Implementation of a Genetic Algorithm for gene clustering

Alberto Catalano, Alessandro Delle Site, Sofia Pietrini

Abstract

Clustering gene expression data is a critical technique in transcriptomics for identifying functionally related genes, yet deterministic methods like K-means often struggle with high dimensionality and susceptibility to local optima. This study presents a Genetic Algorithm (GA) that encodes cluster assignments at the gene level and optimizes the Total Within-Cluster Variation (TWCV) using Pearson correlation distance, a metric well suited for capturing co-expression patterns.

The algorithm integrates geometric crossover, distance-based probabilistic mutation, and dynamic parameter adaptation to balance exploration and exploitation throughout evolution. It was validated on four TCGA cancer datasets with known molecular subtypes and benchmarked against standard K-means using quantitative metrics and signature-gene recovery. Experimental results show that the GA successfully minimizes TWCV and autonomously identifies the optimal number of clusters using the elbow method. Furthermore, comparative analysis reveals that the GA generally outperforms K-means in recovering biological ground truths.

Overall, while computationally more demanding, this bio-inspired approach can provide a more robust framework for uncovering complex transcriptomic patterns.

I. Introduction

Clustering is a fundamental technique in transcriptomics used to group genes with similar mRNA expression profiles, with the aim of identifying functionally related genes [1]. Deterministic methods such as K-means are widely used for clustering, but they often struggle with the high dimensionality of gene expression data and are highly sensitive to initial conditions, frequently converging to local optima rather than the best possible solution [2].

Genetic Algorithms (GAs) have been explored as a robust alternative due to their ability to navigate complex search spaces, making them less prone to becoming trapped in local optima. However, many existing studies we have found in this domain lack open-source implementations or detailed information on the datasets used.

In this report, we present a Genetic Algorithm designed for clustering gene expression data, in which we have used strategies from different literature studies, and hyperparameters that have been refined through experimentation. The algorithm was validated against gene expression datasets where the true number of clusters is known to benchmark its performance, and compared to a standard k-means approach.

II. Dataset

Four TCGA gene expression datasets representing diverse cancer types (BRCA, CRC, GBM, LUAD) and four established molecular subtypes (PAM50 [3], CMS [4], Verhaak [5] and Wilkerson [6]) were selected to evaluate the performance of our algorithm. Gene expression data are typically structured as a matrix, with rows representing individual genes and columns corresponding to patient samples or experimental conditions.

From each original dataset, a subset of 3000 genes yielding the highest variance was retained to focus on the most informative features. A careful verification was performed to ensure that all relevant gene expression signatures were not excluded.

A summary of the four datasets, including their associated tumor type, the name and number of signature genes, and the expected number of clusters, is provided in *Table 1*.

TABLE I
Characteristics of the four datasets, including tumor type, signature gene set name and size, and the expected number of clusters

Dataset Name	Tumor Type/Tissue	Number of Signatures	Expected Number of Clusters
BRCA	Breast	PAM50 – 50	k = 5
CRC	Colorectal	CMS – 40	k = 4
GBM	Glioblastoma	Verhaak – 15	k = 3
LUAD	Lung	Wilkerson – 60	k = 3

III. Methodologies

This section summarizes the methods used in this study and outlines the steps taken to ensure reproducibility. We describe the genetic algorithm representation and objective, the selected evaluation metrics, the fitness and constraint-handling procedures, and the evolutionary operators with their dynamic parameter adaptation. We also detail the elbow-method implementation, data preprocessing steps and the visualization procedures.

A. GA representation and objective

GAs are heuristic search strategies inspired by the principles of natural evolution, which use a population of candidate

[1]Full implementation available on GitHub

solutions that iteratively evolve through selection, crossover, and mutation to approximate a global optimum [7].
In this implementation, each candidate solution (chromosome) is represented as an integer array of length N, where N corresponds to the total number of biological genes in the dataset. Each index in the array represents a specific biological gene, and the value $\{0, \ldots, k-1\}$ represents its cluster assignment.
The optimization objective is to minimize the Total Within-Cluster Variation (TWCV) using Pearson Correlation Distance as the similarity metric. The cost function is defined as:

$$TWCV = \sum_{k=1}^{K} \sum_{x \in C_k} \big(1 - \text{similarity}(x, \mu_k)\big)$$

Where μ_k is the unit-normalized centroid of cluster k computed as the average of all points divided by its length, and the similarity term, calculated via the dot product between a data point vector (x) and a centroid vector (μ_k), corresponds to the Pearson correlation coefficient.

