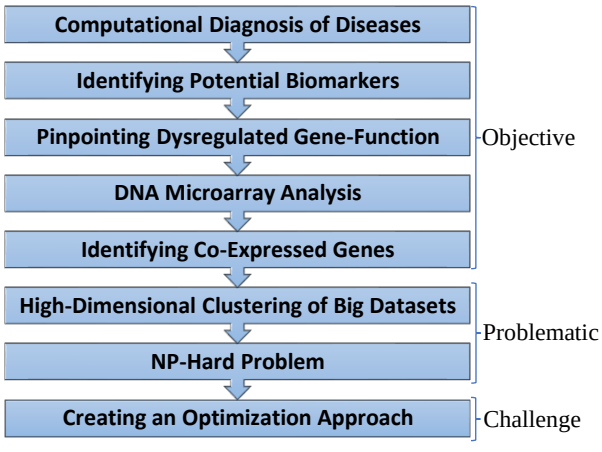

Figure 2. Steps involved in identifying potential biomarkers for diseases diagnosis

Co-expressed genes are genes that exhibit similar expression patterns across a set of samples. However, genes that consistently show coordinated expression changes in disease conditions can provide clues about the functions and pathways these genes are involved in. The similarity in expression suggests that these genes may be

functionally related or involved in the same biological processes [7]. This, in turn, can help identify useful biomarkers for diagnosis or prognosis (see Figure 2).

The assessment of p genes' expression levels under n experimental conditions is frequently depicted as a real-valued matrix of dimensions p x n (Figure 1), where the entries denote the corresponding expression levels. Analysis of the gene expression matrix can be conducted by identifying shared patterns among samples or by identifying shared patterns among genes. In the first scenario, genes are treated as data objects, while samples are considered attributes or dimensions. Employing this approach can enhance our comprehension of gene functionality and aid in identifying the genes associated with specific diseases. Contrarily in the second scenario, where samples are treated as data objects for clustering, with genes acting as attributes. The application of this clustering method improves the assessment of medication effects on genes, potentially paving the way for more personalized treatment strategies. Certain tumor types may exhibit varying responses to specific medications, highlighting the importance of tailoring treatments based on individual characteristics.

Biologist needs computational tools to identify clusters of co-expressed genes using the matrix of expression levels as the input dataset. Indeed, this task becomes challenging when working with high-dimensional datasets, where tens of thousands of genes are measured simultaneously over hundreds of samples, resulting millions of measurements, it presents a challenge for conventional clustering algorithms because it can lead to overfitting, which results in inaccurate and time-consuming clustering. However, clustering of high-dimensional gene expression data is a challenging task that requires optimization of several parameters.

Because the biologist needs to run the algorithm multiple times as the sample changes at each time point, managing running-time becomes a significant challenge. In this paper, we propose an optimization approach to identify co-expression that could serve as potential biomarkers for disease diagnosis, it means finding the optimal partition of clusters with the highest quality while minimizing the running-time.

The rest of this paper is as follows: “related works” section reviews relevant research in the field and demonstrates the gap in knowledge that the current study aims to fill. “Methodology” section delves into the proposed approach. “Experiments and Results” section explores datasets and performance metrics used in experiments, along with the results obtained. “Discussion” section is dedicated to comparison with the state-of-the-art methods, and discusses the reasons behind the performance of our approach. The “Conclusion” section summarizes the main findings and emphasizes the significance of the novel contribution to the field of gene expression analysis and biomarker identification.

2 Related works

Clustering microarray data is challenging due to the curse of dimensionality. This issue arises when there are a large number of genes compared to a relatively small number of samples, making the clustering process more complex.

Table 1 highlights the strengths and limitations of various methodologies used to manage large and high-dimensional genomic datasets, based on this experimental study involving 54,675 genes and 200 samples, resulting in over 10 million measurements.

Table 1: Strengths and limitations of some commonly used methodologies for handling large and high-dimensional genomic datasets.

Methodologies	**Strengths**	**Shortcomings**
Density Peak Clustering [8] (2014)	Density Peak Clustering (DPC) has some advantages, such as being able to handle clusters with different shapes and sizes, as well as being less sensitive to input parameters compared to some other clustering methods.	Usually, density peak clustering fails to recognize clusters with varying densities.
Local Gap Density [9] (2019)	This paper introduces a novel density type, termed "local gap density," within the context of the k-NN graph, specifically tailored for high-dimensional datasets. The local gap density for each data point not only accounts for the count of neighboring points but also incorporates the average distance from the focal point to all others within its nearest neighbor. Consequently, this density definition highlights core points in sparsely populated regions, facilitating their straightforward identification. By leveraging these core points, we effectively discern potential cross-cluster edges in the k-NN graph.	One limitation is that it often struggles to identify clusters with lower density.
Variational Density Peak	VDPC is specifically designed to handle clustering tasks on datasets exhibiting diverse density distributions. this approach begins by introducing a	It operates on the premise that cluster centers exhibit notably higher local density in contrast to their neighboring

