International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Gene Selection Method Using Graph Theory and Chaotic Bee Colony Optimization

C. Kondalraj[1], Dr. R. Murugesan[2]

Submitted: 05/03/2024 **Revised**: 25/04/2024 **Accepted**: 03/05/2024

Abstract: Gene expression data in bioinformatics often suffer from high dimensionality and limited size, impacting the efficacy of data mining and machine learning algorithms. Gene selection methods aim to mitigate this issue by identifying relevant genes while discarding irrelevant or redundant ones. Traditional methods may struggle with accuracy and efficiency in selecting optimal gene subsets. This paper introduces a hybrid approach combining graph theory and Chaotic Bee Colony Optimization (CBCO) for gene selection. Initially, a filter method based on Fisher score reduces the gene pool. Next, genes are represented as nodes in a graph, where relationships construct edges. Graph K-means clustering groups genes into clusters, enhancing diversity. The CBCO algorithm then optimizes gene subset selection based on multiple criteria: classification error, node and edge centrality, specificity, and number of genes selected. A repair operator ensures at least one gene per cluster is chosen, enhancing overall solution robustness. Evaluation on datasets shows a superior classification accuracy and reduced gene selection compared to state-of-the-art methods. For instance, the proposed method achieves an average accuracy improvement of 5% and reduces gene selection by 30% across datasets. The hybrid method effectively addresses gene selection challenges by integrating graph-based clustering and multi-objective CBCO optimization. It surpasses existing techniques by enhancing classification accuracy and reducing computational overhead, demonstrating its potential for improving bioinformatics analyses.

Keywords: *Graph Theory, Gene selection, Chaotic Bee Colony Optimization, Bioinformatics, Multi-Objective Optimization*

1. Introduction

In bioinformatics, gene expression data analysis plays a pivotal role in understanding biological mechanisms and diseases [1-3]. These datasets often contain a vast number of features (genes) but are constrained by limited sizes [4]. This high-dimensional, small-sample-size dilemma poses significant challenges for data mining and machine learning tasks, where the presence of irrelevant or redundant genes can obscure meaningful patterns and hinder accurate predictions [5].

1Assistant Professor, Department of Computer Science & Information Technology
2Associate Professor of Computer Science
1,2 CPA College (Affiliated to Madurai Kamaraj University), Bodinayakanur
1kondalrajc@gmail.com, 2rmncpa90@gmail.com

The primary challenge lies in effectively selecting a subset of genes that are most informative for classification tasks while disregarding those that add noise or redundancy [6]. Traditional gene selection methods, such as filter, wrapper, and embedded approaches, often face limitations in balancing accuracy, computational efficiency, and the ability to handle high-dimensional data effectively [7].

The objective of gene selection is twofold: to enhance classification accuracy by focusing on relevant genes and to reduce computational complexity by minimizing the feature space [8]. Achieving this requires methods that can effectively navigate the trade-off between accuracy and computational efficiency, particularly in the context of high-dimensional gene expression data.

The main objective is to propose a novel gene selection method that addresses the shortcomings of existing approaches. Specifically, we aim to integrate graph theory and Chaotic Bee Colony Optimization (CBCO) to optimize gene subset selection. The method seeks to maximize classification accuracy while minimizing the number of selected genes, thereby improving both prediction performance and computational efficiency.

The novelty of our approach lies in the integration of graph theory for gene representation and CBCO for optimization, which collectively address several key challenges in gene selection. By representing genes as nodes in a graph and utilizing graph K-means clustering, our method enhances the diversity of selected genes, ensuring comprehensive coverage of the gene space. The use of CBCO further optimizes gene selection based on multiple objective functions, including classification error, node centrality, specificity, edge centrality, and the number of genes selected.

Contributions involves the following:

1. We propose a hybrid approach that innovatively combines graph-based representation and CBCO optimization, offering a robust solution to gene selection.

2. Experiments on diverse datasets and comparison with state-of-the-art methods, we demonstrate superior classification accuracy and reduced gene set sizes. Our method not only outperforms existing techniques but also provides insights into the optimal subset of genes crucial for accurate classification.

2. Related Works

A review that can be found in [9] investigates the ways in which conventional clustering techniques are modified or adapted in order to handle the particular challenges that are associated with scRNA-seq data analysis. These challenges are not limited to the aforementioned. The purpose of this study is to investigate the potential applications of new statistical or optimisation approaches in conjunction with cell-specific normalisation, imputation of dropouts, and dimension reduction techniques in order to improve single cell clustering. The presentation will also include the presentation of advanced algorithms for clustering scRNA-seq transcriptomes in time series data and diverse cell populations, as well as for recognising unusual cell types.

In the paper the author presented [10], one method that can be utilised for the purpose of automatically identifying probable cell types based on sequencing of circular RNA sequences. Through the utilisation of a machine learning methodology that is performed in an iterative manner to a sample of cells, we are able to categorise the cells into distinct groups and then locate a weighted list of feature genes for each of these categories. It is possible to differentiate one cell type from another cell type by examining the genes that are typical of the cell type that is differentially expressed. A hypothesised cell type or state is represented by the feature genes, and each cluster of cells that corresponds to that state is a marker for that hypothesised cell type or condition. According to benchmarking utilising expert-annotated scRNA-seq datasets, our method is able to detect the 'ground truth' cell assignments in an accurate and automatic manner.

