University of Baghdad
College of Engineering
JOURNAL OF ENGINEERING

Journal of Engineering
journal homepage: www.jcoeng.edu.iq
Volume 31 Number 8 August 2025

Regularized K-Means Clustering via Fully Corrective Frank-Wolfe Optimization

Ahmed Yacoub Yousif *, [1, 2], **Basad Al-Sarray** [3]

[1] Department of Mathematics, College of Science, University of Baghdad, Baghdad, Iraq
[2] Department of Applied Sciences, University of Technology, Baghdad, Iraq
[3] Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

ABSTRACT

Clustering high-dimensional data remains challenging because traditional *k*-means is sensitive to noise, outliers, and high dimensionality, often leading to unstable performance. The research presents a robust clustering system which combines the Fully Corrective Frank-Wolfe (FCFW) algorithm with *k*-means objective that uses Frobenius norm regularization. The addition of Frobenius norm regularization in the model produces more stable clusters while preventing overfitting and promoting cluster compactness. The proposed method uses probabilistic cluster assignments to enable each data point to join multiple clusters at different membership levels, thus supporting clusters with overlapping boundaries. The Kruskal-Wallis test functions as a feature selection method to identify crucial genes, which then guide the clustering operation toward important features in high-dimensional datasets. The FCFW-regularized *k*-means outperforms traditional *k*-means in all experiments performed on synthetic and real gene expression datasets. On a breast cancer gene expression dataset (GSE10797), it achieved an Accuracy of 89.39%, compared to 58% for traditional *k*-means. Moreover, it surpassed a recent deep subspace clustering method (scPEDSSC) in Adjusted Rand Index by 8.3% on the Goolam single-cell dataset (0.968 vs. 0.885) and 7.2% on the Deng dataset (0.801 vs. 0.729). Overall, the proposed approach attained the highest ARI and Normalized Mutual Information (NMI) scores across five benchmark datasets. These results confirm that the FCFW-regularized *k*-means yields more accurate and stable clustering results, demonstrating robust performance on high-dimensional data.

Keywords: Clustering, Regularized k-means, Fully corrective Frank-Wolfe.

1. INTRODUCTION

Clustering is an unsupervised learning technique used to discover structure in datasets by grouping similar data points **(Raeisi and Sesay, 2022)**. Among clustering algorithms, the *k*-

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2025.08.11

Article received: 27/02/2025
Article revised: 22/05/2025
Article accepted: 11/06/2025
Article published: 01/08/2025

means method is particularly popular due to its simplicity and efficiency. However, k-means is notoriously sensitive to noise, outliers, and the curse of dimensionality. In high-dimensional data, such as gene expression profiles, every point is forced into a cluster even if many features are irrelevant, often leading to unstable or spurious results **(Gao et al., 2023; Yousif and Sarray, 2024)**. To address these issues, recent work has incorporated regularization into k-means to improve robustness. For example, l_1-norm penalties induce sparsity in cluster centers **(Raymaekers and Zamar, 2022)** adaptive feature-weighting schemes improve multi-view clustering, and entropy-based penalties reduce sensitivity to outliers **(Wu and Wu, 2020)**. The development of fuzzy c-means based on morphological reconstruction and membership filtering **(Lei et al., 2018)**. A data partitioning method that includes an object-weighting step to assign higher weights to outliers and objects that cause cluster overlap, described by **(Gondeau et al., 2019)**. **(Yang et al., 2024)** used the anchor graph regularization constrained k-means, which effectively learn the membership matrix of data points and the membership matrix of anchors. **(Jiang et al., 2025)** proposed an entropy-regularized k-means clustering algorithm and added a weight value to the optimization function to ignore out-of-bounds data. Several other clustering methods have been developed behind the k-means clustering algorithms. **(Shiltagh and Hussein, 2015)** proposed a new data aggregation technique for wireless sensor networks (WSNs) based on a modified Voronoi fuzzy clustering algorithm (VFCA) to improve network lifetime and reduce energy consumption. **(Mahdi and Mahmood, 2014)** present an enhanced fuzzy c-means clustering algorithm that incorporates spatial information for MRI brain image segmentation, demonstrating improved noise resistance and region homogeneity. The DBScan algorithm receives dynamic parameter optimization to improve cluster quality and noise resistance according to **(Ghathwan and Mohammed, 2022)**. Spectral clustering with an affinity matrix that includes various constraints has been applied in some applications, such as wireless data processing, to improve clustering accuracy **(Blanza, 2021)**. Spatial clustering techniques have been effectively utilized in the analysis of gene expression data from high-dimensional biomedical datasets, and hybrid optimization methods have been employed to enhance clustering performance **(Salman and Hussain, 2023)**.

Recent studies in engineering and environmental sciences have successfully implemented clustering techniques. **(AL-Kordy and Khudair, 2021)** used cluster analysis to evaluate treatment efficiency trends by classifying wastewater parameters in their study of effluent quality assessment for sewage treatment plants. **(Ahmed and Al-Haleem, 2024)** utilizes well logs and core data from cored wells to predict permeability for uncored wells and intervals, uses an approach integrating rock typing by cluster analysis techniques. This reflects the importance of clustering techniques on applications in scientific and engineering research.

The fully corrective Frank-Wolfe (FCFW) algorithm **(Lacoste-Julien and Jaggi, 2015)** adds an active-set re-optimization step to the classical Frank-Wolfe method, accelerating convergence and improving solution quality. The classical Frank-Wolfe (FW) algorithm **(Canon and Cullum, 1968)** is premised on being able to easily solve (at each iteration) linear optimization problems over the feasible region of interest. applied to the dual structural support vector machine (SVM) objective **(Lacoste-Julien et al., 2013)**. The FCFW updates all weights are reoptimized at each iteration, further enhancing convergence **(Lacoste-Julien and Jaggi, 2015)**. Two new variants of the FW algorithms for stochastic finite-sum minimization have the best convergence of existing stochastic FW approaches for both convex and non-convex objective are introduced by **(Beznosikov et al., 2023)**.

Despite these advances, several challenges remain in clustering high-dimensional data. First, it is a variant of the *K*-means algorithm that is less sensitive to noise and outliers because it uses medoids as cluster centers instead of means that are easily influenced by extreme values **(Sun et al., 2012; Jamail and Moussa, 2020)** and uses binary cluster assignments, which cannot capture overlapping data **(Liu et al., 2023)**. A common limitation of most existing clustering approaches is to assume that genes are separated into disjoint clusters. As genes often have multiple functions and thus can belong to more than one functional cluster, the disjoint clustering results can be unsatisfactory. In addition, due to the small sample sizes of genetic profiling studies and other factors, there may not be sufficient evidence to confirm the specific functions of some genes and cluster them definitively into disjoint clusters **(Teran Hidalgo et al., 2018; Saha et al., 2023)**.

To address these gaps, this work proposes a regularized variant of the k-means clustering model tailored for high-dimensional data. The k-means objective is reformulated with a convex Frobenius-norm penalty on the cluster centroids to control model complexity and prevent overfitting to noisy features. The resulting optimization problem, though convex in the continuous assignment space, is large-scale; hence, a fully corrective Frank–Wolfe algorithm (a gradient-based iterative method) is applied to solve it efficiently. Additionally, a feature selection strategy (using a Kruskal–Wallis statistical filter) is incorporated to identify the most discriminative feature subsets for clustering. This approach effectively reduces noise influence and improves interpretability by highlighting which variables drive the cluster distinctions.

The main contributions of this work are summarized as follows:

1. Introduces the use of the FCFW algorithm to optimize a regularized k-means clustering objective, accelerating convergence, and improving solution quality.
2. Employs a Kruskal-Wallis test for statistical feature selection, seamlessly integrating gene selection into the clustering process to handle the high dimensionality of gene expression datasets.
3. Demonstrates superior clustering performance on five benchmark gene expression datasets.

The remainder of the paper is organized as follows. Section 2 defines the regularized k-means clustering problem formulation. Section 3 describes the FW optimization approach, including the fully corrective variant used in our algorithm. Section 4 presents the feature selection procedure based on the Kruskal-Wallis test. Section 5 details the datasets and experimental setup. Section 6 reports the experimental results and compares the proposed method with other clustering approaches. Section 7 discusses the findings and their implications. Finally, Section 8 concludes the paper and outlines directions for future research.