Clustering [10] (2023)	unique method to identify representative data points, constructing initial clusters based on these representatives, and subsequently analyzing the properties of the identified clusters. Additionally, all data points are divided into distinct levels based on their local density, proposing a unified clustering framework that combines the strengths of both DPC and DBSCAN. As a result, the initially identified clusters spanning various density levels are systematically processed to form the final clusters.	points. Although VDPC can handle clusters of varying shapes and sizes, it poses a computationally demanding and time-consuming combinatorial challenge.
I-k-means-+ [11] (2018)	This paper presents an iterative method to improve the quality of solutions produced by the k-means algorithm. Named iterative k-means minus–plus (I-k-means−+), this technique refines the k-means solution by iteratively removing one cluster (minus), splitting another (plus), and then reapplying the clustering process. The acceleration of I-k-means−+ is achieved through various methods that identify clusters to remove, determine which clusters to split, and speed up the re-clustering process.	The accuracy of I-k-means-+ may be unsatisfactory, particularly when the number of required clusters, k, is large. Additionally, as the number of clusters increases, the algorithm becomes more computationally intensive and time-consuming.
NQ-DBSCAN [12] (2018)	The authors present a novel local neighborhood search technique called NQ-DBSCAN, designed to improve the performance of DBSCAN by significantly reducing the number of unnecessary distance calculations. Theoretical analysis and experimental results show that, with the use of an indexing technique, NQ-DBSCAN achieves an average time complexity of O(n*log(n)), and in the best-case scenario, it operates in O(n) with optimal parameter settings. This efficiency makes NQ-DBSCAN highly suitable for real-time data applications.	Even though NQ-DBSCAN uses neighbor searching with indexing to minimize unnecessary density computations, it remains an enhanced version of DBSCAN. Being a density-based algorithm, it becomes almost ineffective for big and high-dimensional datasets due to the curse of dimensionality, which significantly increases its computational time.
M-CLUBS [13] (2014)	M-CLUBS exceeds the accuracy of earlier hierarchical methods and offers faster computations compared to partition-based approaches. Unlike other algorithms, including k-means and its recent variants, M-CLUBS delivers superior speed and precision. The algorithm features both a divisive and an agglomerative phase, employing a least quadratic distance criterion with unique analytical properties for swift computation during sample repartitioning. M-CLUBS autonomously generates high-quality clusters without user input and shows robustness to noise. It outperforms similar methods like BIRCH in speed and accuracy, and is particularly effective for analyzing microarray data represented as numeric arrays, especially with Euclidean distances.	Indeed, M-CLUBS surpasses other algorithms like BIRCH, k-means, and its improvements such as k-means++ in both speed and accuracy, as shown in the experimental section of the M-CLUBS paper [13]. However, the challenge remains when handling large and high-dimensional datasets (over 10 million measurements in this case), because the algorithm's divisive and agglomerative phases reduce the clustering process's accuracy and efficiency, as indicated in the experimental results section of the paper.

Actually, before choosing the algorithm, it's essential to consider the specific challenges posed by the biological datasets, and both the computational complexity and the speed issue posed by the algorithm.

Nonetheless, Pirim et al [14] asserted that there is no clustering algorithm that exhibits optimal performance across all clustering problems. This reality underscores the importance of creating an algorithm tailored to the specific task at hand.

3 Methodology

The proposed clustering technique is developed to analyze large-scale gene expression profiles (millions of measurements), it groups genes with similar expression patterns across different samples, and also groups samples with similar expression patterns across different genes (as shown in Figure 1). In both cases, performing clustering task with this type of dataset remains a significant challenge.

3.1 Problem formulation

Consider a dataset, ALL, consisting of N genes:
ALL = $\{G_1, G_2, \ldots, G_i, \ldots, G_N\}$, where $G_i = \{S_1, S_2, \ldots, S_j, \ldots, S_D\}$, each S_j is an experimental condition or a sample related to the gene G_i.

The genes should be clustered into a finite number of clusters: C = {C_1, C_2, …, C_k}. The resulted clusters should satisfy the following constraints:

1. $C_i \neq \emptyset \quad i \in \{1, \dots, k\}$
2. $\bigcup_{i=1}^{k} C_i = ALL$
3. $C_i \cap C_j = \emptyset \quad i, j \in \{1, \dots, k\}\ and\ i \neq j$

This problem is known to be NP-hard (Nondeterministic Polynomial time hard). As a consequence, exhaustive search is ineffective, since the number of partitions that may be generated grows substantially as the number of genes and the number of samples, both grow. To address this tough optimization issue, efficient and fast computational tools are required.

3.2 Clustering algorithm

The proposed approach is divided into two phases. Six steps are involved in the first one:

Step 1: Select k genes from ALL at random, to serve as the initial cluster centers.
k_Genes = Random (k, ALL) = {X_1, X_2, …, X_k}.
REST = ALL – k_Genes.

