Project 1: Frog Tail

Course: STATGR5243_001_2025_3 – Applied Data Science

Professor: Dr. Bianca Dumitrascu, TA: Jonathan Huml
Student: Bryndis Lif Bjarnadottir (blb2181)

ABSTRACT

Researchers have identified the regeneration-organizing cell (ROC) as responsible for tail regeneration in the Xenopus laevis tadpole. This project replicates those findings using K-means and Leiden clustering, focusing mainly on Leiden to stay consistent with the original study. K-means achieved higher Silhouette (0.447 vs. 0.280) and Calinski-Harabasz (2419.06 vs. 2157.75) scores, showing better separation and compactness than Leiden. Applying the denoising methods MAGIC and KNN smoothing improved these scores, while the batch integration techniques Harmony and Combat lowered them. Without denoising or batch integration, a Leiden cluster labeled as epidermis (no cluster was labeled ROC) appeared in the same UMAP region as the ROC cells reported in the paper. This is consistent with their finding that ROC cells are located in the epidermis. In all other cases, a ROC-like cluster was identified in the same UMAP space as the original study. This confirms that this analysis successfully detected the ROC cell subset.

1 INTRODUCTION

The Xenopus laevis tadpole can regenerate its tail after amputation at certain stages. Researchers in [1] used single-cell RNA sequencing to find a new cell type that is responsible for this process. This cell, called the regeneration-organizing cell (ROC), coordinates regeneration by moving to the wound site and forming a wound epidermis. It then releases signals that stimulate other cells to divide and rebuild the tail. Here, the objective is to replicate the results of [1] to better understand their methods.

2 METHODS

2.1 Clustering Analysis

First, two clustering algorithms, K-means and Leiden, were applied to the data and their results compared. The optimal number of clusters in K-means was determined using the elbow method. Each cluster was then annotated based on the gene that was most highly expressed within it. To assess clustering performance, the Silhouette score, Adjusted Rand Index (ARI) and Calinski-Harabasz scores were calculated.

The Silhouette coefficient ranges from -1 to +1. A higher positive value means the sample fits well within its assigned cluster, a value near 0 means it lies close to a cluster boundary and a negative value indicates that a point was misclassified. The overall Silhouette score is calculated as the mean of all individual Silhouette coefficients across all samples. ARI is a normalized version of the Rand Index that measures the agreement between two clustering methods [2]. A score of +1 indicates perfect agreement between the two clustering methods, 0 corresponds to random labeling and a negative value indicates worse than random agreement [3]. The Calinski-Harabasz score, which is not discussed in the class textbook, is the ratio of how spread apart the clusters are from one another to how compact the points are inside each cluster [4]. A higher score means that the clusters are better separated and clearly defined.

2.2 Marker Selection and Gene Analysis

Of the two clustering methods, the Leiden method was used moving forward, consistent with the approach in [1]. The resulting clusters were compared to the known ROC genes from Supplementary Table 3 in [1]. Each cell was then assigned a ROC score based on how strongly it expressed these genes. The Leiden cluster with the highest ROC score was considered the most ROC-like cluster. To determine which genes best defined the cluster, two marker selection methods were applied, logistic regression and Wilcoxon rank sum test. The genes identified by both methods were then compared with each other and with those listed in Supplementary Table 3.

2.3 Data Denoising

Two denoising methods were applied separately, and their results were evaluated by comparing the clustering analysis and marker selection outcomes with and without the denoising. The MAGIC (Markov Affinity-based Graph Imputation of Cells) algorithm denoises single-cell RNA sequencing data by sharing information between similar cells. This reduces noise and dropout effects. KNN smoothing denoises the data by identifying each cell's k nearest neighbors using Euclidean distance and averaging its gene expression with that of its neighbors. Both these methods result in smoother data for clustering [5].

2.4 Batch Integration over Time

Lastly, two batch integration methods were applied to the non-denoised data, and their effects on clustering analysis and marker selection were evaluated. Batch integration removes technical differences between datasets from different conditions or experiments while preserving true biological variation. The algorithm Harmony does this by aligning similar cell types into a shared space [6], whereas Combat applies an empirical Bayes approach to adjust gene expression bias across batches [5].