2. MATHEMATICAL PROBLEM FORMULATION

One of the most classical centroid-based clustering algorithms is the k-means clustering **(Li et al., 2021)**, which assigns each point x_i to the corresponding cluster a C_{j} so that every point in each cluster is at a minimal squared Euclidean distance from its center. According to that, the j_{th} cluster of x_i is located at its cluster center, represented by a c_{j} vector of d elements, where k represents the total number of clusters in the dataset.

$$d(x_i, C_j) = \min_{1 \le j \le k} \|x_i - C_j\|^2 \tag{1}$$

The k-means clustering problem aims to minimize the sum of distances between the data points and their cluster centers. K-means is an NP-hard optimization problem in general **(Adams, 2018)** due to the combinatorial nature of assignments. The function in Eq. (1) is non-convex because of the discrete assignment variables $c(i)$. Even for $k = 2$ clusters, finding the global minimum is NP-hard. The classical k-means objective can be written as:

$$\min F(c_1, c_2, \dots, c_k) = \sum_{i=1}^{n} \min_{1 \le j \le k} \|x_i - c_j\|^2 \quad (2)$$

In practice, the k-means is used to find a local minimum. The algorithm alternates between assigning each point to the nearest centroid and recomputing each centroid as the mean of its assigned points. This coordinate descent approach is guaranteed to decrease the objective at each step, but it can converge to a suboptimal partition (a local minimum of Eq. (1)). As discussed, poor initial centroid placement can lead to bad local optima. Moreover, the classic k-means objective lacks any regularization, so it will always reduce the objective by splitting outliers into their own cluster or by using noisy features if those reduce within-cluster variance, even if such clusters or features are not meaningful **(Oyelade et al., 2016; Zhang et al., 2020)**. To address these problems, regularized k-means it includes a sparsity-inducing regularization penalty on the cluster centers to increase stability and robustness.
Let $z_1, z_2, \dots, z_n \in \mathbb{R}^p$ be n data points in a p-dimensional feature space. k-means clustering seeks to partition these n points into k clusters while regularization of k-means clustering, it is useful to introduce soft (probabilistic) cluster assignments. Instead of fixed integer labels $c(i)$, each data point is assigned a distribution over k clusters.
The objective of regularized k-means minimizing the cost function:

$$\min_{C,X} \frac{1}{n} \sum_{i=1}^{n} \|z_i - \sum_{j=1}^{k} X_{ij} C_j\|^2 + \lambda \| C \|_F^2 \quad (3)$$

Here, $C = \{C_1, C_2, \dots, C_k\} \subset \mathbb{R}^{k \times d}$, with $C_j \in \mathbb{R}^d$, are the k-cluster centers, and $X = \{x_1, x_2, \dots, x_n\} \in \{1,2, \dots, k\}^n$ is the cluster assignment matrix.
The optimization is subject to the constraints that each data point is assigned probabilistically to clusters:

$$\sum_{j=1}^{k} X_{ij} = 1,\ X_{ij} \ge 0,\ \forall i \in \{1, \dots, n\}, j \in \{1, \dots, k\}. \quad (4)$$

The term, $\frac{1}{n} \sum_{i=1}^{n} \|z_i - \sum_{j=1}^{k} X_{ij} C_j\|^2$, captures the squared Euclidean distance from each point z_i to the assigned cluster center. $X_{ij} \in [0,1]$ denote the membership of point i in cluster j, with the constraint $\sum_{j=1}^{k} X_{ij} = 1$ for each i (so each point's membership weights form a probability vector over clusters). In matrix form, X is an $n \times k$ assignment matrix where row i (denoted $X_{i,:}$) lies in the $(k-1)$ simplex. In the special case of classical k-means, X_{ij} are binary indicators: $X_{ij} = 1$ if $c(i) = j$ and 0 otherwise. The regularization term is defined as $\lambda \| C \|_F^2$ where $\| C \|_F^2$ is the Frobenius norm of the cluster center matrix C to prevent large values in C. This favors more stable and smoother cluster centers and prevents overfitting of the cluster centers to have excessively large magnitudes. The regularization parameter $\lambda \ge 0$ is to balance the accuracy of clustering and regularization.
The notion $Cx_i = \sum_{j=1}^{k} X_{ij} C_j$ is the cluster center assigned to z_i in a probabilistic approach, to avoid evaluating a large number of possible assignments.

Regularized k-means clustering refines the standard k-means algorithm by introducing soft cluster assignments and regularization on cluster centers. This helps avoid overfitting and improves the robustness of clustering solutions.

3. THE FRANK-WOLFE ALGORITHM AND ITS FULLY CORRECTIVE VARIANT

The FW algorithm **(Frank and Wolfe, 1956; Sarray et al., 2017)**, also known as the conditional gradient method, is a first-order optimization technique that provides a projection-free approach to solving constrained convex optimization problems. Unlike other methods that produce iterates requiring projection back onto the constraint region, the FW algorithm generates feasible solutions by leveraging the compact and convex nature of the constraint set. For more details, see **(Lacoste-Julien and Jaggi, 2015; Wirth et al., 2024)**. Given a smooth convex objective function $f(x)$ and a convex feasible region $C \subseteq \mathbb{R}^n$, the optimization problem addressed by the FW algorithm is formulated as follows:

$$\min_{x \in C} f(x) \tag{5}$$

The Frank-Wolfe algorithm is introduced (also known as the conditional gradient method) and then describes the Fully Corrective Frank-Wolfe variant. It will be shown how FCFW can be applied to solve for the optimal assignment matrix X in a way that is efficient and yields additional theoretical guarantees (like sparse solutions and sometimes faster convergence). The Frank-Wolfe algorithm is a projection-free first-order method for solving constrained convex optimization problems. Given a convex, differentiable function $f(x)$ to minimize over a compact convex set C, a Frank-Wolfe iteration avoids direct gradient descent (which would require projecting onto C) and instead linearizes the objective at the current point and moves toward the minimizer of that linear approximation **(Cherfaoui et al., 2018)**. This method is summarized in the following Algorithm:

Algorithm 1. Frank-Wolfe Algorithm

Input: Initial point $x_0 \in C$. and step-size strategy $\gamma_t \in [0,1]$

Output: Approximate solution $x^* = x_k$ for minimizing $f(x)$ within C.

Process: Begin

Steps 1: Initialize $x_0 \in C$, set iteration counter $k = 0$.

Steps 2: Repeat until convergence or maximum iterations T are reached:

- Step 2.1: Compute the gradient $\nabla f(x_k)$.
- Step 2.2: Solve the linearized subproblem:
$$v_t = \arg\min_{v \in C} \langle \nabla f(x_t), v \rangle$$
- Step 2.3: Compute the step size γ_t use $\gamma_k = \frac{2}{k+2}$.
- Step 2.4: Update the solution:
$$x_{t+1} = (1 - \gamma_t)x_t + \gamma_t v_t$$
- Step 2.5: Check for convergence (e.g., $\|x_{k+1} - x_k\| < \epsilon$).

End

These steps are repeated until convergence. One key feature of Frank-Wolfe is that the iterates remain in C without any projection.

Frank-Wolfe often zig-zags when the optimum lies at a boundary of C. This zig-zagging is because the algorithm always moves along edges of the feasible polytope **(Holloway, 1974)**. The fully corrective step ensures that all previously chosen directions are used in the best possible way at each iteration, not just the last added one. In contrast, standard FW would only adjust the combination gradually via line searches over many iterations. FCFW usually reduces the objective more per iteration, at the cost of solving the small subproblem. In many cases (like quadratic objectives), the corrective step is computationally cheap because it's just solving linear equations. Empirically, FCFW tends to need far fewer iterations to reach high accuracy and avoids the zig-zagging behavior by immediately readjusting weights on previous atoms. This algorithm is described as:

Algorithm 2. Fully Corrective Frank-Wolfe Algorithm

Input: Initial point $x_0 \in \mathcal{C}$.

Output: $x^* = x_t$, the approximate solution

Process: Begin

Steps 1: Initialize starting point $x_0 \in \mathcal{C}$,active set $S_0 = \{x_0\}$ and set $t = 0$.

Steps 2: Repeat until convergence or maximum iterations T :

- Step 2.1: Compute the gradient of the objective function:
$$\nabla f(x_t)$$
- Step 2.2: Solve the linearized subproblem to find the descent direction:
$$v_t = \arg\min_{v \in \mathcal{C}} \langle \nabla f(x_t), v \rangle$$
- Step 2.3: Add the new vertex v_t to the active set:
$$S_{t+1} = S_t \cup \{v_t\}$$
- Step 2.4: Re-optimize over the convex hull of the active set to compute the next iterate:
$$x_{t+1} = \arg\min_{x \in \text{conv}\,(S_{t+1})} f(x)$$
where conv (S_{t+1}) is the convex hull of the active set S_{t+1}.
- Step 2.5: The objective function value $f(x_{t+1})$ improves by less than a threshold by convergence, such as:
$$\|x_{t+1} - x_t\| < \epsilon.$$

End

4. FULLY CORRECTIVE FRANK-WOLFE ALGORITHM FOR REGULARIZED *K*-MEANS CLUSTERING

The Fully Corrective Frank-Wolfe (FCFW) algorithm aims to iteratively refine the active set to find the optimal solution. Unlike the standard Frank-Wolfe algorithm, FCFW performs a full correction over the active set at each iteration, which increases the computational cost per iteration but significantly accelerates convergence overall.

Moreover, FCFW effectively handles sparse problems by removing unnecessary points from the solution vector, making it suitable for high-dimensional optimization where the solution is expected to lie on a low-dimensional face of the feasible region. In the context of Regularized k-means clustering, FCFW combines gradient-based optimization with active set corrections, enhancing both clustering accuracy and convergence speed.