Step 2: If (REST is null) then go to step 6. Else:
Calculate the pairwise distances among k_Genes, the minimum distance among all pairs will be used as a threshold.

$$\underset{Xi,Xj \in k\_Genes}{\text{Min (Distance (Xi, Xj))}} = (\text{Threshold, Xn, Xm})$$

Step 3: Select one gene G, at random from REST, and calculate the pairwise distances between G and every gene $X_i \in$ k_Genes. Then, determine the smallest distance among all pairs (G,X_i), according to the following equation:

$$\underset{\substack{G \in \text{REST} \\ Xi \in k\_Genes}}{\text{Min(Distance(G, Xi))}} = \text{Smallest\_Distance}$$

Step 4: If (Smallest_Distance > Threshold) then

1. Remove both X_n and X_m -determined in step 2- from k_Genes.
2. Add G -got in step 3- to k_Genes.
3. Use the k_Genes elements -which should be k-1 elements- as centroids to create k-1 clusters.
4. Within each cluster C_i determine the farthest gene and its distance from Ci center, such as:

$$\underset{Ri \in Ci}{\text{Max (Distance (Center\_Ci, Ri))}} = (\text{Farthest i , its\_dist})$$

Over all clusters, determine the elected gene R_p having the longest distance according to the following equation:

$$\underset{i \in \{1,\dots,k-1\}}{\text{Max (Farthest i , its\_dist)}} = \text{Rp}$$

Step 5: Add R_p to k_Genes, and go to step 2.

Step 6: Use k_Genes as centroids to generate k clusters.

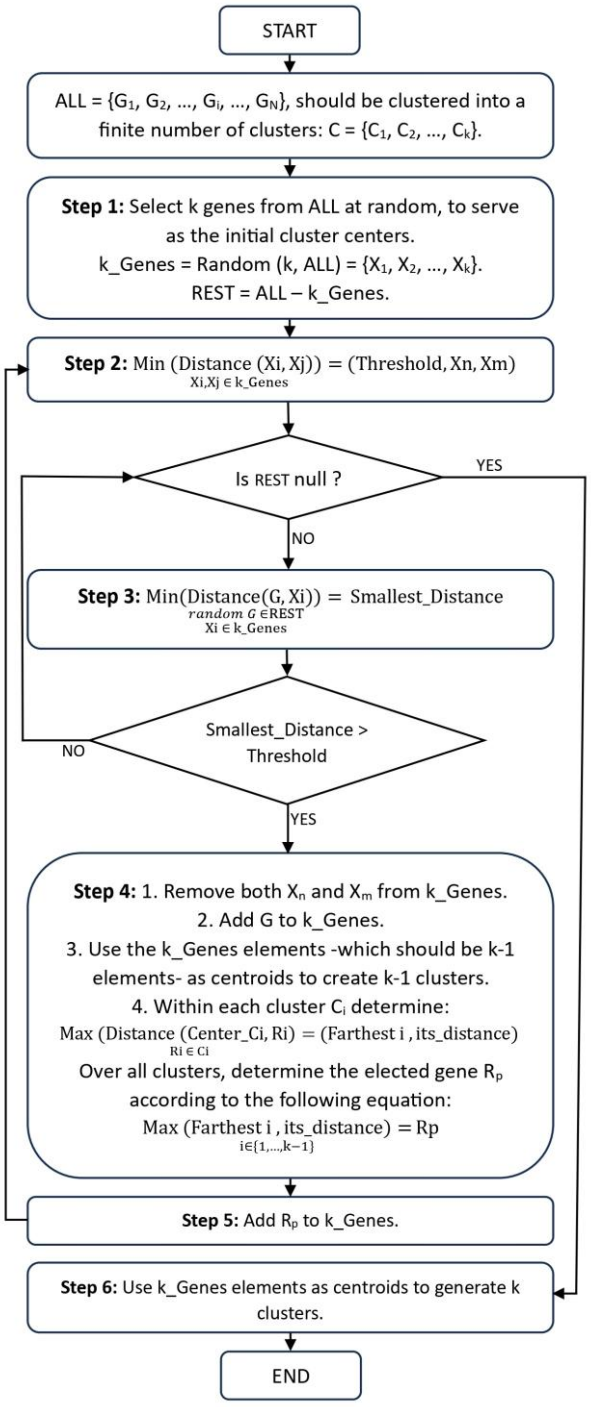

Figure 3: 1st phase flowchart

The second phase involves five steps:

Step 7: The algorithm described in *Phase_1* should be running several times using the same datasets, and the resulting clusters will be stored in a matrix of membership for further processing, as the following pseudo-code illustrates:

```
N ← How_Many_Times;
WHILE (N > 0) {
  Run the algorithm described in Phase_1;
  Store each element membership in the matrix;
  N ← (N - 1);
}
```

Step 8: Identify the clusters $\{Y_1, Y_2, \ldots, Y_k\}$, including genes that are grouped together during the course of all running times;

Step 9: Determine the centers of $\{Y_1, Y_2, \ldots, Y_k\}$ such that each gene is assigned to the nearest center, this will result in the formation of new clusters $\{Z_1, Z_2, \ldots, Z_k\}$;

Step 10: Determine the average distance between the center of each cluster $Z_i \in \{Z_1, Z_2, \ldots, Z_k\}$, and every gene that is related to it;