B. Metric Selection

We selected Pearson Correlation over the standard Euclidean distance as it is considered a more suitable metric for gene expression analysis [8]. In fact, while Euclidean distance focuses on the magnitude of the vectors, Pearson correlation measures the linear dependence between them, and, in the context of gene expression, we are strictly interested in genes that behave similarly (increasing or decreasing together) regardless of their absolute expression levels values. Pearson correlation ensures that genes that are co-regulated are correctly grouped together, even if their magnitudes differ. To compute this metric efficiently within the GA, we utilize the property that the Cosine Similarity of centered data is mathematically equivalent to Pearson correlation. As detailed in the Data Preprocessing paragraph in *Section III-F*, by standardizing and normalizing the data inputs, we can compute the Pearson correlation using a simple vector dot product.

C. Fitness Evaluation and Constraint Handling

Since GAs typically maximize a fitness score, the minimization problem (lowering TWCV) was inverted. The fitness function $f(i)$ for an individual i is dynamically calculated relative to the population's maximum error for that current generation, as proposed in [9]:

$$\text{Fitness} = 1.5 * \text{TWCV}_{\text{max}} - \text{TWCV}_{\text{individual}}$$

where $TWCV_{max}$ represents the maximum error value observed in the current population. The system also implements a strict legality check: solutions containing illegal strings (empty clusters) are heavily penalized. Their fitness score is computed as the product of the legality ratio (the fraction of non-empty clusters) and the lowest fitness value found among the valid solutions (the legal strings) in the population [9]. This creates strong evolutionary pressure to maintain k active clusters.

D. Evolutionary Operators and Dynamic Parameter Adaption

To ensure accurate clustering within the context of gene expression data, we employed geometric operators that act directly on the phenotypic centroids rather than on the raw genotypic labels.

- **Selection:** The algorithm uses *Roulette Wheel Selection*, where the probability of choosing a parent is proportional to its fitness. Additionally, *Elitism* is enforced, ensuring the single best individual is copied unchanged to the next generation to prevent regression.
- **Geometric Crossover**: The centroids of parent A are aligned with the closest centroids of parent B to ensure similar clusters are combined [10]. New "child" centroids are created by linearly interpolating between the matched parents using a mixing parameter and adding Gaussian noise. The dataset is then re-assigned to the nearest new centroid to generate the child chromosome.
- **Mutation:** Rather than randomly flipping cluster assignments, which causes chaotic search patterns, the algorithm utilizes *Distance-Based Probabilistic Mutation* [9]. When a point is selected for mutation, the algorithm calculates its distance to all current centroids. A probability distribution is constructed such that the point is significantly more likely to be reassigned to a geometrically closer cluster. In this way, the mutation nudges points toward clusters that are geometrically closer, avoiding chaotic jumps to far-away clusters.

To balance exploration (searching the global space) and exploitation (refining the best solution), the hyperparameters decay linearly over the course of the simulation:

- **Mutation Rate:** Decays from 0.01 to 0.001
- **Crossover Rate:** Decays from 0.85 to 0.65
- **Noise Level:** Gaussian noise added during crossover decreases to zero, allowing for precise convergence in final generations

Due to this dynamic adaptation, in the early stages, high mutation and crossover rates facilitate the combination of diverse solutions and prevent premature convergence, while reduced rates in later stages allow the algorithm to fine-tune the cluster boundaries and stabilize the optimal structure found.

E. Elbow method implementation:

To automatically determine the optimal number of clusters (k), we implemented an Elbow Method utilizing a geometric "Knee Point" detection algorithm: the GA is executed iteratively for a range of candidate k values, for each of whom, the best TWCV is recorded. Then, rather than relying on visual inspection, the optimal k is calculated mathematically as the point on the TWCV curve that maximizes the perpendicular distance to the line connecting the first and last points of the curve, as proposed in [11].
This identifies the point where adding more clusters no longer yields a significant reduction in error.

F. Data Pre-processing

The raw gene expression data has to be processed through a specific two-step transformation pipeline, which is mathematically mandatory for our specific code to work correctly, to ensure the clustering reflects biologically meaningful co-regulation patterns rather than simple magnitude differences.

- **Row-wise Standardization (Z-Score):**
  The raw expression matrix X was transposed and processed using *StandardScaler* to normalize each gene independently. This centers the expression profile of every gene such that it has a mean of 0 and a standard deviation of 1. Centering the data is mathematically crucial because the cosine similarity of centered vectors is equivalent to their Pearson correlation, a standard metric in gene co-expression analysis.
- **L2 Normalization:**
  Then, each gene vector was normalized to unit length ($\|\mathbf{x}\|_2 = 1$). This step ensures that the magnitude of expression (how highly expressed a gene is overall) does not dominate the clustering; only the shape of the expression profile across samples matters.

G. Plot visualization

Since the gene expression data is high-dimensional (where D equals the number of samples), direct visualization of clusters is impossible. We employed Principal Component Analysis (PCA) to project the data into a valid 2D space.