The general steps of the FCFW algorithm for Regularized k-means are as follows:

Algorithm 3. Fully Corrective Frank-Wolfe Algorithm for Regularized K-Means

Input: Z matrix, k number of clusters, λ regularization parameter.
Output: C_{opt} cluster centers and X_{opt} final cluster matrix.

Process: Begin
Steps 1: Initialization centers C_0 using k-means++ **(Daoudi et al., 2021)**.
Steps 2: Repeat for $t = 1,2, \ldots,$ maxIter:

Step 2.1: Assign points to nearest cluster centers:

$$X_t(i,j) = \begin{cases} 1, & \text{if } j = \arg\min_{j'} \|Z(i,:) - C_t(j',:)\|^2 \\ 0, & \text{otherwise} \end{cases}$$

Step 2.2: Compute the gradient with respect to C_t :

$$\nabla f(C_t) = -\frac{2}{n} X_t^{\top}(Z - X_t C_t) + 2\lambda C_t$$

Step 2.3: Fully corrective step by solve the quadratic program:

$$C_{t+1} = \arg\min_{C} \left\{ \frac{1}{n} \|Z - X_t C\|^2 + \lambda \parallel C \parallel_F^2 \right\}$$

Use the precomputed gradient and Hessian.

$$C_{t+1} = \left(\frac{1}{n} X_t^{\top} X_t + \lambda I\right)^{-1} \left(\frac{1}{n} X_t^{\top} Z\right).$$

Step 2.4: Compute the objective function value:

$$\mathrm{O}_t = \frac{1}{n} \|Z - X_t C_{t+1}\|^2 + \lambda \|C_{t+1}\|_F^2$$

Step 2.5: Check convergence:

$|O_t - O_{t-1}| <$ tol, stop the iteration.

Step 2.6: Return Results:

$$C_{\text{opt}} = C_{t+1}, X_{\text{opt}} = X_t.$$

End

5. STATISTICAL ANALYSIS USING THE KRUSKAL-WALLIS TEST

This section briefly reviews the main statistical technique used in the feature selection process for clustering gene expression data. The Kruskal-Wallis **(Grisci et al., 2019; Meléndez Surmay et al., 2024)** test was utilized to identify the most relevant features (genes) for distinguishing between classes in a high-dimensional dataset. This non-parametric statistical test compares the median values across multiple independent groups and is particularly effective when the assumption of normality, required by parametric methods like ANOVA, is unrealistic.

The null hypothesis for the Kruskal-Wallis test states that the distributions of gene expression levels are identical across all groups, (H_0), while the alternative hypothesis asserts that at least one group differs in median gene expression (H_1). Formally, the hypotheses can be written as:

$H_0: \mu_1 = \mu_2 = \cdots = \mu_k$ vs H_1: Not all μ_j are equal.

The test is based on ranking all observations within each group. The rank sum for group g, denoted as S_g, is defined as:

$$S_g = \sum_{i=1}^{m_g} \text{rank}_{i,g} \quad (6)$$

where rank $k_{i,g}$ represents the rank of the i-th observation in group g, and m_g is the size of group g. The average rank for each group is then computed as:

$$\bar{S}_g = \frac{S_g}{m_g},\ g = 1,2,\dots,k \tag{7}$$

The Kruskal-Wallis statistic, H, is calculated as:

$$H = \frac{12}{M(M+1)} \sum_{g=1}^{k} m_g \left(\bar{S}_g - \frac{M+1}{2}\right)^2 \tag{8}$$

where $M = \sum_{g=1}^{k} m_g$ is the total number of observations. At a given significance level α, H_0 is rejected if $H \geq \chi^2_{k-1,\alpha'}$, where $\chi^2_{k-1,\alpha}$ is the critical value of the chi-square distribution with $k-1$ degrees of freedom.

Clustering high-dimensional gene expression data is challenging due to the large number of features (tens of thousands of genes) relative to sample size and the presence of many irrelevant or noisy genes. To address this, the main statistical preprocessing step in our framework is a feature selection procedure that dramatically reduces dimensionality before clustering. In particular, a non-parametric Kruskal–Wallis (KW) test is employed to identify and retain only the most informative genes. The KW test is a rank-based statistical test that compares the expression levels of a gene across multiple groups (e.g., different cell types) to determine if any group differs significantly from the others. It outputs a p-value for each gene, testing the null hypothesis that all samples have come from the same distribution for that gene (i.e., no differential expression across groups). Genes with low p-values are thus informative, indicating significant expression differences among the groups, whereas high p-values suggest non-informative genes that do not vary meaningfully between conditions. In essence, the clustering algorithm only needs to consider a few hundred dimensions instead of tens of thousands, significantly cutting down on distance calculations and model parameters. This kind of feature selection is known to improve efficiency and scalability in high-dimensional learning tasks by removing irrelevant features.

6. SYNTHETIC AND GENE EXPRESSION DATA

This section presents several experimental results. The proposed algorithm is tested on synthetic datasets and real-world high-dimensional gene expression datasets to evaluate its effectiveness across different clustering scenarios. The experiments were conducted on a system with an Intel Core i5 processor and 16 GB RAM using MATLAB 2024a.

6.1 Datasets and Preprocessing

To improve the accuracy of the proposed regularized k-means algorithm, the following steps were applied according to the type of data (see **Fig. 1**). Firstly, Z-score normalization was used to standardize gene expression data, ensuring all features had a zero mean and unit standard deviation. For a given feature x_i, the normalized value x_i' is calculated as:

$$x_i' = \frac{x_i - \mu}{\tau} \tag{9}$$

where μ is the mean and τ is the standard deviation of the feature. This step eliminates scaling biases, enabling fair comparisons during clustering. Secondly, the Kruskal-Wallis test

was utilized to select the top significant features by determining their statistical significance concerning the target classes. In our approach, every gene in each dataset is ranked by its KW test p-value (reflecting its discriminative ability). Only the top-ranking genes-those most likely to differentiate between sample groups-are selected for clustering. For example, in breast cancer, we retained the top 200 genes with the smallest p-values per dataset, significantly reducing the original feature spaces (e.g., 54,676 genes in the GSE42568 dataset and 22,278 in the GSE10797 dataset). In the single-cell RNA-seq datasets, we retained the top 500 genes with the smallest p-values per dataset, a drastic reduction from the original feature spaces (e.g., 40,315 genes in the Goolam dataset and 12,548 in the Deng dataset).
After this feature selection step, the reduced gene expression data (containing only the informative genes) is passed into the proposed FCFW-regularized k-means clustering algorithm. Integrating the Kruskal-Wallis gene filtering with the FCFW optimized k-means forms a two-stage framework: first, dimensionality is curtailed by removing non-informative genes, and second, clustering is performed on the compact, information-rich feature set.
For synthetic datasets with non-linear structures, a Gaussian kernel transformation was employed to improve cluster separability.
The kernel matrix K is computed as **(He and Zheng, 2018)**:

$$K_{ij} = \exp\left(-\frac{\|z_i - z_j\|^2}{2\sigma^2}\right) \tag{10}$$

with σ controlling the kernel width.
Lastly, the performance of the regularized k-means the algorithm was fine-tuned by adjusting the regularization parameter λ, which governs cluster compactness and separation.

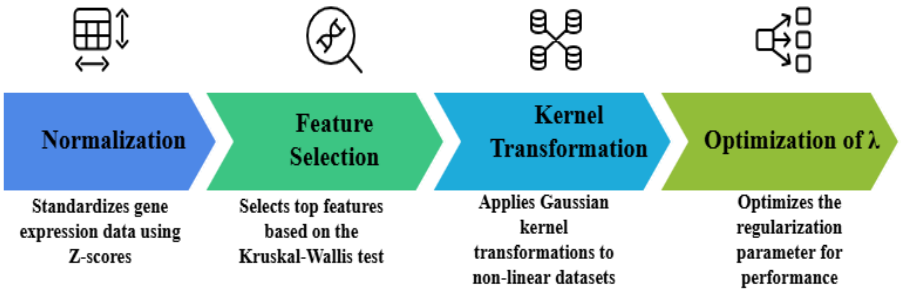
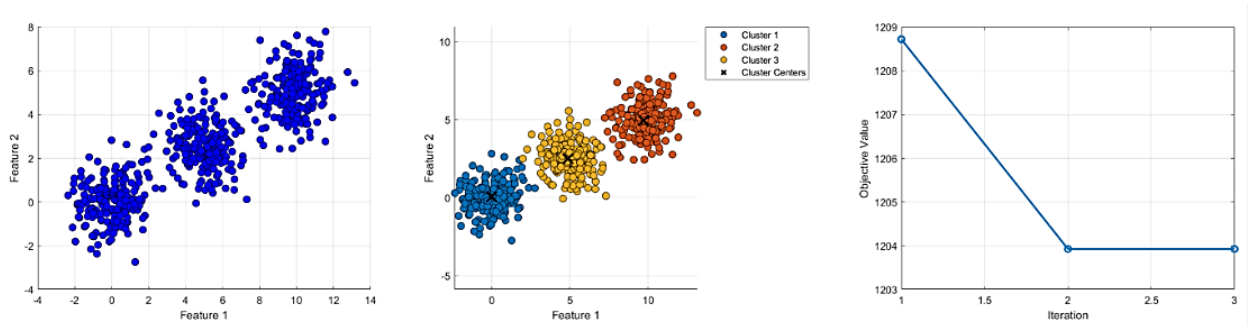


Figure 1. Preprocessing for regularized k-means.