2.5 Code Availability

The code used in the analysis is publicly available at:
`https://github.com/b-lb/Project_1_Frog_Tail/blob/main/FINAL_FROGTAIL.ipynb`

3 RESULTS

3.1 Clustering Analysis

The clusters identified by the two clustering methods, K-Means and Leiden, are shown in Figures 1 and 2, respectively. The K-means clustering produced nine clusters, as determined by the elbow method. These clusters appear relatively compact and well separated in the UMAP space. In contrast, the Leiden algorithm generated more clusters, suggesting it is more sensitive to finer distinctions between cells. However, the clusters are less clearly separated in the UMAP space, as indicated by the evaluation metrics shown in Table 1.

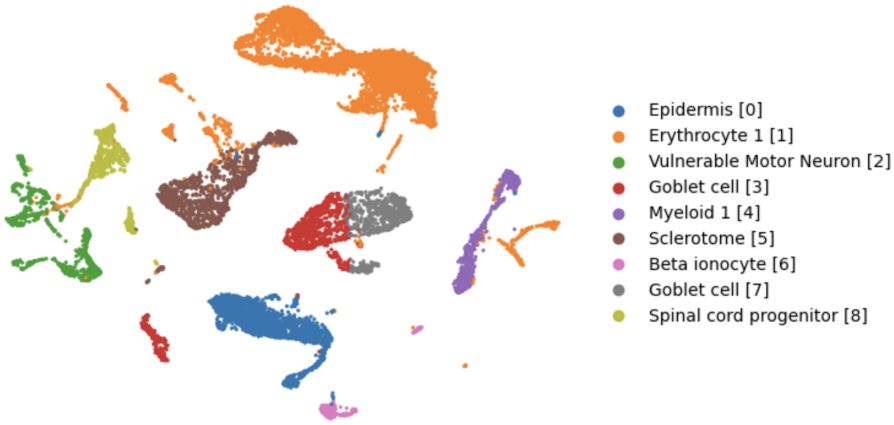

Figure 1. UMAP visualization of K-Means clustering with K = 9 as determined by the elbow method. This was done on the dataset without denoising or batch integration.

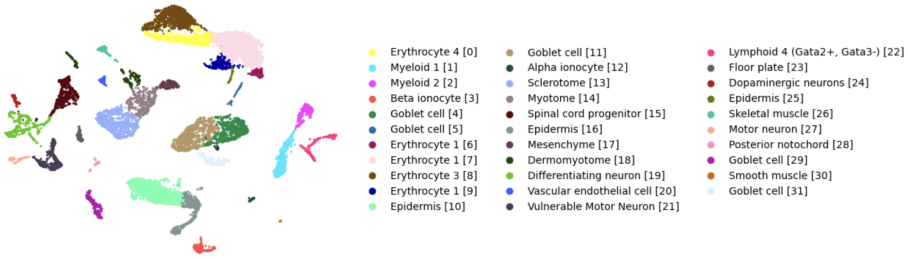

Figure 2. UMAP visualization of Leiden clustering on the dataset without denoising or batch integration.

Model	Silhouette	Calinski-Harabasz	ARI
K-Means	0.448	2419.06	-
Leiden	0.280	2157.75	-
K-Means vs Leiden	-	-	0.424

Table 1. Clustering evaluation metrics for K-Means and Leiden clustering methods on the dataset without denoising or batch integration.

The results in Table 1 show that K-means scores higher than Leiden in both the Silhouette and Calinski–Harabasz tests. This means that K-means clusters are more compact and better separated. The ARI score of 0.424 indicates a moderate agreement between the methods in how they assign cells to clusters.

3.2 Marker Selection and Gene Analysis

A total of 33 ROC genes were found in the dataset (out of 50 in the Supplementary Table 3). The missing genes could be due to differences in preprocessing or analysis parameters, or because of slight name mismatches. This was accounted for as much as possible by removing the long and short suffixes (".L" and ".S") from the gene names before comparison. Each Leiden cluster then was assigned a ROC score and cluster 16 had the highest score, meaning it is likely a ROC cluster. Logistic regression identified 12 genes in that cluster that are also listed in Supplementary Table 3, whereas Wilcoxon rank sum test only identified 2.