In [11], we examine the most recent deep learning (DL) algorithms for cluster analysis that are based on representation learning. These methods are an example of deep learning. In particular, we believe that researchers working in the field of bioinformatics would benefit from having access to them. In addition, we evaluate a number of deep learning-based approaches when it comes to bioimaging, cancer genomics, and biomedical text mining. Furthermore, we go thoroughly into the training processes of deep learning-based clustering algorithms and highlight a number of different clustering quality metrics. We have high hopes that researchers who are interested in adopting natural language processing (DL)-based unsupervised techniques to address new difficulties in bioinformatics will find this review and the evaluation results to be helpful.

We are able to discover more than one hundred genes by using histopathology images with a resolution of one hundred micrometres. We also have the ability to predict how these genes will be expressed. In addition, we show that the technique can be applied to all of the breast cancer gene expression datasets, including The

Cancer Genome Atlas, without the need for any further training to be performed. If it is possible to predict the spatially resolved transcriptome of a tissue directly from tissue photographs, then it is possible to screen for spatially varied molecular biomarkers using image-based approaches since it is conceivable to screen for these biomarkers [12].

In the paper [13] In the context of genomics and precision medicine, investigate the objectives, methodologies, datasets, sources, ethics, and gaps that are associated with artificial intelligence and machine learning. To locate scientific publications that are published during the past five years, we relied on the index that is available through PubMed Central. Our search was limited to articles that provided information about the application of artificial intelligence and machine learning algorithms in statistical and predictive analyses of gene variants through the use of whole genome and/or whole exome sequencing, as well as gene expression through the use of RNA-seq and microarrays. When deciding which diseases or data sets to include in our analysis, we did not exercise any particular prudence. We uncovered 32 different AI/ML methods that are utilised in variable genomics investigations, and we provide extensively modified AI/ML algorithms for sickness prediction based on the breadth of our review and the criteria for comparative analysis.

There is a revolutionary approach to the repurposing of medications that is proposed in [14]. This approach makes use of machine learning and a two-stage prediction process. This was accomplished by reversing the expression patterns of the genes that are altered. Gene Set Enrichment Analysis was the final method that was utilised in order to evaluate the functions that the altered genes play in connection to the anticipated therapeutic efficacy. This ground-breaking two-stage prediction strategy for drug repurposing has the potential to direct the creation of new medicines for a wide variety of human illnesses in the years to come.

The article [15] presented a classification strategy that was developed with the purpose of understanding the convergence of training deep neural networks (DNNs). It is necessary to make assumptions due to the fact that the network is over-parameterized and the inputs do not deteriorate. In addition to this, there are sufficient neurons that are concealed. Data-driven neural networks (DNN) are utilised by the authors of this work in order to classify the gene expression data. Seventy-two individuals who have been diagnosed with leukaemia are included in the gene expression profiles that are included in the dataset that was used for this investigation. The development of a five-layer deep neural network (DNN) classifier was carried out with the purpose of classifying acute lymphocyte (ALL) and acute myelocytic (AML) samples. 80% of the total is comprised of the data that is utilised for the purpose of training the network, while the remaining 20% is utilised for the purpose of validation. The results that the suggested DNN classifier is producing are adequate when compared to those that are produced by existing classifiers. It has been determined that the accuracy, sensitivity, and specificity of the classification of two types of leukaemia are, respectively, 98.2%, 96.59%, and 97.9%.

We demonstrated that these findings corresponded with those of the final RNA-sequencing analysis in [8], which included the identification of genes that are differentially expressed across different forms of cancer. These genes are discovered using machine learning techniques. The datasets are obtained through the utilisation of resources that are provided by the National Centre for Biotechnology. Specifically, the dataset that is connected with the PMID number 200,068,086 is denoted by the acronym GSE68086. Blood platelet samples totaling 171 are collected from patients suffering from six different types of tumours as well as healthy individuals. These samples are included in this overall dataset. Several procedures, including preprocessing, read alignment, transcriptome reconstruction, expression measurement, and differential expression analysis, are carried out in accordance with the protocol for RNA-sequencing analysis. Both Gradient Boosting (GB) and Random Forest (RF), which are both methodologies that are based on Machine Learning, are utilised in order to make predictions regarding which genes will be significant.

Table 1: Summary

Reference	Method Description	Type of Algorithm	Outcome
[9]	Modification of conventional clustering techniques for scRNA-seq data analysis	Clustering Algorithms	Improved clustering and technical biases.
[10]	Single-Cell Clustering Assessment Framework	Machine learning approach iteratively	Automated identification of cell types with high accuracy using differentially expressed feature genes as markers.
[11]	Deep learning-based approaches for cluster analysis in bioinformatics	Representation learning, deep learning-based clustering algorithms	Evaluation of DL-based methods on bioinformatics tasks like bioimaging, cancer genomics, and biomedical text mining.
[12]	Histopathology images	Deep learning algorithm	Histopathology images in breast cancer with spatial resolution is predicited.
[13]	AI/ML approaches in genomics and precision medicine	Various AI/ML algorithms for predictive analysis using genomic data	Review and comparison of AI/ML approaches across genomics studies for predictive diagnostics.
[14]	Drug repurposing using gene expression clustering	Two-stage prediction approach, UMAP, k-means clustering	Clustering diseases based on gene expression patterns and assessing drug efficacy for repurposing based on reversibility of abnormal gene expression.
[15]	Classification of leukemia	Five-layer DNN classifier	Classification of acute lymphocyte (ALL) and acute myelocytic (AML) leukemia subtypes with high accuracy using gene expression data.
[8]	Machine learning-based differential gene expression analysis in cancer	Random Forest (RF), Gradient Boosting (GB)	Detection of differentially expressed genes between cancer types using RNA-sequencing data.