Step 11: Determine genes $\{G_1, G_2, \ldots, G_i, \ldots, G_p\}$, whose distance from the center is greater than the average distance calculated in step 10;

Step 12: The final clusters will emerge if each gene $\in \{G_1, G_2, \ldots, G_i, \ldots, G_p\}$ is assigned to the nearest center;

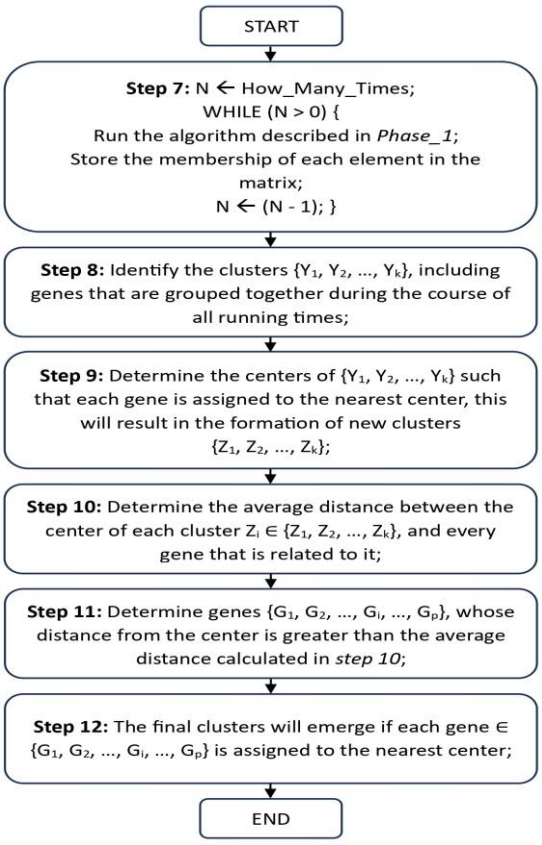

Figure 4: 2nd phase flowchart

3.3 Optimization techniques

The techniques behind this optimization rely on two key strategies: the first involves using a small subset of the overall dataset to identify the optimal starting positions for the initial seeds (as demonstrated in Phase-1). This approach yields a solution very quickly, but with lower accuracy. To address this, we incorporated the second strategy by running Phase-1 multiple times using the same datasets to identify a set of clusters, including genes consistently grouped together across all runs. This second strategy has greatly improved the quality of clustering while keeping the running-time significantly reduced. To tackle the problem of outliers, we further enhanced these techniques by adding Steps 10, 11, and 12.

3.4 Pseudocode

Input:
A matrix (see Figure 1).

Output:
A set of clusters of genes compacted and well separated.

START

Vars:
ALL: a set of genes (vectors);
Membership: a matrix for storing the resulting partitions;
How_Many_Times: integer variable;
Average-Distance: real variable;
Outliers: a set of genes;
Best_Soltuion: a set of clusters;

Method:
ALL: = $\{G_1, G_2, \ldots, G_i, \ldots, G_N\}$;
How_Many_Times: = 3; // Optimal value empirically revealed.
WHILE (*How_Many_Times* > 0) do // Step_7.
{
Membership: = **Phase_1(***ALL***)**; // function *Phase_1* below.
How_Many_Times: = *How_Many_Times* - 1;
}
$\{Y_1, Y_2, \ldots, Y_k\}$: = Intersection (*Membership*); // Step_8.
$\{C_1, C_2, \ldots, C_k\}$: = Centers ($\{Y_1, Y_2, \ldots, Y_k\}$); // Step_9.
$\{Z_1, Z_2, \ldots, Z_k\}$: = Assign (*ALL*, $\{C_1, C_2, \ldots, C_k\}$); // Step_9.
Foreach $Z_i \in \{Z_1, Z_2, \ldots, Z_k\}$ do
{
Average-Distance: = Mean (Z_i); // Step_10.
Outliers: = Identify_outliers (*Average-Distance*, Zi); // Step_11.
}
Best_Soltuion: = Assign (*Outliers*, $\{C_1, C_2, \ldots, C_k\}$); // Step_12.

END

Function Phase_1(*ALL*)

Vars:
k_Genes : a set of selected genes;
REST : a set of remaining genes;
Threshold, *Smallest_Distance* : real variables;
MyTable : table;
Partition : a set of initial clusters;

Method:
k_Genes := random(k,*ALL*); // Step_1.
REST := *ALL* – *k_Genes*; // Step_1.
Threshold := Min (Distance (Xi, Xj)); // Step_2.
∀ Xi,Xj ∈ k_Genes
WHILE (*REST* != null) do
{
Smallest_Distance := Min(Distance(G, Xi)); // Step_3.
random G ∈REST
∀ Xi ∈ k_Genes
IF (*Smallest_Distance* > *Threshold*) // Step_4.
{
k_Genes := *k_Genes* - {X1 , X2};
k_Genes := *k_Genes* + {G};
{C_1, C_i, …, C_{k-1}} := generateClusters(*k_Genes*, *ALL*);
Foreach Ci ∈ {C_1, C_i, …, C_{k-1}} do
{
{*Dist*, *Ri*} := Max (Distance ($Center\_Ci, Ri$)) ;
∀ Ri ∈ Ci
MyTable := Stack({*Dist*, *Ri*});
}
Rp := Max(*MyTable*);
k_Genes := *k_Genes* + {*Rp*}; // Step_5.
Threshold := Min (Distance (Xi, Xj));
∀ Xi,Xj ∈ k_Genes
} // of IF
} // of WHILE
Partition := generateClusters(*k_Genes*, *ALL*); // Step_6.
Return *Partition*;