6.2 Synthetic Data Description and Results

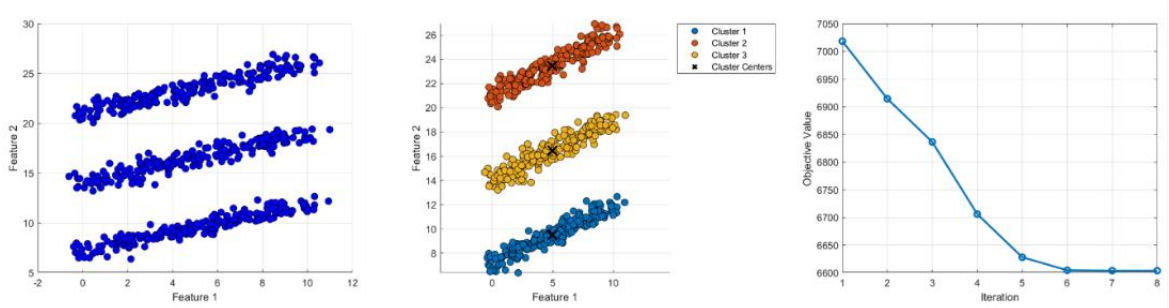
The regularized k-means algorithms' effectiveness was tested using four datasets: compact blobs, linear clusters, with slope, two concentric circles, and two moons. These datasets posed various clustering difficulties, ranging from easily separable clusters to intricate nonlinear structures.
In the Compact Blobs dataset, 600 points were equally distributed among 3 Gaussian clusters, each with a standard deviation of 1. Using a regularization parameter, $\lambda = 0.005$, the algorithm was able to identify the clusters and converge in 3 iterations (see **Fig. 2**).



(a): Data distribution before clustering (b): Clustering results with cluster centers (c): Objective function convergence

Figure 2. Clustering results for the compact blobs dataset.

The Linear Clusters dataset had 600 points arranged in 3 linear clusters, some randomness was introduced through Gaussian noise with a standard deviation of 0.5 with $\lambda = 0.001$, the algorithm worked well to distinguish between the clusters and reached convergence in 8 iterations (see **Fig. 3**).

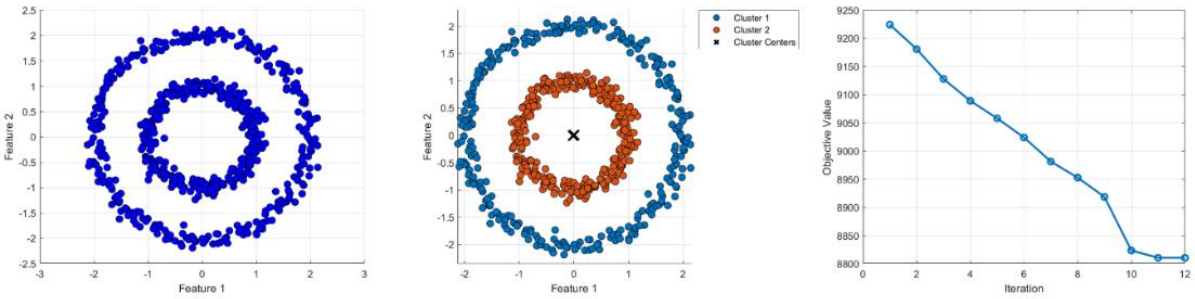


(a): Data distribution before clustering (b): Clustering results with cluster centers (c): Objective function convergence

Figure 3. Clustering results for linear clusters with slope.

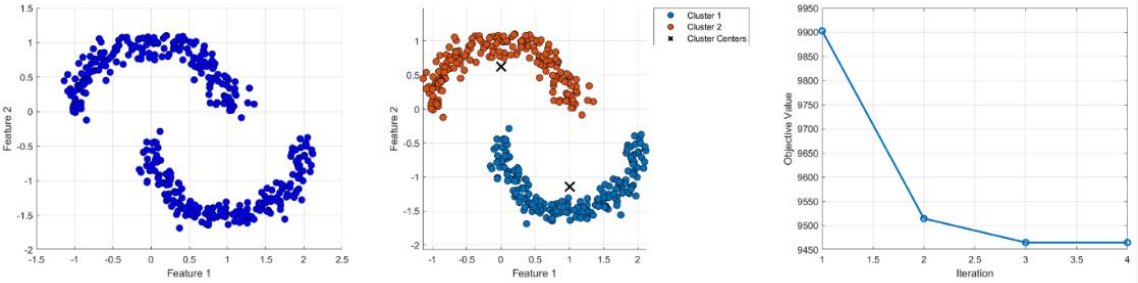
The two concentric circles dataset uses 400 points randomly distributed in two circles, with additional Gaussian noise of level 0.09. Since the data was nonlinear, a Gaussian kernel transformation with $\sigma = 0.3$ was used to transform the data into a higher-dimensional space where the clusters could be separated by a linear classifier. With $\lambda = 0.3$, the algorithm was able to identify the clusters correctly, and it converged in 12 iterations (**Fig. 4**).

In the two moons dataset, 400 points were arranged in two crescent-shaped clusters with boundaries that overlap. More complexity was introduced by the Gaussian noise with a standard deviation of 0.1. When kernel transformation with $\sigma = 0.5$ was used the separability of the clusters was improved, such that the algorithm was able to correctly cluster the data with $\lambda = 0.01$. Convergence was reached in just 4 iterations (**Fig. 5**).



(a): Data distribution before clustering (b): Clustering results with cluster centers (c): Objective function convergence

Figure 4. Clustering results for the two concentric circles dataset.



(a): Data distribution before clustering (b): Clustering results with cluster centers (c): Objective function convergence

Figure 5. Clustering results for the two moons dataset.

Therefore, the results of the presented method have shown the flexibility and robustness of the regularized k-means an algorithm for a variety of clustering scenarios, starting from simple linear structures and ending with complex nonlinear patterns. λ and σ were crucial to select for best performance, and the fast convergence of the objective function in all experiments indicates the method's computational stability. This is a solid basis to further extend the algorithm to more complex, real-world datasets.

6.3 Breast Cancer Gene Expression Data: Analysis and Comparison

The breast cancer gene expression data was used to assess the efficacy of the regularized k-means algorithm on high-dimensional real-world datasets. In this section, regularized k-means is compared with different classification methods, and the datasets (GSE42568, GSE45827, and GSE10797) are examined based on their characteristics, clustering performance, and statistical metrics **(Table 1).**

For the high dimensionality, the Kruskal-Wallis test was employed to select 200 statistically significant features. This feature reduction maintains low computational cost and retains the most significant information for clustering.

After feature selection, regularized k-means was applied to each dataset. The performance of the algorithm was optimized by tuning the regularization parameter (λ) and choosing the one that produced the lowest objective value.

Table 1. Dataset breast cancer characteristics.

Dataset	Samples	Genes	Classes	Platform	Description
GSE42568	116	54,676	2	GPL570	Breast tissue samples, including normal and cancer subtypes.
GSE10797	66	22,278	3	GPL571	Breast tissue samples were classified into normal, stromal, and epithelial classes.
GSE45827	151	54,676	6	GPL570	Breast cancer samples were categorized into HER2+, Basal, Luminal A, Luminal B, normal-like, and cancer epithelial cell lines

Table 2 shows the Precision and F1-score metrics for each class across datasets. Regularized k-means was compared to k-means and other classification methods, such as support vector machines (SVM), random forest (RF), and others from **(Feltes et al., 2019; Grisci et al., 2019)**. The comparison of accuracy across these methods is presented in **Table 3**. As shown in **Figs. 6 to 8**, the clustering results and the objective function convergence for the datasets with respect to a number of iterations.

Table 2. Precision and F1-score for each dataset.

Metric	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Dataset
Precision	1.0000	0.9902	-	-	-	-	Breast_GSE42568
Precision	0.8621	0.8889	1.0000	-	-	-	Breast_GSE10797
Precision	0.9762	1.0000	1.0000	0.8750	0.9655	0.9655	Breast_GSE45827
F1-Score	0.9655	0.9951	-	-	-	-	Breast_GSE42568
F1-Score	0.8772	0.8727	1.0000	-	-	-	Breast_GSE10797
F1-Score	0.9880	0.9831	1.0000	0.9333	0.9655	0.9492	Breast_GSE45827

Table 3. Accuracy comparison of regularized k-means and other classification methods.