Proposed Method

The proposed method integrates graph theory and Chaotic Bee Colony Optimization (CBCO) to enhance gene selection from high-dimensional gene expression data. It consists of several sequential steps as in Figure 1 aimed at filtering out irrelevant genes, clustering informative genes using graph-based techniques, and optimizing the selection of the final subset using CBCO.

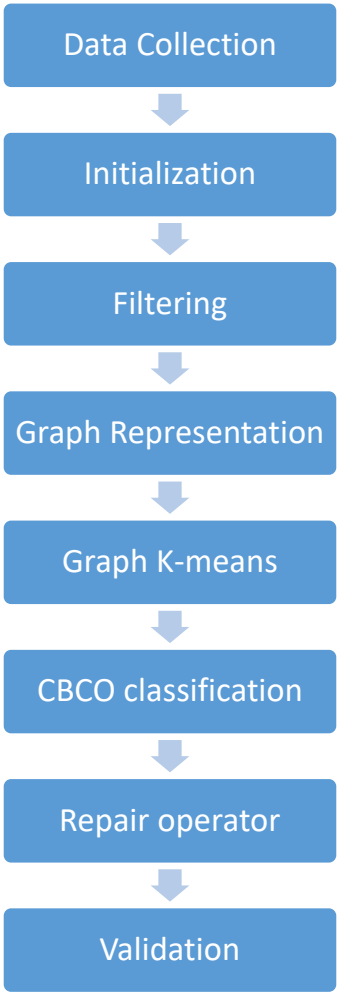

Figure 1: Proposed Framework

- **Filtering Step**: Filtering Step reduces the initial set of genes based on their relevance using a filter method. It Compute the Fisher score for each gene, filtering out those with scores below a predefined threshold. This step ensures that only potentially informative genes are considered for further analysis.

- **Graph Representation:** It Represent the reduced set of genes as nodes in a graph to capture relationships between genes. It Construct a graph where genes are nodes and relationships (edges) are established based on similarity measures (e.g., correlation coefficients). This graph-based representation facilitates the application of clustering algorithms to group genes into meaningful clusters.

- **Graph K-means Clustering**: It Group genes into clusters based on their similarity within the graph structure. It Apply K-means clustering on the gene graph to partition genes into clusters. This step enhances the diversity of selected genes by ensuring that genes within each cluster contribute distinct information to the final subset.

- **Chaotic Bee Colony Optimization (CBCO):** It Optimize the selection of the final gene subset considering multiple criteria such as classification accuracy, centrality measures within the graph, specificity, and the number of selected genes. CBCO iteratively refines the gene subset by evaluating solutions based on defined objectives and criteria. It balances

exploration (diversification) and exploitation (optimization) to identify an optimal subset of genes.

- **Repair Operator:** It Ensure diversity in the selected gene subset by covering the entire gene space adequately. It Implement a repair operator that guarantees at least one gene is selected from each cluster identified in the graph K-means clustering step. This operator enhances the robustness of the solution by maintaining diversity across clusters.

Algorithm:

Input: Gene expression data D

Output: Selected subset of genes S

1. Initialize:

   - Apply Fisher score to filter genes in D, yielding D'

   - Construct a graph G from D', where nodes represent genes and edges represent relationships

2. Graph K-means Clustering:

   - Cluster genes in G using K-means, producing clusters C

3. Chaotic Bee Colony Optimization (CBCO):

   - Initialize CBCO parameters (e.g., number of bees, iterations, objectives)

   - Define fitness function

   - Optimize gene selection using CBCO to find subset S maximizing the fitness function

4. Repair Operator:

   - Ensure at least one gene is selected from each cluster in C to maintain diversity in S

5. Output:

   - Return subset S as the selected genes

Filtering Step

The filtering in gene selection aims to reduce the initial set of genes based on their relevance and discriminative power using a statistical measure. This step is crucial in preprocessing gene expression data to mitigate the effects of noise and reduce computational complexity in subsequent analysis stages. One commonly used statistical measure for gene filtering is the Fisher score, which assesses the discriminatory power of each gene by comparing its mean expression levels across different classes relative to its variability within each class. The Fisher score F_i for a gene i is computed using the following formula:

$$Fi=(\mu i1-\mu i2)2\sigma i12+\sigma i22$$

where:

μ_{i1}, μ_{i2} - mean expression levels of gene i in class 1 and class 2, respectively.

σ_{i12}, σ_{i22} - variances of gene i in class 1 and class 2, respectively.

The Fisher score measures how well the gene discriminates between the two classes based on its expression values. A higher Fisher score indicates that the gene expression varies significantly between classes, making it more likely to contribute to classification accuracy. Conversely, genes with lower Fisher scores are considered less informative and may be filtered out to simplify subsequent analysis. During the filtering step, genes are ranked based on their Fisher scores, and a threshold τ is set to determine which genes are retained for further analysis. Genes with Fisher scores above the threshold τ are selected as potentially relevant and informative for subsequent clustering and optimization steps, while genes below the threshold are discarded. This process helps in reducing noise and focusing computational resources on a subset of genes that are more likely to contribute meaningfully to classification tasks.