- **Dimensionality Reduction:** The code computes the first two Principal Components of the dataset, capturing the directions of maximum variance.
- **Final Outputs:** The final outputs show two side-by-side figures:
  - **Convergence Plot:** Tracks the best TWCV score over generations to verify that the Genetic Algorithm successfully minimized the error.
  - **Cluster Scatter Plot:** Displays the genes in the reduced PCA space, colored by their cluster assignment. This allows for a visual sanity check of the separation between groups.

IV. Results

The proposed Genetic Algorithm was evaluated on the four datasets presented in *Section II*. In this report, we focus on presenting the results obtained from the Breast invasive carcinoma (TCGA - BRCA) dataset, consisting of five known molecular subtypes: Basal, Her2-enriched, Luminal, Normal-like/Other, and Proliferation. Complete plots and clustering results for the remaining datasets, which exhibited similar performance, are available in the GitHub repository.
The convergence and cluster scatter plot are reported in *Figure 1*.

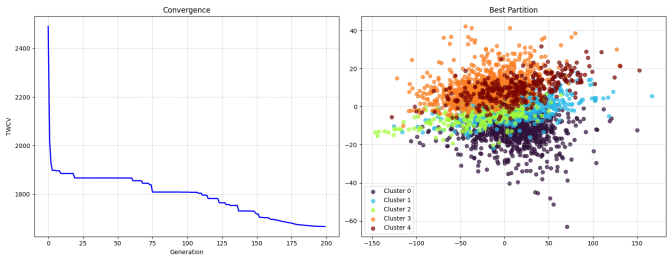

Fig. 1. *The left panel illustrates the minimization of Total Within-Cluster Variance (TWCV) across generations. The right panel corresponds to the optimal solution found, showing the spatial distribution of the five identified clusters.*

On the left panel, the convergence profile tracks the minimization of the Total Within-Cluster Variation across 200 generations. The curve exhibits a characteristic behavior often seen in evolutionary strategies: a sharp initial descent (first 10 generations) reflects the rapid elimination of poor solutions, while subsequent step-like drops indicate the algorithm's capability to escape local optima through the use of dynamic mutation and crossover operations. The stabilization of the curve in the last generations confirms that the population has converged to a steady state.
The plot on the right displays the final five-cluster solution projected into a 2D space using PCA. While the clusters appear to overlap spatially, most notably the *orange* (cluster 3) and *red* (cluster 4) groups, this is an artifact of the projection method rather than a defect in the clustering itself. In fact, the GA grouped genes based on Pearson correlation, which prioritizes the shape of the expression patterns (co-regulation). In contrast, the PCA visualization projects data based on global variance, which is inherently tied to Euclidean magnitude. Consequently, the cluster boundaries may appear less distinct in this 2D projection, but it is not an indication of poor clustering.

It is important to note that while we fixed $k = 5$ for this experiment to align with the five known PAM50 subtypes, the Elbow method actually indicated an optimal trade-off at $k = 4$. The reason for this discrepancy becomes evident when analyzing the comparison matrix.
As shown in *Figure 2 (left)*, the GA achieved perfect classification (100% match) for four of the five subtypes: Basal, Her2-enriched, Luminal, and Proliferation. However, the "Normal-like/Other" samples were not identified as a distinct cluster, instead, they were dispersed across the other four groups. This suggests that the "Normal-like/Other" expression signature is less distinct or possesses lower internal cohesion compared to the other tumor types, leading the algorithm to mathematically prefer a 4-cluster solution.
Consequently, the choice of $k = 5$ forced the algorithm to search for a partition that did not naturally emerge from the data structure. Despite this limitation regarding the "Normal-like/Other" class, the GA demonstrated superior performance compared to the standard K-means approach *(Figure 2, right)*. The GA successfully captured the biological heterogeneity, achieving a perfect one-to-one mapping for four distinct

molecular subtypes (Basal, HER2-enriched, Luminal, and Proliferation), assigning 100% of their signature genes to unique, non-overlapping clusters. On the contrary, K-means failed to distinguish between the Basal and Proliferation signatures; it collapsed these two biologically distinct groups into a single cluster (C0), effectively merging 100% of Basal genes and 95% of Proliferation genes, losing critical biological structure that the GA correctly preserved.