Dataset	Regularized k-means	k-means	SVM	Random Forest	ZeroR	HC	NB	DT	KNN	MLP
GSE42568	**99.14%**	62%	99%	97%	87%	88%	99%	94%	98%	99%
GSE10797	**89.39%**	58%	82%	65%	41%	44%	67%	65%	55%	53%
GSE45827	**97.35%**	70%	94%	95%	27%	34%	93%	80%	80%	58%

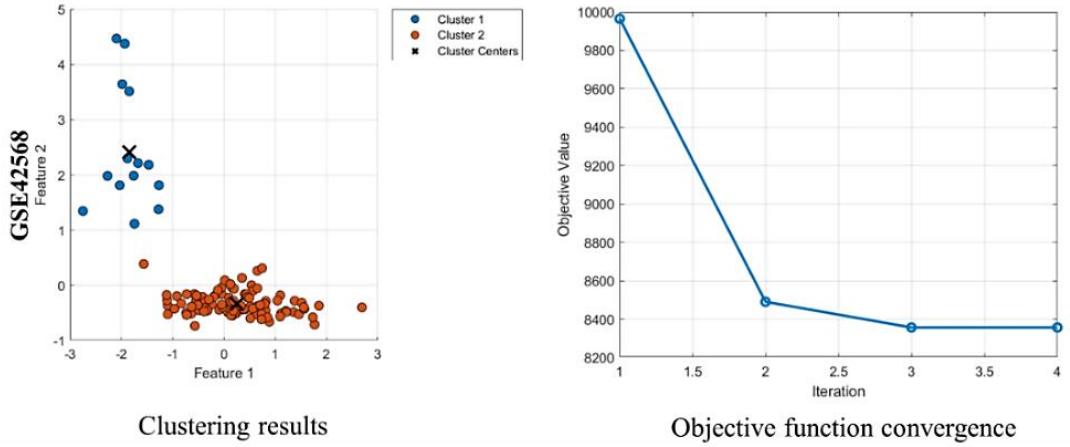


Figure 6. Clustering results for dataset GSE42568, the data was clustered into two groups with a regularization parameter of $\lambda = 0.001$.

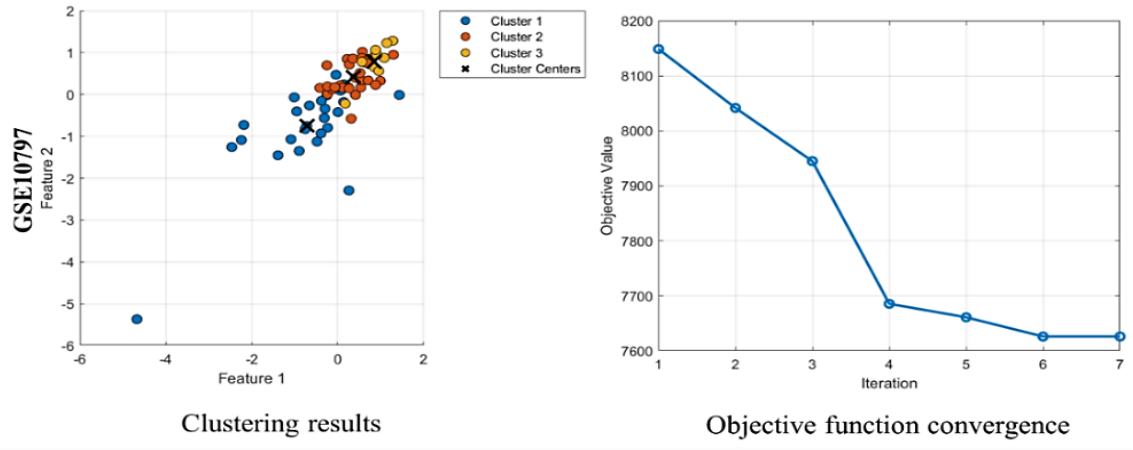


Figure 7. Clustering results for dataset GSE10797, the data was clustered into two groups with a regularization parameter of $\lambda = 0.005$.

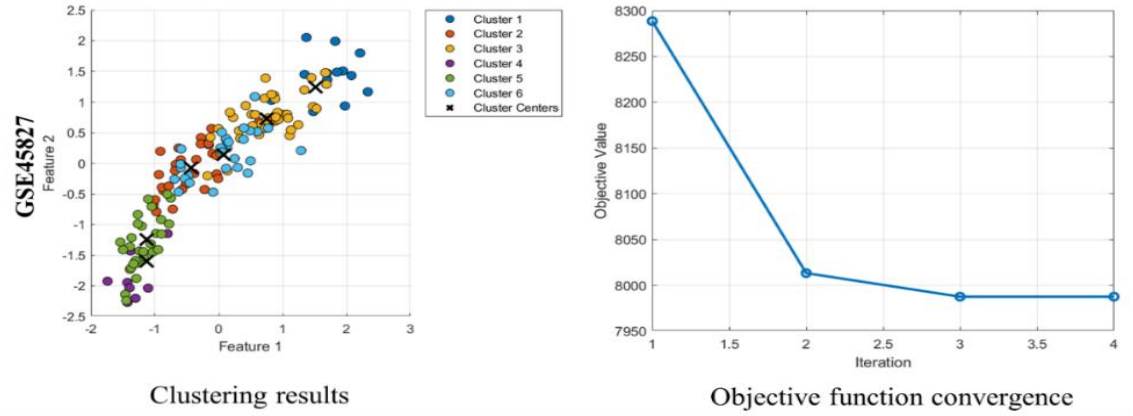


Figure 8. Clustering results for dataset GSE45827, the data was clustered into two groups with a regularization parameter of $\lambda = 0.009$.

6.4 Comparative Performance Assessment on scRNA-seq Datasets

This experiment aims to evaluate the proposed FCFW-regularized k-means an algorithm on challenging single-cell RNA sequencing (scRNA-seq) clustering tasks and compares its performance against existing state-of-the-art methods. Two well-known scRNA-seq benchmark datasets were utilized, as summarized in **Table 4.**

The Deng dataset contains single-cell gene expression profiles from mouse embryonic development that were first described by **(Deng et al., 2014)**. The dataset includes 7 distinct cell types, which represent different stages of embryonic development starting from zygote through 2-cell, 4-cell, 8-cell, morula, and blastocyst. The Goolam dataset **(Goolam et al., 2016)** contains mouse embryo cells at 5 cell-type stages from 2-cell to 16-cell stage with a total of 124 single cells.

For each dataset, the top 500 genes with the highest KW significance (lowest p-values) were selected as features for clustering. After feature selection, the FCFW-regularized k-means a clustering algorithm was applied to each scRNA-seq dataset. In this method, k-means clustering is enhanced with a Frobenius-norm regularization term and optimized using the Fully Corrective Frank–Wolfe (FCFW) iterative procedure.

Table 4. Dataset Goolam and Deng characteristics.

Dataset	Samples	Genes	Classes	Description
Goolam	124	40,315	5	Mouse embryonic cells at different developmental stages from oocyte to blastocyst.
Deng	135	12,548	7	Mouse preimplantation embryos covering multiple stages, focusing on gene expression dynamics and allele-specific patterns.

The evaluation of clustering results used multiple established metrics to assess the predicted cluster labels against the actual cell type labels, used Normalized Mutual Information (NMI) **(Strehl and Ghosh, 2002)** and Adjusted Rand Index (ARI) **(Meilă, 2007)** to measure between clustering results and truth clustering. The clustering accuracy, which values between 0 and 1, where 1.0 represents a complete recovery of true classes. The class-wise metrics presented in **Table 5** for each dataset show how the clustering performance distributes across different classes.

Table 5. Precision and F1-score of regularized k-means for the Goolam and Deng dataset.

Metric	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Dataset
Precision	1.0000	0.9846	1.0000	0.6667	0.6667	-	-	Goolam
Precision	0.6667	0.4737	0.5263	0.9333	0.9189	0.9583	0.3333	Deng
F1-Score	0.9677	0.9922	1.0000	0.6667	0.6667	-	-	Goolam
F1-Score	0.5714	0.4865	0.5128	0.9655	0.9315	0.9388	0.4000	Deng

To contextualize the performance of FCFW-regularized k-means we compared it against nine existing clustering methods relevant to scRNA-seq data analysis. These include classical methods and recent specialized algorithms: NMF (non-negative matrix factorization-based clustering), SIMLR (a multi-kernel learning method for single-cell clustering), and several dedicated single-cell clustering techniques (scCCL, scBKAP, scMCKC, scDCC, scDSSC, SSRE) **(Wei et al., 2025)**.

The FCFW-regularized k-means was reported to achieve superior performance on numerous scRNA-seq datasets, making it a strong competitor for our evaluation. **Table 6** presents the clustering accuracy results for all methods on the Deng and Goolam datasets, in terms of NMI and ARI, we visualized the clustering assignments and convergence trends for each dataset (**Figs. 9 and 10**).

Table 6. Comparative clustering performance of FCFW-regularized k-means.