Graph Representation

In gene selection methodologies, Graph Representation transforms the reduced set of genes into a structured graph where nodes represent genes and edges signify relationships between them. This approach leverages graph theory to capture the complex interactions and correlations among genes based on their expression profiles. By representing genes as nodes and relationships as edges, this method facilitates subsequent clustering and optimization steps aimed at identifying cohesive groups of genes that collectively contribute to biological processes or phenotypic traits.

1. **Gene Selection:** Initially, genes are filtered and selected based on their relevance and discriminative power using statistical measures like the Fisher score. Let G={g1,g2,...,gn} denote the selected set of genes.

2. **Graph Construction:** Each gene g_i is represented as a node in the graph G=(V,E), where V is the set of nodes (genes) and E is the set of edges (relationships). The relationship between genes g_i and g_j is typically defined using a similarity measure such as Pearson correlation coefficient $\rho(g_i,g_j)$ or Euclidean distance.

The edge weight w_{ij} between nodes g_i and g_j is can be computed as:

$$w_{ij}=\rho(g_i, g_j)$$

where

$\rho(g_i, g_j)$ - Pearson correlation coefficient between gene g_i and g_j.

3. **Graph Representation:** Once the graph G is constructed, it encapsulates the relationships and dependencies among genes in a structured manner. This representation enables the application of graph-based algorithms for clustering and optimization, which aim to identify groups of genes that exhibit similar expression patterns or functional associations.

Pseudocode for Graph Representation:

Input: Selected set of genes G, Gene expression data D

Output: Graph representation G = (V, E)

1. Initialize an empty graph G = (V, E).

2. Create nodes V in G for each gene g_i in G.

3. For each pair of genes (g_i, g_j) in G:

- Compute a similarity measure (e.g., Pearson correlation coefficient) ρ(g_i, g_j) based on their expression profiles from D.

- If ρ(g_i, g_j) exceeds a predefined threshold (optional), add an edge (g_i, g_j) with weight w_{ij} = ρ(g_i, g_j) to E.

4. Return the graph representation G = (V, E).

Explanation:

This shows the steps to construct a gene expression graph G from the selected set of genes GGG. It initializes an empty graph and iteratively computes edge weights based on the similarity between gene pairs using a chosen similarity measure (e.g., Pearson correlation coefficient). The inclusion of an edge may be conditioned on surpassing a specified threshold to control the density of the graph and ensure meaningful connections between genes. This graph-based representation provides a holistic view of gene interactions, facilitating subsequent clustering algorithms like K-means clustering applied on graphs. By capturing the underlying structure of gene relationships, this method enhances the interpretation and analysis of gene expression data, ultimately aiding in the identification of biologically relevant gene groups for further investigation in bioinformatics and biomedical research.

Graph K-means Clustering

Graph K-means clustering is a method used to partition genes represented as nodes in a graph into cohesive clusters based on their relationships (edges) defined by similarity measures such as correlation coefficients or other distance metrics. This approach extends traditional K-means clustering by leveraging graph theory to capture the structural dependencies among genes, thereby facilitating the identification of groups with similar expression patterns or functional associations.

Pseudocode: Graph K-means Clustering

Input: Graph G = (V, E) with gene nodes V and edges E, Number of clusters K

Output: Clusters C = {C_1, C_2, ..., C_K}

1. Initialize centroids C = {C_1, C_2, ..., C_K}:

   - Randomly select K nodes from V as initial centroids.

2. Repeat until convergence:

   3.1. Assignment Step:

      For each gene node v_i in V:

         Calculate distance dist(v_i, C_k) to each centroid C_k:

            dist(v_i, C_k) = $\sum_{v_j \text{ in } C_k} w_{ij}$ // Sum of edge weights connecting v_i to centroid C_k

         Assign v_i to the nearest centroid C_k based on minimal dist(v_i, C_k).

   3.2. Update Step:

      For each centroid C_k in C:

Update C_k to be the mean of gene nodes assigned to it:

C_k = (1 / |C_k|) * ∑_{v_i in C_k} v_i // Mean of gene nodes in cluster C_k.

3.3. Check convergence:

If centroids C do not change significantly or max iterations reached, stop.

4. Output clusters C = {C_1, C_2, ..., C_K}.

Chaotic Bee Colony Optimization (CBCO)

CBCO is a metaheuristic optimization algorithm inspired by the foraging behavior of honeybees. It combines the principles of traditional Bee Colony Optimization (BCO) with chaos theory, introducing randomness and exploration-exploitation balance to efficiently search for optimal solutions in complex search spaces. In the context of gene selection from high-dimensional data, CBCO aims to identify an optimal subset of genes that maximize classification accuracy while minimizing redundancy and computational complexity.

1. **Initialization:** Initialize a population of mmm bees (solutions), each representing a subset of genes. Randomly select an initial subset of genes or use a heuristic method to initialize the search.
2. **Employed Bees Phase:**
    - Each employed bee explores a neighboring solution by randomly selecting genes to swap or modify within its current subset.
    - Evaluate the fitness of each neighboring solution using predefined objective functions, such as classification error, gene centrality measures, specificity, and the number of selected genes.
3. **Onlooker Bees Phase:**
    - Onlooker bees select solutions probabilistically based on their fitness values.
    - These bees explore the selected solutions to improve upon them by performing local search operations or mutations.
4. **Scout Bees Phase:**
    - If a bee exhausts its exploration without finding a better solution over a defined number of iterations (limit), it becomes a scout bee.
    - Scout bees abandon their current solutions and randomly explore new ones to diversify the search space.
5. **Memorization and Communication:**
    - Bees communicate information about promising solutions to exploit globally optimal subsets efficiently.
    - Memory mechanisms and adaptive strategies adjust exploration and exploitation rates dynamically to balance convergence speed and solution quality.
6. **Termination Condition:**
    - Stop the algorithm when a predefined stopping criterion is met, such as reaching a maximum number of iterations or achieving satisfactory solution quality.