4 Experiments and results

To assess the effectiveness of the suggested methodology, an empirical study was carried out using large-scale gene expression datasets obtained through technologies like microarrays. These datasets provide a comprehensive view of the transcriptomic landscape, revealing which genes are upregulated or downregulated in specific biological samples. Performance is measured using certain metrics, such as the Davis-Bouldin index for quality evaluation and the running-time measure.

4.1 Datasets used in experiments

We conducted global gene expression profiling and pathway analysis on the hematopoietic stem cells (HSC) of 183 MDS patients in comparison with the HSC of 17 healthy controls in order to get insight into the molecular pathophysiology of the myelodysplastic syndromes (MDS). Immunodeficiency, apoptosis, and chemokine signaling are among of the most severely dysregulated gene pathways in early MDS, while deregulation of the DNA damage response and checkpoint pathways characterizes advanced MDS.

The aberrant pathways that have been found are probably essential to the MDS HSC phenotype and provide novel insights on the disorder's molecular etiology, which in turn opens up new avenues for therapeutic intervention [15].

In the research, expression datasets from bone marrow CD34+ cells of MDS taken from both healthy controls and MDS patients. Samples were hybridized to Affymetrix GeneChip Human Genome U133 Plus 2.0 arrays. Actually, 54675 genes are measured simultaneously over 200 of experiments (samples) resulting in 10 935 000 measurements. Expression datasets being used in the study are available in this website [16].

4.2 Performance metrics used in experiments

The most difficult aspect of clustering analysis is figuring out how effective the suggested approach is. Clustering results are often evaluated using validity indices. Additionally, processing high-dimensional datasets takes a lengthy time, which makes the running-time index crucial.

The Davis-Bouldin Index (DB Index) is a measure used in cluster analysis to evaluate the quality of clustering. It's named after two researchers, David L. Davies and Donald W. Bouldin, who introduced it in their paper "A Cluster Separation Measure". The DB Index evaluates clustering by considering both the within-cluster similarity and the between-cluster dissimilarity. It is calculated as the average similarity between each cluster and its most similar neighboring cluster, divided by the average dissimilarity between each cluster and its most dissimilar neighboring cluster. A lower DB Index value indicates better clustering, where clusters are more separated from each other and more compact within themselves. In practical terms, when using the DB Index, you would typically aim to find the clustering solution that minimizes the index value. However, like many clustering evaluation metrics, it should be used in conjunction with other measures and with a good understanding of the data and the context in which clustering is being applied.

The Davis-Bouldin Index (DB score) is formulated as follows: Let C_i denote the i^{th} Cluster, μ_i denote the centroid of cluster C_i, and σ_i denote the average distance from each point in cluster C_i to the centroid μ_i. The centroid μ_i and the dispersion σ_i can be calculated using various distance metrics, such as Euclidean distance. Then, for each pair of clusters C_i and C_j, the dissimilarity $d(C_i, C_j)$ is calculated as the distance between their centroids μ_i and μ_j. The DB Index for a set of clusters $\{C_1, C_2, \ldots, C_k\}$ is given by the formula:

$$DB\ Index = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}\left\{\frac{\sigma_i + \sigma_j}{d(C_i, C_j)}\right\}$$

In this formula:

- k is the total number of clusters.
- σ_i and σ_j are the average distances from points in clusters C_i and C_j to their respective centroids.
- $d(C_i, C_j)$ is the distance between the centroids of clusters C_i and C_j.

The DB Index essentially compares the within-cluster scatter to the between-cluster separation, aiming to find clusters that are both internally cohesive and well-separated from each other. A lower DB Index value indicates better clustering performance.

Running time remains a crucial concern, especially when handling massive amounts of highly dimensional data. Running-time score assesses how fast the clustering procedure moves along. This is an important indicator for biologist who must execute the method several times before obtaining an average result, particularly when it is stochastic.

In fact, the performance of the computer used also impacts the running-time score. This research used a personal computer (PC) with the following specifications:

- Intel Core: i5-6300U / total Cores: 2 / total threads: 4 / processor base frequency: 2.40 GHz / max turbo frequency: 3.00 GHz.
- RAM Size: 16Go DDR4 / RAM frequency: 2133 MHz.
- Hard Disk: 240Go SSD NVMe provide speeds up to 3500 MB/s.