Method	NMI Deng	ARI Deng	NMI Goolam	ARI Goolam
NMF	0.605	0.356	0.572	0.404
SIMLR	0.639	0.384	0.731	0.608
scCCL	0.766	0.589	0.742	0.790
scBKAP	0.743	0.477	0.683	0.517
scMCKC	0.717	0.524	0.789	0.644
scDCC	0.726	0.525	0.661	0.440
scDSSC	0.637	0.379	0.601	0.559
SSRE	0.813	0.650	0.829	0.668
scPEDSSC	0.785	0.729	0.878	0.885
Regularized k-means	**0.819**	**0.801**	**0.922**	**0.968**

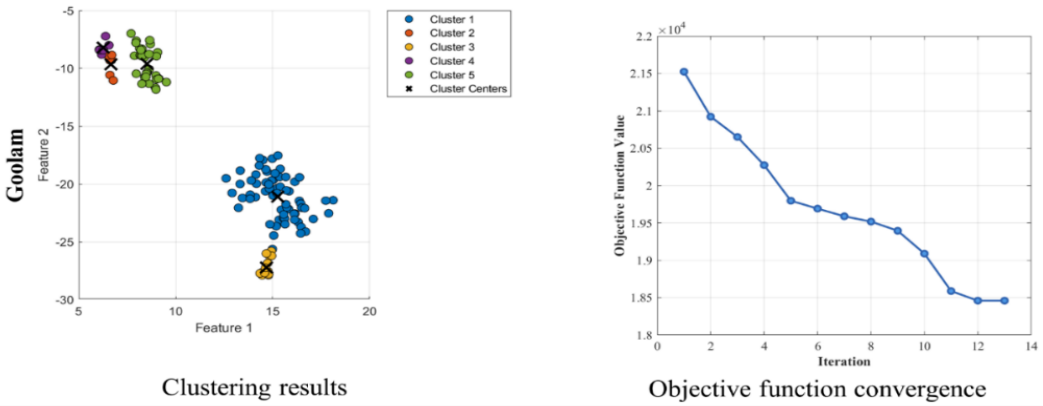


Figure 9. Clustering results for dataset Goolam, the data was clustered into two groups with a regularization parameter of $\lambda = 0.013459$.

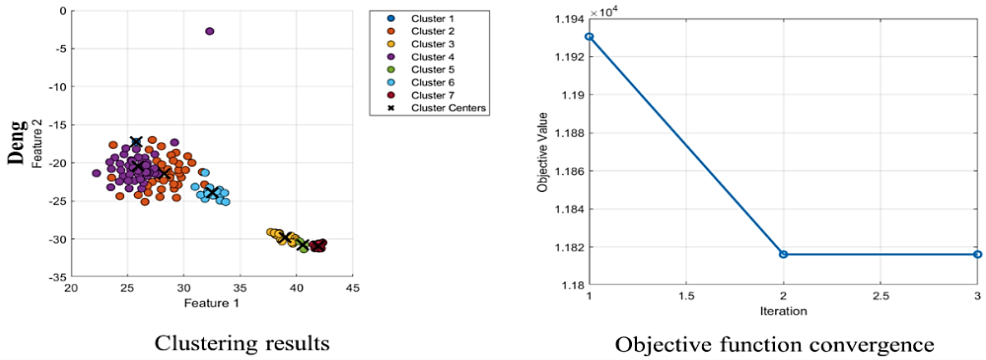


Figure 10. Clustering results for the dataset were clustered into five groups with a regularization parameter of $\lambda = 0.019442$.

7. RESULTS AND DISCUSSION

The proposed FCFW-regularized k-means approach was evaluated on five gene expression datasets—three bulk gene expression sets (GSE42568, GSE10797, GSE45827) and two single-cell RNA-seq sets (Deng and Goolam). Across all datasets, the method demonstrated fast convergence and accurate clustering, as evidenced by the visualization of cluster assignments alongside objective function values. In each case, the objective function dropped sharply in the first few iterations and quickly reached a stable minimum, indicating efficient optimization. For example, in the GSE42568 dataset, the algorithm converging to a final objective value 8.36×10^3 after only four iterations, successfully separating the samples into two distinct clusters. Likewise, in the GSE10797 dataset, the objective value fell to 7.63×10^3 within seven iterations, and the resulting three clusters aligned with the known tissue categories. Even for the more complex GSE45827 breast cancer dataset, the method required only about four iterations to converge 7.98×10^3 and produced six clear clusters corresponding to distinct tumor subtypes. These results highlight not only the effectiveness of the FCFW optimizer, which rapidly minimizes the k-means objective, but also the value of the Kruskal-Wallis feature selection in preprocessing. By selecting the top 200 most informative genes for each microarray dataset, the algorithm focused on relevant features, leading to well-differentiated clusters of the expected classes (e.g., separating normal vs. tumoral samples or distinguishing different cancer subpopulations) and avoiding noise from thousands of irrelevant genes.

For the single-cell datasets, Deng and Goolam, the proposed method similarly achieved strong performance in clustering cells into their known developmental stages while maintaining fast convergence. In the Goolam dataset, the objective function value steadily decreased and stabilized after roughly 13 iterations, with the algorithm converged to a final objective value 18.460×10^3, despite the high dimensionality, indicating that the FCFW-based optimization efficiently handled the large feature space. In the Deng dataset, the method also performed well, grouping cells into seven clusters that align with the expected embryonic stages. The objective function converged extremely quickly to 11.816×10^3 after just 3 iterations, remaining stable thereafter, underscoring the efficiency of the algorithm even as the number of clusters grows. **Fig. 11** presents the p-values of differences in the expression levels between sample types. **Figs. 12 and 13** shows the boxplots of the gene expression levels for three datasets.

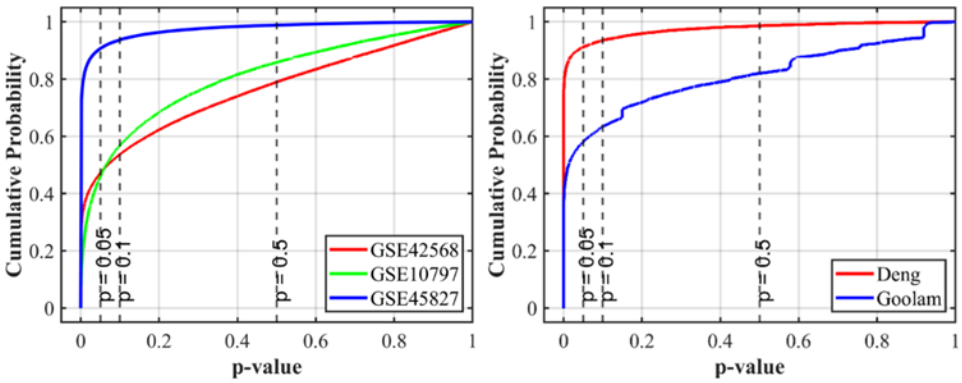


Figure 11. Cumulative distribution function (CDF) of p-values for GSE datasets

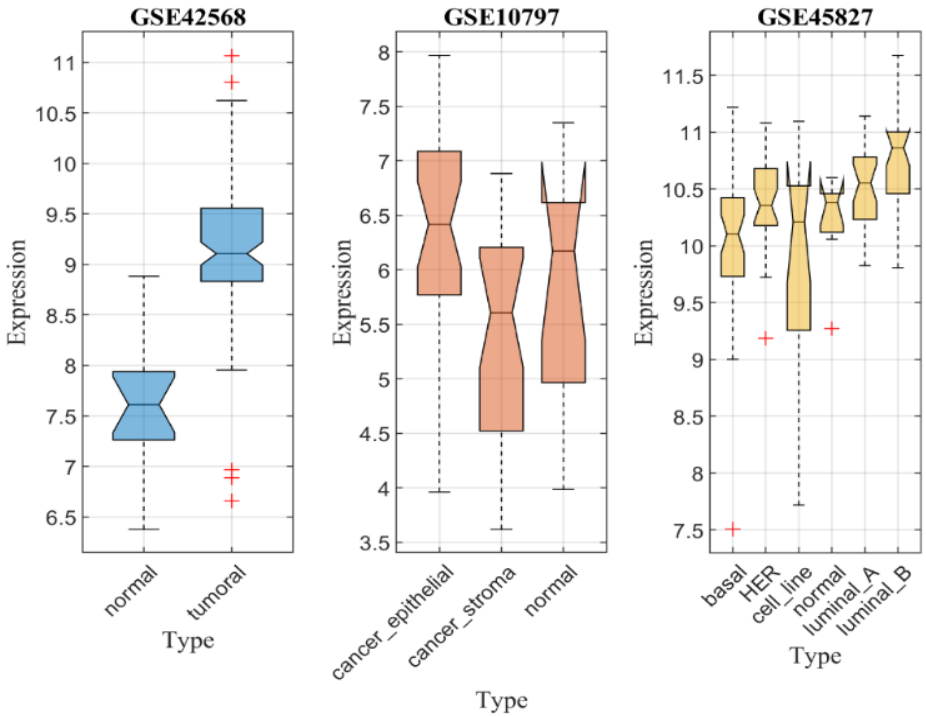


Figure 12. Boxplots of the expression of the gene across the different sample types in the datasets GSE42568, GSE10797, and GSE45827. The x-axis is the sample types, and the y-axis is the expression levels.