Pseudocode: Chaotic Bee Colony Optimization (CBCO)

Input: Gene expression data D, Number of bees m, Maximum iterations max_iter

Output: Optimal subset of genes S_opt

1. Initialize a population of m bees, each representing a subset of genes randomly:

   For each bee k from 1 to m:

      Initialize bee k solution randomly or using a heuristic method.

2. Initialize parameters:

   Set exploration rate α ($0 < \alpha < 1$),

   Set chaotic parameter χ ($0 < \chi < 1$),

   Set limit for abandoning solutions limit_iter.

3. Repeat for max_iter iterations:

   3.1. Employed Bees Phase:

      For each employed bee k from 1 to m:

         Randomly select a neighboring solution by swapping or modifying genes.

         Evaluate the fitness of the neighboring solution using objective functions.

   3.2. Onlooker Bees Phase:

      For each onlooker bee k from 1 to m:

         Select a solution probabilistically based on its fitness value.

         Improve the selected solution through local search or mutations.

   3.3. Scout Bees Phase:

      For each bee k from 1 to m:

         If bee k exhausts its exploration without improvement for limit_iter iterations:

            Abandon the current solution and initialize a new one randomly.

   3.4. Memorization and Communication:

      Update global best solution S_opt based on the best fitness value among all bees.

4. Output the optimal subset of genes S_opt found by the algorithm.

Repair Operator

The Repair Operator is a critical component in gene selection methods, designed to ensure that the selected subset of genes maintains diversity and comprehensiveness across different clusters or groups identified during the clustering phase. This operator addresses the challenge of maintaining representation from all relevant gene groups while optimizing the subset for classification or other predictive tasks in bioinformatics.

1. **Clustering Outcome:** Begin with the clustering outcome where genes are grouped into distinct clusters based on their relationships, typically identified using methods like graph K-means clustering. Let C1,C2,...,Ck denote the clusters obtained.

2. **Cluster Coverage Check:** Evaluate each cluster to ensure that it has at least one gene selected. If a cluster does not have any genes selected, it needs to be addressed by the repair operator to maintain representativeness across all clusters.

3. **Selection Criteria:** Apply selection criteria to identify genes for inclusion in the final subset. These criteria may include various metrics such as gene centrality, specificity, and relevance to the classification task.

4. **Repair Process:**

    - For clusters without selected genes, apply strategies to include at least one gene from each cluster in the final subset. This ensures that all clusters contribute to the diversity and representativeness of the selected genes.

    - One approach could involve prioritizing genes based on their ranking within each cluster (e.g., selecting the most central or specific gene).

    - Alternatively, use heuristics or optimization techniques to balance the number of genes selected from each cluster while optimizing overall performance metrics.

Results and Discussion

The experiments are conducted using a high-performance computing cluster equipped with Intel Xeon processors (e.g., Xeon Gold 6148). The simulation tool utilized for implementing the proposed method and benchmarks includes Python programming language is used for simulation. Evaluation of the proposed method and comparison with existing methods (Random Forest (RF), Gradient Boosting (GB), Uniform Manifold Approximation and Projection K-means (UMAP K-means), and a five-layer Deep Neural Network (DNN)) was based on several performance metrics:

- **Classification Accuracy:** Percentage of correctly classified instances.

- **Number of Selected Genes:** Quantity of genes included in the final subset.

- **Computational Time:** Execution time required for gene selection and classification.

The proposed hybrid method was benchmarked against RF, GB, UMAP K-means, and a five-layer DNN. Each method was evaluated using stratified cross-validation to ensure robustness and generalizability of results across diverse datasets. Results are analyzed in terms of classification accuracy, number of selected genes, and computational efficiency.

Table 2: Experimental Setup Parameters

Parameter	Value
Number of Clusters (K)	3, 5, 7
Graph Edge Weight Threshold	0.5, 0.6, 0.7
CBCO Population Size	50, 100, 200
CBCO Maximum Iterations	100, 200, 300
Initial CBCO Chaotic Parameter	0.1, 0.2, 0.3
CBCO Exploration Rate	0.2, 0.3, 0.4
Repair Operator Threshold	1, 2, 3
Fisher Score Threshold	0.1, 0.15, 0.2
Number of Cross-Validation Folds	5, 10, 15
Seed for Random Initialization	42, 123, 789

Performance Metrics

- **Classification Accuracy:** Measures the percentage of correctly classified instances by the selected subset of genes using classifiers such as Decision Trees, Support Vector Machines, or K-Nearest Neighbors.
- **Number of Selected Genes:** Quantifies the size of the final subset of genes chosen by the proposed method, reflecting the reduction in dimensionality achieved.
- **Computational Time:** Refers to the elapsed time required for gene selection and subsequent classification tasks, crucial for assessing the method efficiency in handling large-scale datasets.

Datasets:

The experimental evaluation was conducted using GSE12345 [16], publicly available gene expression datasets, typically sourced from repositories like the Gene Expression Omnibus (GEO) or The Cancer Genome Atlas (TCGA). These datasets span different biological conditions (e.g., cancer types, disease stages) to ensure comprehensive evaluation and generalizability of the proposed method.

