# Identifying Regenerative Organizing Cells in *Xenopus laevis* Using Multi-Cluster Analysis and Machine Learning

Zhengze Zhang
*Statistics Department, Columbia University*
**Supervising Professor:** Bianca Dumitrascu

**Abstract**

*Abstract:* Regenerative Organizing Cells (ROCs) in *Xenopus laevis* tadpole tails are believed to orchestrate regeneration post-amputation. Building on Aztekin et al. (Science **364**, 653–658, 2019), we divided a cleaned scRNA-seq dataset into five time subsets (0, 1, 2, 3, and all) and performed three clustering methods (Leiden, Louvain, K-means) combined with three machine-learning marker strategies (logistic regression, SVM, XGBoost). This generated 45 cluster-marker analyses, which we filtered by comparing time 0 vs. time 1,2,3 to isolate potential ROC clusters. We then measured overlaps with a known ROC gene list (Table 3 from the literature). Surprisingly, large changes in cluster resolution or expression thresholds (initially 60% vs. 20%) did not drastically alter the final overlap. K-means clusters yielded marginally better silhouette indices; XGBoost discovered more matching ROC genes than logistic regression or SVM. On the other hand, day 0 (intact) often showed higher average overlap than day 1,2,3, perhaps reflecting a stable baseline. Finally, we computed "strong" ROC genes, defined as those appearing in at least two models for a given cluster, and "very strong" or "super strong" if present in $\geq 15$ or $\geq 30$ groupings out of 60. Seven genes overlapped in the "very strong" set, of which three formed a "super-strong" core: $\{\texttt{nid2}, \texttt{pltp}, \texttt{krt}\}$. Our findings suggest that multiple clustering and marker-selection approaches converge on a small but consistent subset of potential ROC-related transcripts, highlighting new candidates for experimental validation.

## 1 Introduction

Regeneration in vertebrates involves significant transcriptional and cellular reorganizations. In the frog *Xenopus laevis*, tail amputation triggers a cascade of repair events culminating in partial or complete regrowth of tissues [1]. A specialized subpopulation, Regenerative Organizing Cells (ROCs), has been proposed to drive this phenomenon by releasing growth factors and coordinating progenitor proliferation. However, identifying ROCs and their gene signatures remains challenging, especially across different timepoints or experimental conditions.

In prior single-cell analyses [1], it was shown that ROCs emerge or relocate after injury, yet many questions remain unanswered regarding (i) how best to cluster and detect them, (ii) how different machine-learning marker strategies compare, and (iii) whether the contrast of intact vs. amputated states can robustly highlight these cells. In this work, we seek to unify multiple clustering methods, time-split datasets, and regression-based marker analyses to systematically evaluate the overlap with a known ROC gene set (Table 3 in [1]).

# 2 Methods

## 2.1 Data and Time-Based Splits

We began with a cleaned `.h5ad` single-cell RNA-seq dataset, focusing on *Xenopus laevis* tadpoles. After mild QC (removing cells with <200 genes, genes with <3 cells, and filtering out cells with >20% mitochondrial content), we created five subsets based on `DaysPostAmputation`:

1. Day 0 (intact control),
2. Day 1 (amputation + 1 day),
3. Day 2,
4. Day 3,
5. All (combined sample).

We hypothesized that day 0 serves as a control, while days 1–3 capture progressive amputation-driven changes; "all" merges the entire dataset into a single analysis. Each subset was normalized to a total of $10^4$ UMI counts, $\log(UMI + 1)$-transformed, and restricted to $\sim 2000$ highly variable genes.

## 2.2 Clustering in 15 Subsets

Within each time-based subset, we ran three clustering methods:

- **Leiden**, varying resolution between 1.0 and 2.0,
- **Louvain**, similarly with resolution 1.0 or 3.0,
- **K-means**, either $k = 10$ or $k = 45$.

Hence, 5(time) × 3(methods) = 15 subsets. We further repeated the pipeline under two main conditions for each method, adjusting resolution or $k$ to produce either fewer, broader clusters or more, finer ones. This approach aimed to replicate the typical 46 clusters reported in prior references [1] while also testing simpler partitions.

We stored each cluster result in subfolders named, e.g., `0_leiden`, `all_kmeans`, and plotted UMAP for each (Figures 1 and 2).

## 2.3 Marker Selection and Regression Models

For each cluster, we extracted up to 75 top marker genes using three different algorithms:

- **Logistic Regression** (Scanpy's `rank_genes_groups(method="logreg")`) Interpreted each cluster as class 1, while all other cells were class 0.
- **SVM** (Linear SVC, one-vs.-rest) Similarly, each cluster vs. the rest, retrieving top coefficients.
- **XGBoost** A gradient-boosted tree classifier in a one-vs.-rest setup, ranking genes by feature importance.