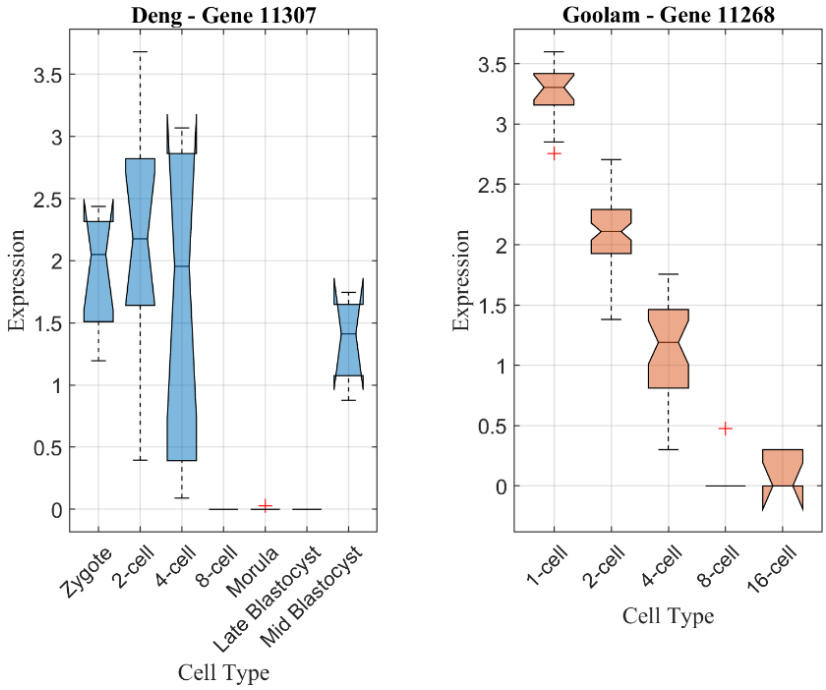


Figure 13. Boxplots of the expression of the gene across the different sample types in the datasets Deng and Goolam. The x-axis is the sample types, and the y-axis is the expression levels.

8. CONCLUSION

The paper introduces the FCFW algorithm that enhances the regularized *k*-means clustering framework, especially for high-dimensional data. The proposed regularization method using Frobenius norm maintains an appropriate balance between cluster compactness and separation, and the probabilistic assignment of data points provides robustness against noise while allowing for overlapping clusters. The FCFW algorithm accelerates convergence speed through active solution re-optimization that reduces redundancy and improves stability. The experiments on synthetic datasets demonstrate how kernel transformations improve cluster separability. The proposed algorithm achieves better performance than traditional *k*-means and other classification methods when applied to real-world expression data. The algorithm's robustness becomes evident through extensive preprocessing techniques, which include *Z*-score normalization and feature selection through the Kruskal-Wallis test and optimization of the regularization parameter (λ) for multi-class and imbalanced datasets. The algorithm demonstrates its practicality for real-world applications through its improved accuracy, F1-score, NMI and ARI performance in high-dimensional data analysis. The research demonstrates that FCFW-regularized *k*-means functions as an efficient and robust clustering framework suitable for various high-dimensional datasets, especially in biomedical applications. The future research direction involves optimizing the algorithm for extensive datasets and merging it with deep learning approaches to boost performance in complex clustering operations.

NOMENCLATURE

Symbol	Description	Symbol	Description
Z	Dataset matrix of size $n \times d$	C	Cluster centers matrix of size $k \times d$
X	Cluster assignment matrix	k	Number of clusters
n	Number of data points	d	Data dimension (number of features)
λ	Regularization parameter	$\| C \|_F^2$	Frobenius norm of the cluster centers matrix
$\nabla f(C)$	Gradient of the objective function	v_t	Descent direction in the Frank-Wolfe method
S_t	Active set of cluster centers in FCFW	γ_t	Step size parameter
H	Kruskal-Wallis test statistic	K_{ij}	Gaussian kernel function
O_t	Objective function value at iteration t	σ, τ	Standard deviation
$\mathbb{R}^d$	d-dimensional real space	C_j	Cluster center j
z_i	Data point i in d-dimensional space	$\sum_{j=1}^{k} X_{ij}$	Sum of assignments for data point i across all clusters
X_{ij}	Assignment of data point i to cluster j	$\min_{C,X}$	Optimization problem in regularized k-means

Acknowledgements

The authors express their sincere gratitude to the University of Baghdad for providing the necessary resources and support that facilitated the successful completion of this research.

Credit Authorship Contribution Statement

Ahmed Yacoub Yousif: Conceptualization, methodology, formal analysis, writing-original draft, software implementation, validation. Basad Al-Sarray: Supervision, review, editing, validation.

Declaration of Competing Interest

The authors declare that they have no financial or other material conflicts of interest that could be construed as affecting the results or interpretation of their manuscript.

REFERENCES

Adams, R.P., 2018. K-means clustering and related algorithms. *Princeton University*.

Ahmed, V.M. and Al-Haleem, A.A., 2024. Permeability prediction for Ajeel Oilfield/Tertiary Reservoir by integrating rock typing approach with FZI method. *Journal of Engineering*, 30(12), pp. 96–111. https://doi.org/10.31026/j.eng.2024.12.07.

AL-Kordy, S.U. and Khudair, B.H., 2021. Effluent quality assessment of sewage treatment plant using principal component analysis and cluster analysis. *Journal of Engineering*, 27(4), pp. 79–95. https://doi.org/10.31026/j.eng.2021.04.07.

Beznosikov, A., Dobre, D. and Gidel, G., 2023. Sarah frank-wolfe: Methods for constrained optimization with best rates and practical features. *arXiv preprint arXiv:2304.11737*.

Blanza, J., 2021. Wireless propagation multipaths using spectral clustering and three-constraint affinity matrix spectral clustering. *Baghdad Science Journal*, 18(2), P. 1001. https://doi.org/10.21123/bsj.2021.18.2(Suppl.).1001.

Canon, M.D. and Cullum, C.D., 1968. A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. *SIAM Journal on Control*, 6(4), pp. 509–516. https://doi.org/10.1137/0306032.

Cherfaoui, F., Emiya, V., Ralaivola, L. and Anthoine, S., 2018. Frank-Wolfe algorithm for the exact sparse problem. *arXiv preprint arXiv:1812.07201*.

Daoudi, S., Anouar Zouaoui, C.M., El-Mezouar, M.C. and Taleb, N., 2021. Parallelization of the K-means++ clustering algorithm. *Ingénierie des Systèmes d'Information*, 26.(1)

Deng, Q., Ramsköld, D., Reinius, B. and Sandberg, R., 2014. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science,* 343(6167), pp. 193–196. https://doi.org/10.1126/science.1245316.

Feltes, B.C., Chandelier, E.B., Grisci, B.I. and Dorn, M., 2019. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. *Journal of Computational Biology*, 26(4), pp. 376–386. https://doi.org/DOI:10.1089/cmb.2018.0238.

Frank, M. and Wolfe, P., 1956. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1–2), pp. 95–110. https://doi.org/10.1002/nav.3800030112.

Gao, C.X., Dwyer, D., Zhu, Y., Smith, C.L., Du, L., Filia, K.M., Bayer, J., Menssink, J.M., Wang, T. and Bergmeir, C., 2023. An overview of clustering methods with guidelines for application in mental health research. *Psychiatry Research*, 327, P. 115265. https://doi.org/10.1016/j.psychres.2023.115265.

Ghathwan, K.I. and Mohammed, A.J., 2022. Intelligent bat algorithm for finding eps parameter of DbScan clustering algorithm. *Iraqi Journal of Science,* pp. 5572–5580. https://doi.org/10.24996/ijs.2022.63.12.41.

Gondeau, A., Aouabed, Z., Hijri, M., Peres-Neto, P.R. and Makarenkov, V., 2019. Object weighting: A new clustering approach to deal with outliers and cluster overlap in computational biology. *IEEE/ACM transactions on computational biology and bioinformatics,* 18(2), pp. 633–643. https://doi.org/10.1109/TCBB.2019.2921577.

Goolam, M., Scialdone, A., Graham, S.J.L., Macaulay, I.C., Jedrusik, A., Hupalowska, A., Voet, T., Marioni, J.C. and Zernicka-Goetz, M., 2016. Heterogeneity in Oct4 and Sox2 targets biases cell fate in 4-cell mouse embryos. *Cell,* 165(1), pp. 61–74. https://doi.org/10.1097/01.ogx.0000488738.30718.bf.

Grisci, B.I., Feltes, B.C. and Dorn, M., 2019. Neuroevolution as a tool for microarray gene expression pattern identification in cancer research. *Journal of Biomedical Informatics,* 89, pp. 122–133. https://doi.org/10.1016/j.jbi.2018.11.013.

He, Y. and Zheng, Y., 2018. Short-term power load probability density forecasting based on Yeo-Johnson transformation quantile regression and Gaussian kernel function. *Energy,* 154, pp. 143–156. https://doi.org/10.1016/j.energy.2018.04.072.

