Project 1: Identify Regenerative Organizing Cell (ROC) in Frog Tail Skin

Chenjishi Lin (cl4618)

October 10, 2025

Abstract

Single-cell RNA-seq data from regenerating frog tails are analyzed to identify the Regenerative Organizing Cell (ROC) within the skin. After preprocessing the data, dimensionality reduction was performed using principal component analysis (PCA), and clustering was done with Leiden to yield a mean silhouette of 0.2825. For marker selection and gene analysis, clusters were then annotated through consensus markers from the Wilcoxon and t-test. The ROC corresponds to Leiden cluster C35 with wound-epidermis genes FN1, KRT16, and MMP11. Data denoising, kNN and iterative kNN, kept the same silhouette score, but changed the top-ranked markers. Batch integration methods, Harmony and BBKNN, reduced the separability to approximately 0.2399 and 0.2764, respectively, while preserving marker rankings. The ROC marker set did overlap with the Supplementary Table 3 gene list, supporting our identified cell and genes.

1 Introduction

A regenerative organizing cell is a specialized cell found in Xenopus tadpoles that plays a crucial role in tail regeneration [1]. ROCs would be in the epidermis during normal tail development, but relocate to the amputation site in tadpoles that can regenerate. They would form a wound epidermis at the injury site and release signals to progenitor cells to regrow. The goal of this analysis is to locate the ROC in scRNA-srq data of regenerating frog tails, and to denote the process for identifying. In addition, quantify how denoising and batch integration influence clustering and marker discovery.

2 Methods

2.1 Code Availability: GitHub repo

2.2 Clustering Steps

Raw single-cell RNA-seq data were loaded as an AnnData object and preprocessed as given. The count is initially normalized, then log-transformed per cell, filtered to remove uninformative features and low-complexity cells, and to focus on biological signals, 2300 highly variable genes were selected.

Dimensionality reduction was conducted using principal component analysis (PCA) with ARPACK solver. A kNN graph was constructed in PCA space using 15 neighbors and the first 30 principal components. This gives connectivities and distance graphs in adata object. Then, the two clustering algorithms with a resolution of 1.0, Louvain and Leiden, produced cluster labels. The quality of these clusters is assessed through a mean silhouette score computed in PCA space using NumPy. Furthermore, the agreement between clustering options is measured using three pairwise-comparison indices computed from the contingency table of labels [5]. Namely, Rand Index, Adjusted Rand Index, and Normalized Mutual Information that measures the mutual information between labeling, and normalized by entropy [3]. For visualization, a UMAP embedding was made as well.

2.3 Marker Selection and Gene Steps

Maker genes that define each cluster are identified by the Wilcoxon rank-sum test and a variance-overestimating t-test implemented in Scanpy [6]. Wilcoxon rank-sum compares one cluster against all

others by ranking all observations and asking whether the rank from the first group is higher or lower than that from the second group. For the t-test, it compares a difference in means to the amount of variability in the data. If the difference is relatively big compared to the noise, the test suggests that it's not by chance. The calculation is done by the difference in means divide by the estimated standard error.

The ranked results are then reshaped into long DataFrames, and each clusters are kept with the top-50 markers per test. Then, for annotation and comparisons, consensus labels were made by prioritizing the intersection of the two tests' top-k lists. Specifically, we take the intersection of the top-k gene lists as high-confidence markers, and if fewer than N markers remain, it is padded from the union by the lowest average rank across methods. The result is a small gene set, 3 in this case, which is the consensus label for that cluster.

2.4 Data Denoising & Batch Integration

To test the robustness, denoised copies were made by kNN smoothing and iterative kNN smoothing of the expression matrix X. This is done by row-normalizing the connectivity graph through the .obsp["connectivities"] and left multiplying X to get the averaged neighbor expression for single pass and iterative two pass. The labels from Leiden were reused, PCA recomputed for silhouette, and re-ran marker detection (Wilcoxon and t-test) so that any changes in the marker sets reflect the denoising of X rather than relabeling.