4.3 Empirical results

Table 2 illustrates the Davis-Bouldin score and running-time score corresponding to different values of the variable "How_Many_Times". It's apparent that as the value of "How_Many_Times" increases, the Davis-Bouldin score remains relatively stable around 0.92, indicating consistent clustering quality. However, there's a noticeable increase in running time, suggesting potential scalability concerns with larger values of "How_Many_Times".

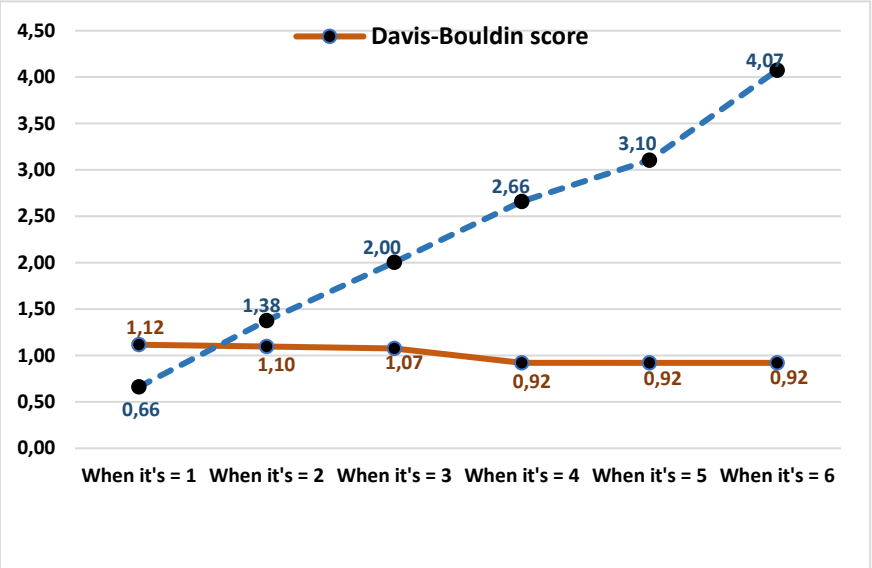

Figure 5: Empirical results corresponding to different values of "*How_Many_Times*"

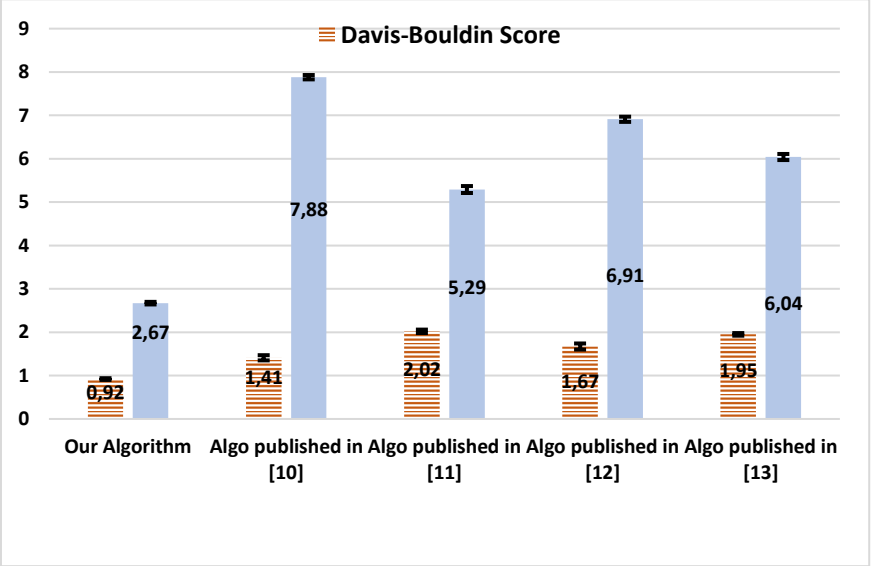

Figure 6: Outcomes of running-time and Davis-Bouldin scores

Table 2. Influence of different "How_Many_Times" values

How_Many_Times (see step_7)	Davis-Bouldin score	Running-time Score (in hours)
When it's = 1	1,12	0,66
When it's = 2	1,10	1,38
When it's = 3	1,07	2,00
When it's = 4	0,92	2,66
When it's = 5	0,92	3,10
When it's = 6	0,92	4,07

Figure 5 shows how both the Davis-Bouldin score and the running-time score fluctuate with different values of "How_Many_Times". A lower Davis-Bouldin score indicates better clustering quality. It's notable that as the "How_Many_Times" value increases, the Davis-Bouldin score consistently diminishes until the "How_Many_Times" value reaches 4, where Davis-Bouldin score stabilizes around 0.92, suggesting a consistent level of clustering quality. Figure 5 demonstrates also the running-time score across various values of the "How_Many_Times". It becomes evident that as the "How_Many_Times" value escalates, so does the running-time score, hinting at potential scalability issues with higher "How_Many_Times" values. The ideal setup for our algorithm aims to minimize both the Davis-Bouldin score and the running time. Referring to Table 2, it's evident that the optimal value for "How_Many_Times" should be 4.