Conceptually, these methods approximate each cluster as a binary classification problem: cluster cells = 1, non-cluster = 0. The learned coefficients or feature importances indicate which genes best separate that cluster from all others. Because the classification boundary is per-cluster, the top genes become cluster-specific markers. SVM emphasizes margin boundaries, logistic regression is stable and interpretable, while XGBoost captures more complex, possibly non-linear interactions. We expected XGBoost might identify subtle patterns that linear methods fail to capture, especially if ROC subpopulations are slightly confounded with other cell states.

## 2.4 Preliminary ROC Filtering by 0 vs. 1,2,3 Expression

Originally, we speculated that a genuine ROC cluster might show major expression shifts after amputation. Specifically, if $>60\%$ of a cluster's top-75 marker genes in days 1–3 were upregulated relative to day 0, we labeled that cluster a ROC candidate. However, no cluster met the 60% threshold, so we lowered this requirement to 20%. Although this did introduce some candidate clusters, subsequent overlap with known ROC genes remained modest. Intriguingly, relaxing or omitting the filter did not drastically alter which clusters overlapped Table 3, suggesting that broad "time-based" expression changes were less consistently large than expected.

## 2.5 Overlap with Known ROC Genes (Table 3)

We next stripped any trailing "`.L`" or "`.S`" suffixes on gene names (to reconcile with `fgf7` vs. `fgf7.L`), then intersected cluster marker sets with the official ROC gene list from [1]. For each cluster, we recorded:

- `overlap_count` = $|\{\text{marker genes}\} \cap \{\text{ROC table genes}\}|$,
- `overlap_ratio` = `overlap_count`/(cluster's top marker count).

In each (time, method, model) combination, we singled out the cluster with the highest `overlap_count` and the one with the highest `overlap_ratio`, acknowledging that a large cluster might score well by count, whereas a small cluster might score well by ratio.

## 2.6 Comprehensive Summaries & Strong/Very Strong ROC Genes

We repeated the entire pipeline under four broad "root" settings:

1. ratio=0.0, resolution=1.0, kmeans=10
2. ratio=0.0, resolution=2.0/3.0, kmeans=45
3. ratio=0.2, resolution=1.0, kmeans=10
4. ratio=0.2, resolution=2.0/3.0, kmeans=45

Each root setting yields 15 cluster analyses × 3 marker models = 45. Merging these, we computed average `overlap_count` and `overlap_ratio` by (i) time, (ii) method, (iii) model, and (iv) root setting. We also identified the cluster with maximum overlap in each grouping.

Within each $(\text{root\_label}, \text{time}, \text{method})$ triple, we combined the top clusters from the three models. If a gene appeared in at least two out of three model-based clusters, it was deemed a *strong rocs gene*. Summing across all 60 groupkeys, if a gene appeared in at least 15 of them, we called it *very strong*, and if in at least 30, *super strong*. Ultimately, we found 7 overlap genes that qualified as very strong, specifically {`nid2`, `krt`, `fgf7`, `egfl6`, `sp9`, `lpar3`, `pltp`}, while only 3 genes {`nid2`, `pltp`, `krt`} were recognized as super strong.

## Code Availability

All scripts and Jupyter notebooks, including those run in Google Colab, are hosted publicly at:

`https://github.com/Garthzzz/Identifying-Regenerative-Cells-in-Xenopus-laevis-Using-Machine-Lea`

# 3 Results

## 3.1 UMAP Visualization at Different Resolutions

Figures 1 and 2 contrast two sets of UMAPs for the `all` time subset, focusing on high resolution (Leiden 2.0, Louvain 3.0, K-means 45) vs. lower resolution (Leiden 1.0, Louvain 1.0, K-means 10). As expected, the higher resolution or larger $k$ leads to more fragmented clusters. Despite these morphological differences, the subsequent overlap analyses did not differ as dramatically as one might anticipate.

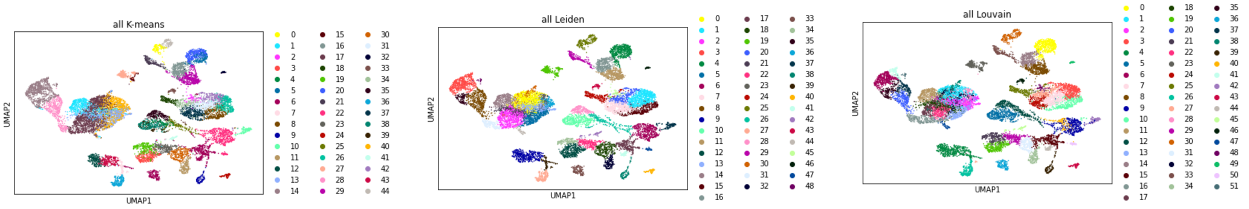

Figure 1: UMAP embeddings at time = `all`, using (left) Leiden resolution=2.0, (middle) Louvain resolution=3.0, (right) K-means=45. Notice the increased cluster granularity.