To integrate cells across DaysPostAmputation, the numeric column is converted to categorical labels D0-D3, and applied the methods Harmony and BBKNN. Harmony was implemented on the PCA embedding, then computed silhouette on the Harmony-integrated PCA and reused the Leiden labels. Harmony corrects time effects by adjusting cell embeddings in PCA space. It's modeling the embedding as a mixture of batch-specific offsets and iteratively doing the following: assigns cells to soft clusters, estimates batch-specific deviations for each clusters, and removes these deviations to produce an integrated embedding [2]. BBKNN builds the neighbor graph so that each cell draws a balanced number of neighbors from each time group. Instead of correcting the coordinates, it's rewiring the kNN graph to reduce batch-driven neighborhoods [4]. Since these methods did not modify X in this workflow, marker selection ranking is expected to remain the same.

3 Results

3.1 Clustering

Clustering on the PCA and kNN graph produced coherent partition with both Louvain and Leiden as shown in Figure 1. Quantitatively, Leiden showed higher separability than Louvain, with a mean silhouette of 0.2825 compared to 0.2483 for Louvain. This indicates that cells in the Leiden clusters

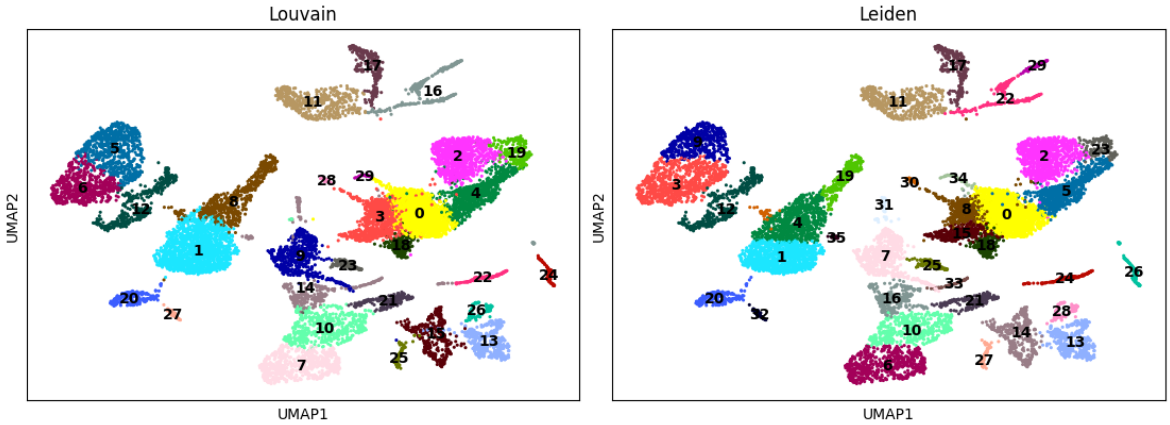


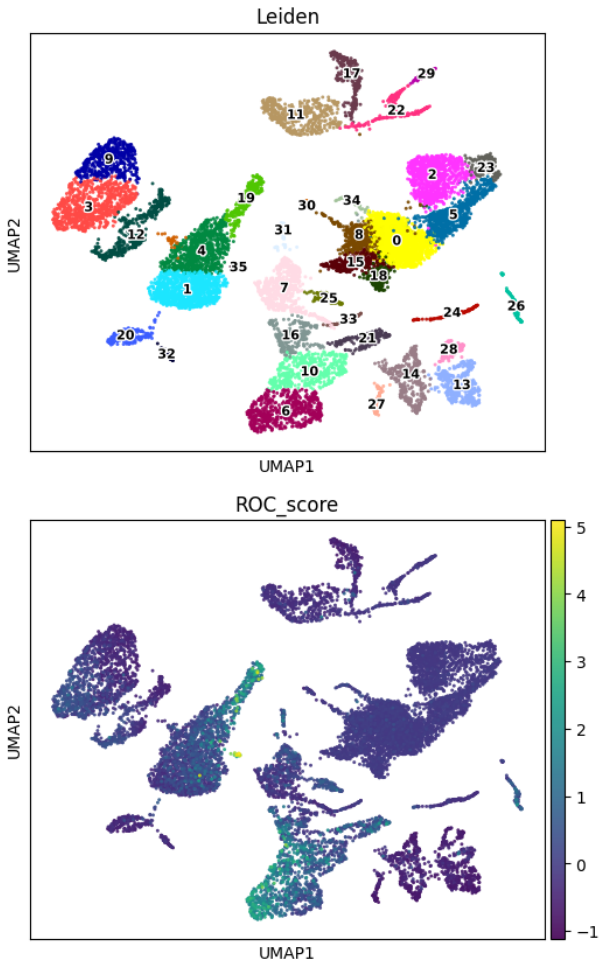
Figure 1: Clustering UMAPs colored by Louvain and Leiden.