To evaluate the effectiveness of the proposed method, we conducted an experimental study utilizing a genuine gene expression dataset accessible at [16]. Specifically, this dataset comprises 54,675 genes measured simultaneously across 200 experiments (samples), totaling 10,935,000 measurements to be grouped into 30 clusters. Algorithms [10], [11], [12], [13], including ours, are stochastic; each value in Table 3 represents the average of 10 runs.

Table 3: Performance measurement outcomes

Algorithms	Davis-Bouldin Score	Standard Deviation of DB	Running-Time Score	Standard Deviation of RT
Our Algorithm	0,92	0,02	2,67	0,03
Algo published in [10]	1,41	0,06	7,88	0,05
Algo published in [11]	2,02	0,04	5,29	0,08
Algo published in [12]	1,67	0,07	6,91	0,06
Algo published in [13]	1,95	0,03	6,04	0,07

Table 3 provides a comparative perspective in relation to the Davis-Bouldin score and the running-time score (in hours) attained by our proposed algorithm, alongside those of recently published ones referenced in the related-work section.

The error bars displayed on Figures 6 represent standard deviation values, which measure the spread of data relative to the mean. A low standard deviation suggests that the data points are closely around the mean, while a high standard deviation indicates that the data is more widely dispersed. Subsequently, the smaller the standard deviation, the better the result.

5 Discussion

Table 3 compares different algorithms based on their clustering quality and running-time score. Our algorithm stands out with the lowest Davis-Bouldin score of 0.92 with smaller standard deviation, suggesting superior clustering quality compared with those algorithms summarized in the related works section. In terms of running time, our algorithm again performs well with a score of 2,67 indicating relatively fast processing. Overall, our algorithm appears to strike a good balance between clustering quality and computational speed, outperforming all competitors in both metrics. We can indeed quantify this improvement using the following formulas:

0,92 / ((1,41 + 2,02 + 1,67 + 1,95) / 4) = 0,52 => an average 48% improvement in quality.

2,67 / ((7,88 + 5,29 + 6,91 + 6,04) / 4) = 0,40 => an average 60% reduction in running-time.

Furthermore, a low standard deviation indicates that the values obtained across multiple runs are very consistent, demonstrating the robustness of our algorithm.

In fact, the performance differences are primarily due to the adoption of an extremely-fast partitional strategy without initially focusing on quality, which was modest during the first phase. The trick is to iterate this first phase multiple times to enhance quality while maintaining minimal running time. This optimization approach reveals the optimal number of loops needed for quality improvement to plateau, as shown in Figure 5, with the optimal number being 4. Furthermore, we have reassigned the outliers as detailed in steps 10, 11 and 12 in order to improve the clustering quality.

In this experimental study, datasets on the scale of ten million measurements were used, requiring 16 GB of RAM and an Intel Core i5-6300U 2.40 GHz 2 Cores. In terms of scalability, our approach can handle even larger datasets extending to hundreds of millions of measurements, provided that the computational resources are upgraded. However, our approach may face potential limitations when applied to datasets approaching a billion measurements.

While cancer can spread rapidly in the human body, biologists recognize the need for computational tools that offer both speed and accuracy in the diagnosis and prognosis process. Actually, the proposed approach can accurately and quickly identify clusters of varying shapes, sizes, and densities by employing optimization techniques to tackle this NP-HARD challenge. The resulting clusters help identify co-expressed genes, pinpoint dysregulated

gene functions, and ultimately identify potential biomarkers for disease diagnosis.

6 Results validation

To validate the results, we need to use external validity metrics like F-measure. F-measure compares the obtained clusters to benchmark classes. If we have a reference partition P of the dataset (a benchmark likely based on existing domain knowledge), we can evaluate the clusters C by assessing their similarity to P using statistics such as F-measure. This measure combines the precision and recall concepts from information retrieval. We then compute the recall and precision for each class in the cluster as described in [17].

$$Recall(i, j) = \frac{n_{ij}}{n_i}$$

$$Precision(i, j) = \frac{n_{ij}}{n_j}$$

Were n_{ij} is the number of objects of class i that are in cluster j, n_j is the number of objects in cluster j, and n_i is the number of objects in class i. The F-measure of cluster j and class i is given by the following equation:

$$F(i, j) = \frac{2\,.\,Recall(i,j)\,.\,Precision(i,j)}{Precision(i,j) + Recall(i,j)}$$

The F-measure values range from 0 to 1, with higher values reflecting better clustering quality. An F-measure of 1 indicates that C is identical to P, representing an optimal solution.

We need a dataset where the number of classes is already known by experts. Reference [18] provides a dataset that tracks gene expression changes caused by smoking and its progression to cancer. This dataset includes 107 samples, with 58 from tumor tissues and 49 from non-tumor tissues. The samples come from 20 never smokers, 26 former smokers, and 28 current smokers, and each sample includes expression levels for 22,283 genes, amounting to a total of 2,384,281 measurements (107 samples x 22,283 genes).

After applying our proposed algorithm to this dataset, we achieved an F-measure of 0.9627, indicating that the resulting clusters are 96.27% similar to the classes previously identified by experts.