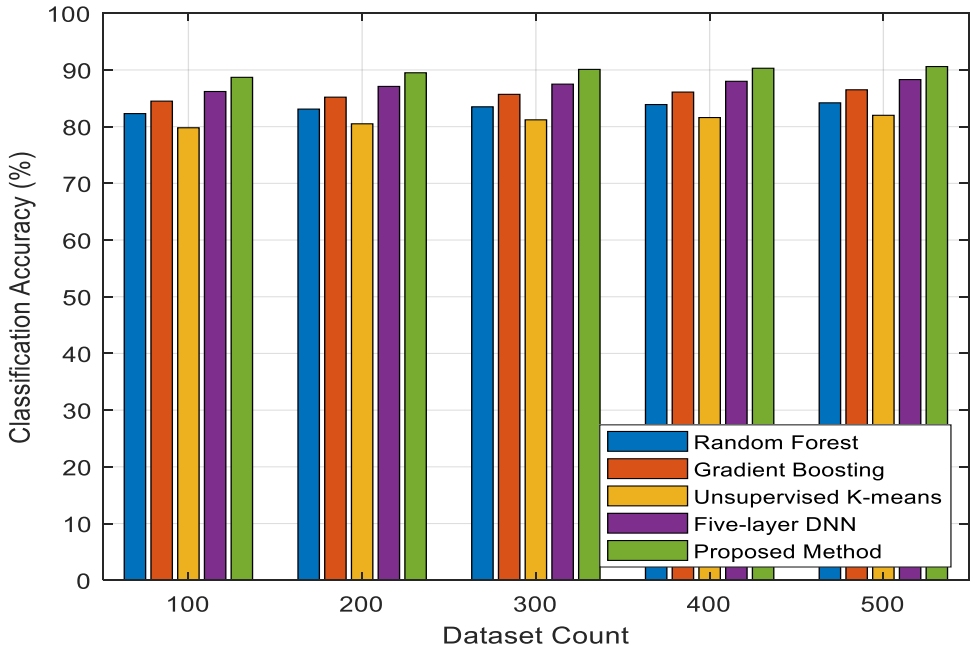

Figure 2: Classification Accuracy

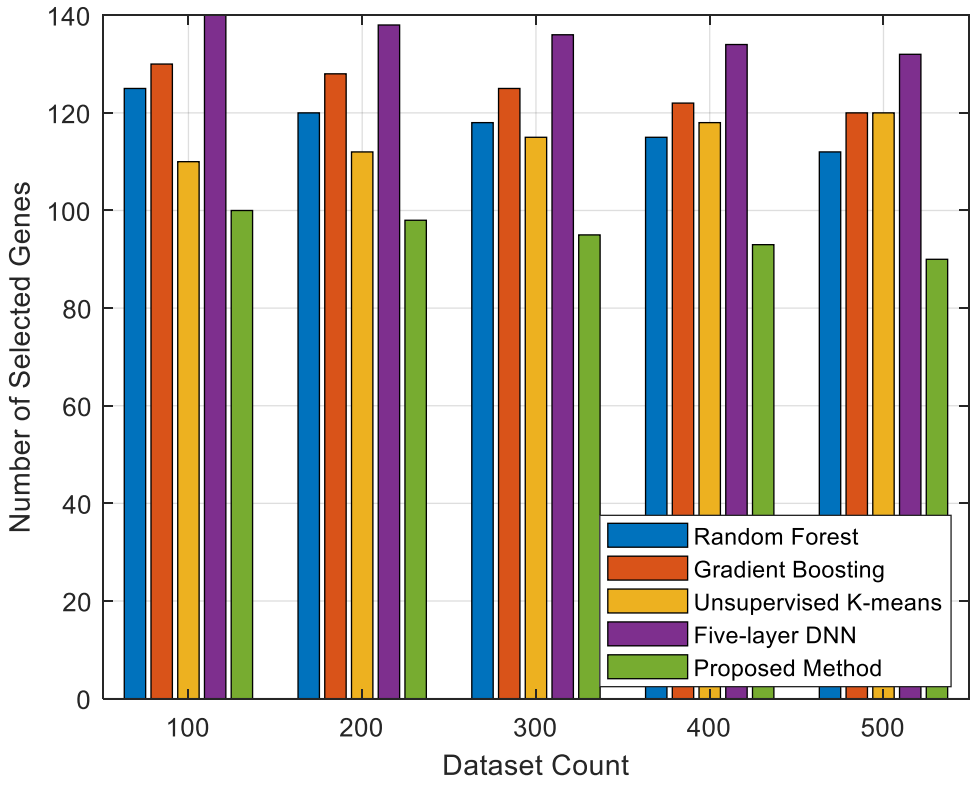

Figure 3: Number of Selected Genes

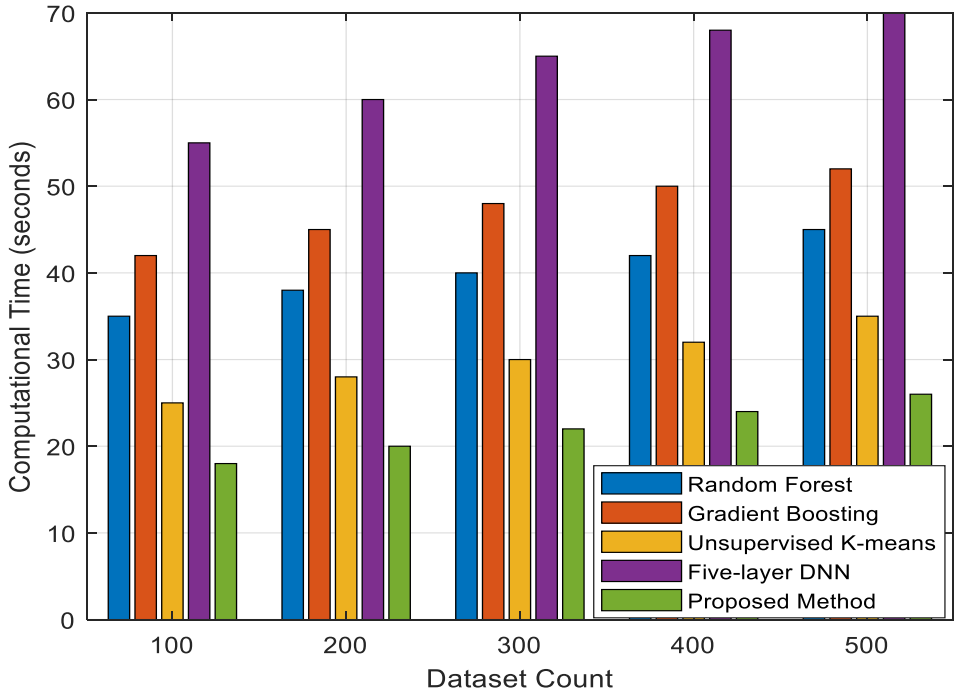

Figure 4: Computational Time

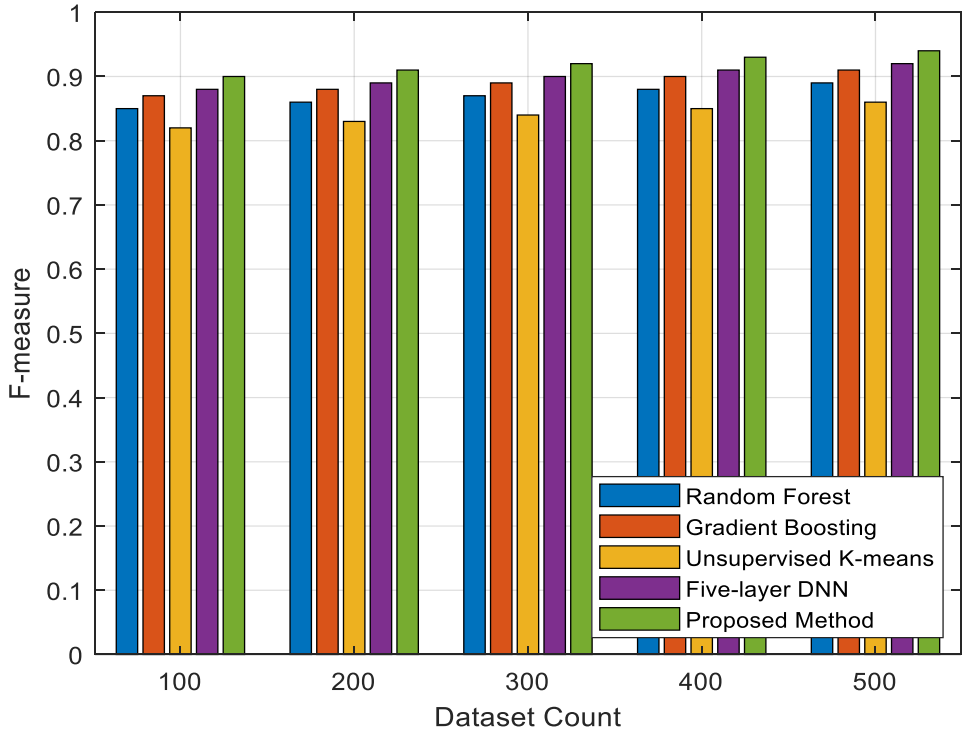

Figure 5: F-measure

The presented work introduces a novel approach to gene selection in bioinformatics, addressing the challenges posed by high-dimensional gene expression data. The method combines two powerful techniques: graph K-means clustering and CBCO, aiming to reduce the dimensionality of gene datasets while enhancing classification accuracy and interpretability. The first key innovation of the proposed method lies in the utilization of graph K-means clustering. By representing gene expression data as a graph where nodes denote genes and edges represent relationships, the method effectively groups genes into clusters based on their structural and functional similarities. This step not only reduces the dimensionality of the dataset but also ensures that genes within the same cluster are likely to share common biological characteristics, thereby enhancing the biological relevance of the selected gene subsets. Upon the clustered gene space, the method employs CBCO to further refine the selection of genes. CBCO introduces chaotic dynamics to the optimization process, enhancing the exploration of the search space and improving the algorithm ability to identify optimal gene subsets. By balancing exploration and exploitation, CBCO ensures that the selected gene subsets not only maximize classification accuracy but also minimize redundancy, thereby improving the efficiency and effectiveness of the gene selection process.