are, on average, more compact and more distant from clusters in the 30-PC space. Despite this difference in the separability, the agreement between the two partitions was high. The Rand Index was 0.9784, which means nearly 98% of all the cell pairs received the same or different cluster decision. The Adjusted Rand Index was 0.8019, which also shows a strong agreement, implying only minor boundary differences (small splits/merges or a few relabeled cells), with the overall cluster structure preserved. The normalized mutual information got 0.9028, which means that the two labels share most of their information content. Together, these metrics support a stable global structure across methods. Figure 1 summarizes these results by displaying UMAPs colored by Leiden and Louvain, which illustrates a broad correspondence of regions, and the small number of boundary shifts is visually there.

3.2 Marker Selection

Marker selection was done for every cluster by contrasting one cluster against all others using two independent tests in Scanpy, i.e., the Wilcoxon rank-sum test and variance-inflated t-test. For each method, the top 50 genes per cluster are kept, as well as compared lists. The two test generally agreed with each other, where the per-cluster overlaps between them ranged from 32/50 to 50/50, with many clusters at or above 45/50. This strong agreement shows that the top markers are consistent across tests and reflect the same biological signal.

Then, the regenerative organizing cell (ROC) was identified within the skin manifold by examining the consensus markers and expression maps. Each cluster's top-k marker sets and the ROC's consensus



Cluster	Consensus (3 genes)
0	C0: Xelaev18002241m, ca2, Xelaev18045095m
1	C1: Xelaev18016080m, col1a2, col1a1
2	C2: Xelaev18045082m, Xelaev18047452m, Xelaev18047460m
3	C3: Xelaev18040854m, Xelaev18045649m, Xelaev18034081m
4	C4: lum, lum, col1a1
5	C5: Xelaev18047462m, ca2, hbg2
6	C6: col9a1, col9a2, col9a3
7	C7: efcab10, krt8, loc100490646
8	C8: Xelaev18002241m, ca2, MGC75752
9	C9: umod, Xelaev18012141m, Xelaev18015037m
10	C10: tnc, loc101733581, ptx3
11	C11: crp, Xelaev18018927m, unnamed
12	C12: itln1, loc100127564, loc100494945
13	C13: stmn2, snap25, stmn2
14	C14: elavl4, sncb, tuba1b
15	C15: Xelaev18002241m, Xelaev18022531m, ca2
16	C16: vim, lama4, nid2
17	C17: prss57, mpo, slurp1l
18	C18: Xelaev18002241m, ckm, ca2
19	C19: col14a1, apoc1, frem2
20	C20: Xelaev18040968m, atp6v1g3, cystm1
21	C21: hpd, vcan, ptx3
22	C22: arhgdib, lcp1, lcp1
23	C23: ccnb1, cdk1, Xelaev18047460m
24	C24: tnnt3, Xelaev18018009m, actc1
25	C25: loc100490646, loc100490646, shh
26	C26: prx, cdh5, icam5
27	C27: tal2, scg2, pkd2l1
28	C28: map1b, atp1b3, elavl4
29	C29: ctsl, loc100493999, gzmh
30	C30: Xelaev18045649m, fos, fos
31	C31: eln2, loc101731328, loc100497103
32	C32: atp6v1b1, slc26a2, atp6v0d2
33	C33: ednrb2, slc45a2, rab38
34	C34: ca2, col1a1, Xelaev18002241m
35	C35: fn1, krt16, mmp11