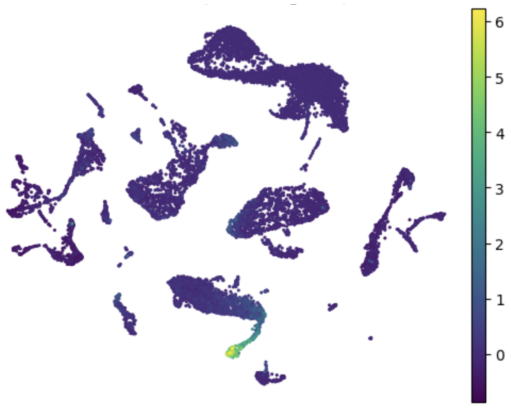

Figure 3. UMAP visualization of the location of the ROC genes according to [1]. Yellow indicates high expression of ROC genes while purple indicates low or no expression.

Figure 2 shows the Leiden clusters and Figure 3 shows the expression of ROC genes reported in [1]. The ROC genes are expressed in the same region as cluster 16 in the Leiden clustering. This cluster is labeled as epidermis, which agrees with the findings of [1], where ROC-like cells were identified in the epidermis. However, it is worth noting that Leiden clusters 10 and 25 are also labeled as epidermis, but according to the paper, no ROC cells were found in those clusters.

3.3 Effects of Denoising

3.3.1 MAGIC

The Leiden clustering identified 54 clusters when the MAGIC denoising algorithm was applied, compared to 31 before denoising. The Silhouette score increased to 0.328 and the Calinski-Harabasz score also improved (3036.22). Unlike before denoising, it was now higher for Leiden than for K-means. However, the ARI score decreased to 0.216.

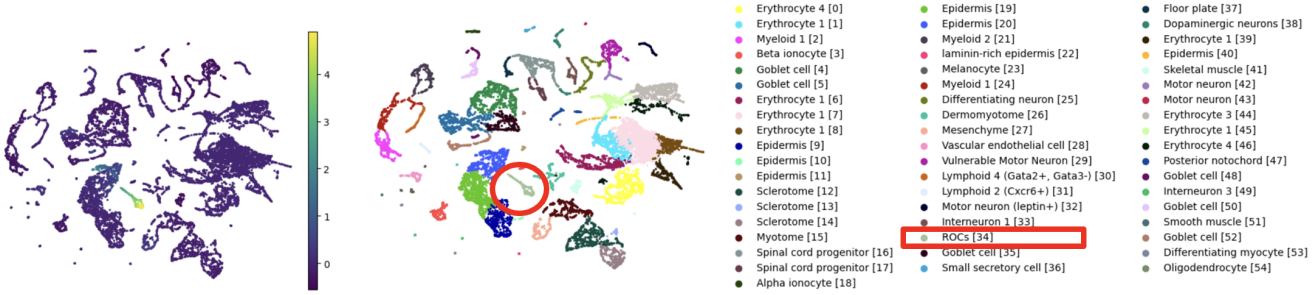

Figure 4. UMAP visualization of the location of the ROC genes according to [1] to the left and the Leiden clustering with MAGIC to the right. The red circle highlights the ROC region identified by the Leiden algorithm.

Leiden cluster 34 was the most ROC-like cluster and was also labeled as ROC by the Leiden algorithm (see Figure 4). This cluster occupies the same UMAP region as the ROC cluster in the original dataset. Logistic regression and Wilcoxon rank sum test both identified 18 overlapping genes with Supplementary Table 3. The methods agreed on 16 of these genes.

3.3.2 KNN Smoothing

The Leiden clustering identified 46 clusters when the KNN smoothing denoising algorithm was applied. The Silhouette score stayed roughly the same (0.296), while the Calinski-Harabasz score increased (2895.33). It was again better than for the K-means clustering (2797.36). The ARI score decreased to 0.267.

Leiden cluster 32 was labeled as ROC by the algorithm and was also identified to be the most ROC-like cluster. As in previous analyses, it appeared in the same UMAP region as the ROC gene cluster from the original study. Logistic regression identified 19 overlapping genes in the cluster with Supplementary Table 3, and Wilcoxon rank sum test identified 20 genes.

3.4 Effects of Batch Integration

3.4.1 Harmony

The Harmony algorithm resulted in 37 Leiden clusters. Again, the Silhouette score stayed about the same (0.278), while the Calinski-Harabasz score dropped noticeably (1409.51). This value was much lower than for K-means (2239.21), meaning that the Leiden clusters were less well separated and less compact. The ARI score did not change much (0.408).