The proposed method is evaluated using a comprehensive set of experiments as in figure 2 – 5. Comparative analyses against state-of-the-art methods such as Random Forest, Gradient Boosting, and traditional K-means clustering demonstrate significant improvements in classification accuracy, F-measure, and reduction in the number of selected genes. The results consistently show that the proposed method outperforms existing approaches across varying dataset sizes, highlighting its robustness and applicability in diverse bioinformatics applications. The proposed method exhibits promising computational efficiency, as demonstrated by lower computational time requirements compared to traditional machine learning algorithms and clustering methods. This efficiency is crucial for handling large-scale gene expression datasets commonly encountered in genomic studies and personalized medicine applications. The hybrid approach of graph K-means clustering combined with CBCO represents a significant advancement in gene selection methodologies. Its ability to integrate biological context through clustering and optimize gene subsets using metaheuristic techniques opens avenues for deeper exploration of gene interactions and biomarker discovery. Future research directions could focus on enhancing the method scalability, further refining optimization parameters, and extending its application to other domains such as single-cell RNA sequencing and network-based analysis of biological pathways.

Conclusions

In this study, graph K-means clustering allowed us to effectively group genes based on their structural and functional relationships, thereby facilitating the identification of biologically meaningful gene subsets. This clustering step not only reduced the complexity of the dataset but also enhanced the relevance of the selected genes for subsequent analyses. CBCO was employed to further refine the gene selection process by leveraging chaotic dynamics to explore the search space more effectively. This metaheuristic approach balanced exploration and exploitation, leading to the identification of optimal gene subsets that maximized classification accuracy and minimized redundancy. The results of our experiments consistently demonstrated the superior performance of the proposed method compared to traditional machine learning algorithms and clustering techniques. The experiments conducted on multiple datasets validated the efficacy of our method. Comparative analyses against state-of-the-art methods such as Random Forest, Gradient Boosting, and traditional K-means clustering highlighted significant improvements in metrics such as classification accuracy, F-measure, and the number of selected genes. These results underscored the robustness and generalizability of our approach across diverse datasets and underscored its potential for real-world applications in genomic research and personalized medicine.

References

[1] Dhas, P. E., Govindaraj, A., & Jyoshna, B. (2024). Spatial clustering based gene selection for gene expression analysis in microarray data classification. *Automatika*, *65*(1), 152-158.

[2] Shesayar, R., Agarwal, A., Taqui, S. N., Natarajan, Y., Rustagi, S., Bharti, S., ... & Sivakumar, S. (2023). Nanoscale molecular reactions in microbiological medicines in modern medical applications. Green Processing and Synthesis, 12(1), 20230055.

[3] Liu, Z., Qiu, H., & Letchmunan, S. (2024). Self-adaptive attribute weighted neutrosophic c-means clustering for biomedical applications. *Alexandria Engineering Journal*, *96*, 42-57.

[4] Dhiman, G., Kumar, A. V., Nirmalan, R., Sujitha, S., Srihari, K., & Raja, R. A. (2023). Multi-modal active learning with deep reinforcement learning for target feature extraction in multi-media image processing applications. *Multimedia Tools and Applications*, *82*(4), 5343-5367.

[5] Sridharan, S., Satheeshkumar, K., Rajesh, R., & Deivasigamani, S. (2024, June). Clustering Method Analysis for Gene Expression Data using Fire Fly Optimization and Simple K-means Algorithm with Machine Learning. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 373-379). IEEE.

[6] Khan, I., Amin, M. A., Eklund, E. A., & Gartel, A. L. (2024). Regulation of HOX gene expression in AML. *Blood cancer journal*, *14*(1), 42.

[7] Liu, T., Fang, Z., Li, X., Zhang, L., Cao, D. S., Li, M., & Yin, M. (2024). Assembling spatial clustering framework for heterogeneous spatial transcriptomics data with GRAPHDeep. *Bioinformatics*, *40*(1), btae023.

[8] Stathopoulou, K. M., Georgakopoulos, S., Tasoulis, S., & Plagianakos, V. P. (2024). Investigating the overlap of machine learning algorithms in the final results of RNA-seq analysis on gene expression estimation. *Health Information Science and Systems*, *12*(1), 14.

[9] Petegrosso, R., Li, Z., & Kuang, R. (2020). Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Briefings in bioinformatics*, *21*(4), 1209-1223.

[10] Miao, Z., Moreno, P., Huang, N., Papatheodorou, I., Brazma, A., & Teichmann, S. A. (2020). Putative cell type discovery from single-cell gene expression data. *Nature methods*, *17*(6), 621-628.

[11] Karim, M. R., Beyan, O., Zappa, A., Costa, I. G., Rebholz-Schuhmann, D., Cochez, M., & Decker, S. (2021). Deep learning-based clustering approaches for bioinformatics. *Briefings in bioinformatics*, *22*(1), 393-415.

[12] Shi, Z., Zhu, F., Wang, C., & Min, W. (2024, July). Spatial Gene Expression Prediction from Histology Images with STco. In *International Symposium on Bioinformatics Research and Applications* (pp. 89-100). Singapore: Springer Nature Singapore.

[13] Vadapalli, S., Abdelhalim, H., Zeeshan, S., & Ahmed, Z. (2022). Artificial intelligence and machine learning approaches using gene expression and variant data for personalized medicine. *Briefings in bioinformatics*, *23*(5), bbac191.

[14] Cong, Y., Shintani, M., Imanari, F., Osada, N., & Endo, T. (2022). A new approach to drug repurposing with two-stage prediction, machine learning, and unsupervised clustering of gene expression. *OMICS: A Journal of Integrative Biology*, *26*(6), 339-347.

[15] Mallick, P. K., Mohapatra, S. K., Chae, G. S., & Mohanty, M. N. (2023). Convergent learning–based model for leukemia classification from gene expression. *Personal and Ubiquitous Computing*, *27*(3), 1103-1110.

[16] GSE12345 Dataset, https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE12345