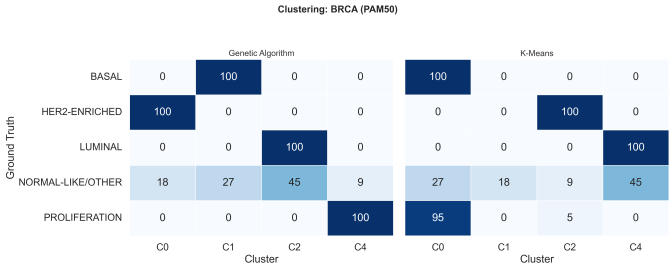

Fig. 2. *Comparison of GA and K-means clustering of PAM50 signature genes. Numbers indicate the percentage of signature genes correctly clustered, while colors represent clustering quality. A value of 100% indicates that all signature genes were assigned to the correct cluster.*

In terms of biological fidelity, the GA emerges as the clear winner. In *Figure 3*, beyond biological relevance, we compared the algorithms using three quantitative metrics: Silhouette score, final fitness value, and runtime. The GA achieved consistently superior cluster quality, recording higher Silhouette Scores in all four cancer types compared to K-means, indicating better-defined and more cohesive clusters. Similarly, the algorithm consistently reached lower TWCV values, confirming its ability to find better optima. However, the GA's superior biological fidelity comes at a significant computational cost: it is orders of magnitude slower (approximately 10^3 seconds vs < 10 seconds), characterizing it as a high-precision solution suitable for offline analysis where accuracy is preferred over speed.

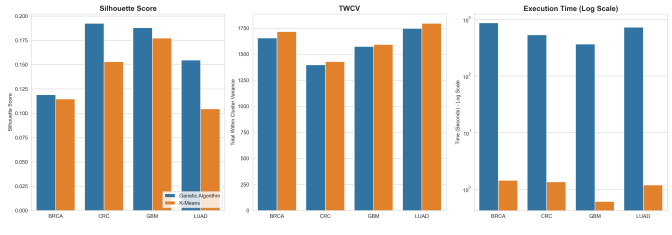

Fig. 3. *Comparison of GA (blue) and K-means (orange) across all datasets. The evaluated metrics include the silhouette score, fitness function (TWCV), and runtime.*

V. Conclusion

In this report, we developed and validated a Genetic Algorithm specifically designed to address the challenges of clustering high-dimensional gene expression data.

Due to the scarcity of open-source implementations and the lack of detailed dataset documentation in existing literature, our initial objective of comparing multiple bio-inspired clustering algorithms proved infeasible. Therefore, we focused exclusively on the development of a GA, synthesizing effective strategies and hyperparameter configurations.

By integrating geometric operators, a distance-based mutation strategy, and an inverted fitness function minimizing Total Within-Cluster Variation based on Pearson correlation, the proposed method effectively navigated the complex search space of transcriptomic profiles.

Our experimental results on the TCGA-BRCA dataset highlight the algorithm's distinct advantage over standard K-means clustering. While K-means failed to resolve key biological signatures, such as the distinction between Basal and Proliferation signatures, the GA achieved perfect classification purity for four out of five PAM50 molecular subtypes. The only limitation observed was the inability to separate the "Normal-like/Other" subtype, a result consistent with the mathematical optimal $k = 4$ suggested by the Elbow method.

Quantitatively, the GA consistently achieved lower TWCV scores, proving its effectiveness in minimizing correlation-based error. However, this precision comes at a significant computational expense, with runtimes orders of magnitude higher than K-means. These performance trends were consistent across the remaining three datasets.

VI. Contributions

We equally contributed to the methodology design, literature review, dataset acquisition and implementation.

References

[1] X. Yu, G. Yu, and J. Wang, "Clustering cancer gene expression data by projective clustering ensemble," *PLoS One*, vol. 12, no. 2, p. e0171429, 2017.

[2] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386, 2004.

[3] A. Thennavan, F. Beca, Y. Xia, S. Recio, K. Allison, L. Collins, G. Tse, Y. Chen, S. Schnitt, K. Hoadley, A. Beck, and C. Perou, "Molecular analysis of tcga breast cancer histologic types," *Cell Genomics*, vol. 1, no. 3, p. 100067, 2021.

[4] S. Buechler, M. Stephens, A. Hummon, and et al., "Colotype: a forty gene signature for consensus molecular subtyping of colorectal cancer tumors using whole-genome assay or targeted rna-sequencing," *Scientific Reports*, vol. 10, p. 12123, 2020.

[5] W. Teo, K. Sekar, P. Seshachalam, and et al., "Relevance of a tcga-derived glioblastoma subtype gene-classifier among patient populations," *Scientific Reports*, vol. 9, p. 7442, 2019.

[6] T. C. G. A. R. Network, "Comprehensive molecular profiling of lung adenocarcinoma," *Nature*, vol. 511, pp. 543–550, 2014.

[7] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm- a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 380–384.

[8] P. Jaskowiak, R. Campello, and I. Costa, "On the selection of appropriate distances for gene expression data clustering," *BMC Bioinformatics*, vol. 15, no. Suppl 2, p. S2, 2014.

[9] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, "Incremental genetic k-means algorithm and its application in gene expression data analysis," *BMC Bioinformatics*, vol. 5, p. 172, 2004.

[10] P. Kudova, "Clustering genetic algorithm," in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, Regensburg, Germany, 2007, pp. 138–142.

[11] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, Minneapolis, MN, USA, 2011, pp. 166–171.