Holloway, C.A., 1974. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming,* 6, pp. 14–27. https://doi.org/10.1007/BF01580219.

Jamail, I. and Moussa, A., 2020. Current state-of-the-art of clustering methods for gene expression data with RNA-Seq. In: *Applications of Pattern Recognition*. IntechOpen. https://doi.org/10.5772/intechopen.94069.

Jiang, P., Cao, J., Yu, W. and Nie, F., 2025. A robust entropy regularized K-means clustering algorithm for processing noise in datasets. *Neural Computing and Applications,* pp. 1–16. https://doi.org/10.1007/s00521-024-10899-4.

Lacoste-Julien, S. and Jaggi, M., 2015. On the global linear convergence of Frank-Wolfe optimization variants. *Advances in neural information processing systems*, 28.

Lacoste-Julien, S., Jaggi, M., Schmidt, M. and Pletscher, P., 2013. Block-coordinate Frank-Wolfe optimization for structural SVMs. In: *International Conference on Machine Learning.* PMLR. pp. 53–61.

Lei, T., Jia, X., Zhang, Y., He, L., Meng, H. and Nandi, A.K., 2018. Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering. *IEEE Transactions on Fuzzy Systems,* 26(5), pp. 3027–3041. https://doi.org/10.1109/TFUZZ.2018.2796074.

LI, X., CAI, L., LI, J., YU, C.K.W.A.I. and HU, Y., 2021. A survey of clustering methods via optimization methodology. *Journal of Applied & Numerical Optimization,* 3(1). https://doi.org/10.23952/jano.3.2021.1.09.

Liu, H., Chen, J., Dy, J. and Fu, Y., 2023. Transforming complex problems into K-means solutions. *IEEE transactions on pattern analysis and machine intelligence,* 45(7), pp. 9149–9168.

Mahdi, S.S. and Mahmood, R.S., 2014. MR brain image segmentation using spatial fuzzy C-means clustering algorithm. *Journal of Engineering,* 20(09), pp. 78–89. https://doi.org/10.31026/j.eng.2014.09.06.

Meilă, M., 2007. Comparing clusterings—An information based distance. *Journal of multivariate analysis,* 98(5), pp. 873–895. https://doi.org/10.1016/j.jmva.2006.11.013.

Meléndez Surmay, R., Giraldo Henao, R. and Rodríguez Cortes, F., 2024. Kruskal-Wallis test for functional data based on random projections generated from a simulation of a Brownian motion. *TecnoLógicas,* 27(59).

Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghiren, E., Ameh, F., Achas, M. and Adebiyi, E., 2016. Clustering algorithms: Their application to gene expression data. *Bioinformatics and Biology insights,* 10, P. BBI-S38316. https://doi.org/10.4137/BBI.S38316.

Raeisi, M. and Sesay, A.B., 2022. A distance metric for uneven clusters of unsupervised k-means clustering algorithm. *IEEE Access,* 10, pp. 86286–86297. https://doi.org/10.1109/ACCESS.2022.3198992.

Raymaekers, J. and Zamar, R.H., 2022. Regularized k-means through hard-thresholding. *Journal of Machine Learning Research*, 23(93), pp. 1–48.

Saha, J., Tanvir, R.H., Hassan Samee, M.A. and Rahman, A., 2023. Probabilistic clustering of cells using single-cell RNA-seq data. *bioRxiv,* pp. 2012–2023.

Salman, A. and Hussain, B.A., 2023. Gene expression analysis via spatial clustering and evaluation indexing. *Iraqi Journal for Computer Science and Mathematics,* 4(1), pp. 24–34. https://doi.org/10.52866/ijcsm.2023.01.01.004.

Sarray, B. Al, Chrétien, S., Clarkson, P. and Cottez, G., 2017. Enhancing Prony's method by nuclear norm penalization and extension to missing data. *Signal, Image and Video Processing*, 11, pp. 1089–1096.

Shiltagh, N.A. and Hussein, M.A., 2015. Data aggregation in wireless sensor networks using modified Voronoi fuzzy clustering algorithm. *Journal of Engineering,* 21(4), pp. 42–60. https://doi.org/10.31026/j.eng.2015.04.03.

Strehl, A. and Ghosh, J., 2002. Cluster ensembles---a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec), pp. 583–617. https://doi.org/10.1162/153244303321897735.

Sun, W., Wang, J. and Fang, Y., 2012. Regularized k-means clustering of high-dimensional data and its asymptotic consistency.

Teran Hidalgo, S.J., Zhu, T., Wu, M. and Ma, S., 2018. Overlapping clustering of gene expression data using penalized weighted normalized cut. *Genetic epidemiology,* 42(8), pp. 796–811.

Wei, X., Wu, J., Li, G., Liu, J., Wu, X. and He, C., 2025. scPEDSSC: Proximity enhanced deep sparse subspace clustering method for scRNA-seq data. *PLOS Computational Biology*, 21(4), P. e1012924. https://doi.org/10.1371/journal.pcbi.1012924.

Wirth, E., Pena, J. and Pokutta, S., 2024. Fast Convergence of Frank-Wolfe algorithms on polytopes. arXiv preprint arXiv:2406.18789.

Wu, Z. and Wu, Z., 2020. An enhanced regularized k-means type clustering algorithm with adaptive weights. IEEE Access, 8, pp. 31171–31179. https://doi.org/10.1109/ACCESS.2020.2972333.

Yang, X., Zhao, W., Xu, Y., Wang, C.-D., Li, B. and Nie, F., 2024. Sparse K-means clustering algorithm with anchor graph regularization. Information Sciences, 667, P. 120504. https://doi.org/10.1016/j.ins.2024.120504.

Yousif, A.Y. and Sarray, B. Al, 2024. Convex optimization techniques for high-dimensional data clustering analysis: A review. Iraqi Journal for Computer Science and Mathematics, 5(3), P. 29. https://doi.org/10.52866/ijcsm.2024.05.03.022.

Zhang, X., He, Y., Jin, Y., Qin, H., Azhar, M. and Huang, J.Z., 2020. A robust k-means clustering algorithm based on observation point mechanism. Complexity, 2020(1), P. 3650926. https://doi.org/10.1155/2020/3650926.

تجميع البيانات باستخدام *k*-means المنتظم عبر تحسين فرانك–وولف التصحيحي الكامل

أحمد يعقوب يوسف [2،1]*, بسعاد علي السراي [3]

[1] قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق

[2]قسم العلوم التطبيقية، الجامعة التكنولوجية، بغداد، العراق

[3]قسم علوم الحاسوب ، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

لا يزال تجميع البيانات عالية الأبعاد يمثل تحديًا كبيرًا، نظرًا لحساسية خوارزمية k-means التقليدية للضوضاء والقيم الشاذة وارتفاع الأبعاد، مما يؤدي غالبًا إلى أداء غير مستقر. يعرض هذا البحث نظام تجميع قوي يجمع بين خوارزمية Fully Corrective Frank-Wolfe (FCFW) مع هدف k-means المدعوم بتنظيم يعتمد على معيار فروبينيوس. يسهم إدراج تنظيم معيار فروبينيوس في النموذج في إنتاج مجموعات أكثر استقرارًا، مع تقليل احتمالية الإفراط في التكيف وتعزيز تماسك التجمعات. يعتمد الأسلوب المقترح على تخصيص احتمالي للبيانات، مما يسمح لكل نقطة بيانات بالانتماء إلى عدة مجموعات بدرجات عضوية مختلفة، وبالتالي دعم التجمعات ذات الحدود المتداخلة. كما تم استخدام اختبار Kruskal-Wallis كطريقة لاختيار الميزات لتحديد الجينات الهامة التي توجه عملية التجميع نحو الخصائص ذات الأهمية في البيانات عالية الأبعاد. وقد تفوقت خوارزمية k-means المنتظمة بـ FCFW على k-means التقليدية في جميع التجارب التي أُجريت على بيانات تركيبية وحقيقية لتعبير الجينات. فعلى سبيل المثال، حققت دقة بنسبة 89.39% على مجموعة بيانات لتعبير جينات سرطان الثدي (GSE10797)، مقارنة بـ 58% فقط لـ k-means التقليدية. كما تفوقت على طريقة حديثة للتجميع في الفضاء العميق الفرعي (scPEDSSC) في مؤشر Adjusted Rand Index بنسبة 8.3% على مجموعة بيانات Goolam للخلايا الأحادية (0.968 مقابل 0.885)، وبنسبة 7.2% على مجموعة بيانات Deng (0.801 مقابل 0.729). بشكل عام، حقق النهج المقترح أعلى القيم في مؤشري ARI وNMI عبر خمس مجموعات بيانات معيارية، مما يؤكد أن k-means المنتظمة باستخدام FCFW توفر نتائج تجميع أكثر دقة واستقرارًا، وتظهر أداءً قويًا في التعامل مع البيانات عالية الأبعاد.

الكلمات المفتاحية: التجميع، k–means المُنظَّم، خوارزمية فرانك–وولف المصححة بالكامل.