7 Conclusion

Functional modules are groups of genes that work together to perform specific biological functions or participate in common pathways. Clustering helps in identifying these modules by grouping genes with similar expression patterns. The assumption is that genes that behave similarly under certain conditions may be functionally related. Clustering analysis applied to gene expression data has proven to be a valuable tool for uncovering functional modules and understanding gene regulation in the context of genomic diseases, particularly cancer.

Actually, clustering research in gene expression datasets becomes challenging when dealing with high-dimensional datasets, where tens of thousands of genes are measured simultaneously over hundreds of experiments.

This paper provides an optimization approach that identifies potential biomarkers for diseases diagnosis through co-expression analysis. This approach has the ability to handle clusters with different shapes, sizes and densities, outperforming the other algorithms in terms of clustering quality and computational speed.

Code and datasets availability

The code and datasets supporting the findings of this study can be obtained from the corresponding author upon request.

Conflicts of interest

The authors declare no conflicts of interest.

Funding statement

This study did not receive any funding in any form.

References

[1] Arash Kianianmomeni. More light behind gene expression. Trends in Plant Science, (19)8: 488-490, 2014. https://doi.org/10.1016/j.tplants.2014.05.004

[2] Daniel Castro-Roa, Nikolay Zenkin. Methodology for the analysis of transcription and translation in transcription-coupled-to-translation systems in vitro. Methods, 86(1): 51-59, 2015. https://doi.org/10.1016/j.ymeth.2015.05.029

[3] Rui Dilão. The regulation of gene expression in eukaryotes: Bistability and oscillations in repressilator models. Journal of Theoretical Biology, 340(1): 199-208, 2014. https://doi.org/10.1016/j.jtbi.2013.09.010

[4] Lan Sun, Lingyue Gao, Yingxi Zhao, Yuqing Wang, Qianhui Xu, Yiru Zheng, Jiali Chen, He Wang, and Lihui Wang. Understanding and Targeting the Epigenetic Regulation to Overcome EGFR-TKIs Resistance in Human Cancer. Recent Patents on Anti-Cancer Drug Discovery, 18(4): 506-516, 2023. https://doi.org/10.2174/1574892818666221201145810

[5] Xiaoqing Peng, Wanxin Cui, Xiangyan Kong, Yuannan Huang, and Ji Li. DMR_Kmeans: Identifying Differentially Methylated Regions Based on k-means Clustering and Read Methylation Haplotype Filtering. Current Bioinformatics, 19(5): 490-501, 2024. https://doi.org/10.2174/0115748936245495230925112419

[6] Abootaleb Sedighi, Paul C.H. Li. Challenges and Future Trends in DNA Microarray Analysis. Comprehensive Analytical Chemistry, 63(1): 25-46, 2014.

https://doi.org/10.1016/B978-0-444-62651-6.00002-7

[7] Jinzeng Wang, Qi Lv, Xujuan Li, Ya Liu, Kang Liu, and Haiyun Wang. Post-transcriptional and translational regulation modulates gene co-expression behavior in more synchronized pace to carry out molecular function in the cell. Gene, 587(2): 163-168, 2016. https://doi.org/10.1016/j.gene.2016.04.055

[8] Alex Rodriguez, Alessandro Laio. Clustering by fast search and find of density peaks. Science, 344(6191): 1492-1496, 2014. https://doi.org/10.1126/science.1242072

[9] Ruijia Li, Xiaofei Yang, Xiaolong Qin, and William Zhu. Local gap density for clustering highdimensional data with varying densities. Knowledge-Based Systems, 184(1): 104905, 2019. http://dx.doi.org/10.1016/j.knosys.2019.104905

[10] Yizhang Wang, Di Wang, You Zhou, Xiaofeng Zhang, and Chai Quek. VDPC: Variational density peak clustering algorithm. Information Sciences, 621(1): 627-651, 2023. https://doi.org/10.1016/j.ins.2022.11.091

[11] Hassan Ismkhan. I-k-means−+: An iterative clustering algorithm based on an enhanced version of the k-means. Pattern Recognition, 79(1): 402-413, 2018. https://doi.org/10.1016/j.patcog.2018.02.015

[12] Yewang Chen, Shengyu Tang, Nizar Bouguila, Cheng Wang, Jixiang Du, and HaiLin Li. A fast-clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. Pattern Recognition, 83(1): 375-387, 2018. https://doi.org/10.1016/j.patcog.2018.05.030

[13] Masciari E, Mazzeo GM, and Zaniolo C. Analysing microarray expression data through effective clustering. Information Sciences, 262(1): 32-45, 2014. https://doi.org/10.1016/j.ins.2013.12.003

[14] Harun Pirim, Burak Ekşioğlu, Andy Perkins, and Cetin Yüceer. Clustering of high throughput gene expression data. Computers & Operations Research, 39(12): 3046-3061, 2012. https://doi.org/10.1016/j.cor.2012.03.008

[17] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi and Elvia M. Quiroz. Internal versus External cluster validation indexes. International Journal of Computers and Communications, 5(1): 27-34, 2011. https://www.naun.org/main/UPress/cc/20-463.pdf