The most ROC-like cluster was number 17 and it was also labeled as such (see Figure 5). As seen in the denoising results, this ROC cluster occupies the same UMAP region as the ROC-like genes in the original paper (see Figure 5). Logistic regression identified 15 overlapping genes with Supplementary Table 3, while Wilcoxon rank sum test found 9, all but one overlapping with those found with logistic regression.

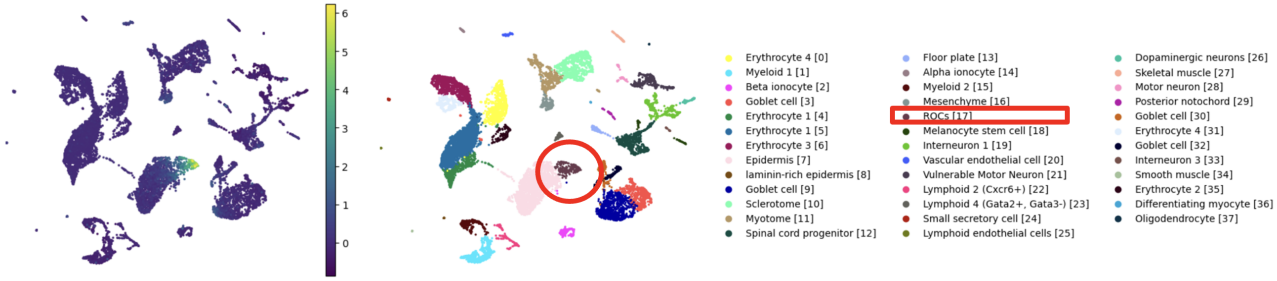

Figure 5. UMAP visualization of the location of the ROC genes according to [1] to the left and the Leiden clustering with Harmony to the right. The ROC region identified by the Leiden clustering is highlighted in red.

3.4.2 Combat

Applying the Combat algorithm resulted in 41 Leiden clusters. The Silhouette score dropped to 0.228, the Calinski-Harabasz score decreased to 1250.54, which was much lower than the score for K-means (1914.92). The ARI score also dropped to 0.226.

Leiden cluster number 15 was labeled as ROC, and it was also identified as the most ROC-like cluster. Again, this cluster occupies the same space on the UMAP as the ROC genes in the original paper. Wilcoxon rank sum test found 6 shared genes with Supplementary Table 3, while logistic regression found 13.

4 CONCLUSION

K-means and Leiden clustering were compared using Silhouette, Calinski-Harabasz and ARI scores, and K-means performed better. To stay consistent with the original paper, Leiden clustering was used for the rest of the analysis but still compared with K-means. Leiden cluster 16, labeled epidermis (no cluster was labeled ROC), appeared in the same UMAP region as the ROC cells from the paper, consistent with their findings that ROC cells are located in the epidermis. The Leiden metrics improved after denoising but worsened after batch integration, suggesting that denoising is more effective. In all cases, the ROC-like cells appeared in the same UMAP region as in the original paper, suggesting that this analysis successfully identified the ROC cells.

5 ACKNOWLEDGMENTS

The author used OpenAI's ChatGPT (2025) to refine the text for readability and assist with code development. All interpretations and conclusions are the author's own.

REFERENCES

[1] C. Aztekin, T. W. Hiscock, J. C. Marioni, J. B. Gurdon, B. D. Simons, and J. Jullien. Identification of a regeneration-organizing cell in the xenopus tail. *Science*, 364(6441):653–658, 2019. doi: 10.1126/science.aav9996.

[2] K. P. Murphy. *Probabilistic Machine Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, London, England, 2022. ISBN 9780262046824. URL https://probml.github.io/pml-book/book1.html.

[3] scikit-learn: adjusted_rand_score, 2025. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score. Accessed: 4 October 2025.

[4] scikit-learn: calinski_harabasz_score, 2025. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score. Accessed: 4 October 2025.

[5] Open Problems for Single Cell Analysis Consortium. Open problems, 2022. URL https://openproblems.bio/. Accessed: 4 October 2025.

[6] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P.-R. Loh, and S. Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0.