Figure 2: Clustering UMAPs colored by Louvain and Leiden.

set to the gene list from Supplementary Table 3. Using these consensus sets, ROC is identified as Leiden cluster C35. C35 is characterized by the wound-epidermis genes FN1, KRT16, and MMP11, which appeared as top-ranked markers in both tests. A module score is built from ROC markers to quantify the assignment, and C35 had the strongest score of 4.03. Figure 2 visualizes this module score on the UMAP and shows a tight spatial concentration of ROC signal in the C35 island, consistent with the qualitative impression from Leiden UMAP.

We also compared cluster-wise top-50 marker sets with Supplementary Table 3. Overlaps across the atlas were limited, as expected for tissue-specific panels collected under different conditions. Notably, the ROC consensus retained FN1, KRT16, and MMP11, supporting the interpretation of a wound-epidermis-like organizer within the skin compartment.

3.3 Data Denoising and Time Integration

We next evaluated the impact of denoising and time integration. The mean silhouette for single and two passes kNN remains unchanged, meaning the manifold geometry was stable under smoothing. Since denoising changes the expression matrix, it shifts how gene-level differences are weighted. For the ROC cluster, overlaps were roughly around 25/50 for single and 20/50 for iterative for Wilcoxon. For t-test, overlaps were 38/50 for single, and 34/50 for iterative. This highlights that smoothing changes gene rank the highest without moving cluster boundaries.

Finally, integrating cells across time modestly reduced separability, where the mean silhouette is 0.2399 with Harmony and 0.2764 with BBKNN. Because these methods act on the embedding rather than on the expression matrix, rankings were unchanged. The ROC assignment to C35 and its core markers remained stable after time integration.

A drop in silhouette after Harmony usually means better mixing across batches/time (less separation due to batch), while stable DE overlaps indicate the underlying X wasn't altered.

4 Conclusion

Overall, Leiden delivered a slightly better geometric separation than Louvain where both partitions are still highly consistent with RI 0.9784, ARI 0.8019, NMI 0.9038, indicating a stable global cell geography. Both methods of marker analysis return high within-cluster agreement and support a clear consensus-based annotation. Using these markers, we were able to identify the regenerative organizing cell as cluster C35 enriched for FN1, KRT16, and MMP11 genes. Denoising shifts marker rankings, while integration over Days Post Amputation improves cross-time alignment. These findings provide consistent evidence that C35 represents the ROC in the frog tail skin.

References

[1] C. Aztekin et al. "Identification of a regeneration-organizing cell in the *Xenopus* tail". In: *Science* 364 (2019), pp. 653–658. DOI: 10.1126/science.aav9996.

[2] Ilya et al. Korsunsky. "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature Methods* 16 (2019), pp. 1289–1296. DOI: 10.1038/s41592-019-0619-0.

[3] Fabian Pedregosa et al. $sklearn.metrics.normalized_mutual_info_score$. Documentation for scikit-learn 1.5 (stable release). https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html. scikit-learn developers. 2024.

[4] Krzysztof Polanski et al. "BBKNN: fast batch alignment of single cell transcriptomes". In: *Bioinformatics* 36.3 (2020), pp. 964–965. DOI: 10.1093/bioinformatics/btz625.

[5] William M. Rand, Nguyen X. Vinh, and Julian Epps. *Comparing Partitions Tutorial: Normalized Mutual Information and Related Indices*. Accessed: 2025-10-07. 2014. URL: http://www.comparingpartitions.info/?link=Tut14.

[6] Wolf, F. Alexander and Angerer, Philipp and Theis, Fabian J. *scanpy.tl.rank_genes_groups*. Scanpy Documentation (v1.10.0). https://scanpy.readthedocs.io/en/stable/generated/scanpy.tl.rank_genes_groups.html. Theis Lab, Helmholtz Zentrum München. 2024.