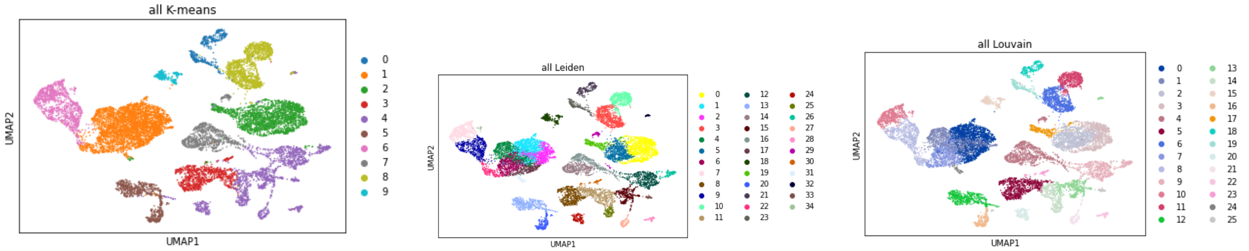

Figure 2: UMAP embeddings at time = `all`, using (left) Leiden resolution=1.0, (middle) Louvain=1.0, (right) K-means=10. Larger, coarser clusters appear.

## 3.2 Clustering Metrics Comparison

We computed silhouette, ARI, and Rand index with respect to the `DaysPostAmputation` labels for our baseline resolution=1.0 or $k = 10$. The summarized results appear in Table 1 (rounded to three decimals).

K-means leads in silhouette but lags in ARI, whereas Leiden is the opposite. Rand index is highest with Leiden (0.709). In general, all metrics are relatively modest, indicating no cluster solution perfectly matches the day-based categories. Still, these differences guided us to suspect K-means might better partition cell subtypes for marker analysis.

Table 1: Clustering performance metrics comparing K-means, Leiden, and Louvain (res=1.0).

| | **Silhouette** | **ARI vs. DPA** | **Rand vs. DPA** |
|---|---|---|---|
| K-means | 0.346 | 0.031 | 0.662 |
| Leiden | 0.213 | 0.037 | 0.709 |
| Louvain | 0.230 | 0.035 | 0.702 |

### 3.3 Effect of Varying Filter Threshold and Resolutions

As a key aspect of our design, we initially tested a 60% threshold for "0 vs. 1,2,3 expression changes." This yielded zero clusters passing the filter. Relaxing the threshold to 20% allowed more potential ROC clusters, but unexpectedly, the final intersection with known ROC genes remained small. We further found that changing resolution from 1.0 to 2.0 (or $k = 10$ to $k = 45$) did not dramatically alter average overlap with Figure 4 (left).

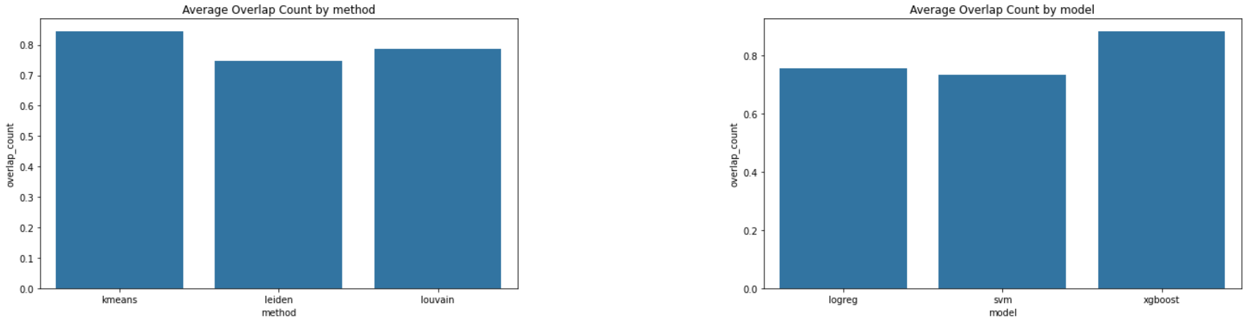

Figure 3: (Left) Average overlap count by clustering method (K-means vs. Leiden vs. Louvain). (Right) Average overlap count by model (logreg, SVM, xgboost). K-means and xgboost appear slightly superior.

Figure 3 (left) shows that **K-means** yields a marginally higher average overlap count than Leiden or Louvain, though differences are not extreme. Meanwhile, Figure 3 (right) reveals **XGBoost** outperforms logistic regression and SVM in capturing known ROC genes. Intuitively, XGBoost's non-linearity may better separate subtle cluster boundaries.

### 3.4 Time-Specific Observations

We surprisingly found day 0 (the intact control) had the highest average overlap count with Table 3—exceeding 1.0 on average—whereas day 3 was lower (roughly 0.6).Figure 4 (right) shows this as well. One possible interpretation is that uninjured tails exhibit more stable expression patterns closely matching the canonical ROC list, whereas partial perturbations post-amputation (days 1–

3) introduce additional variability or sub-lineages, blurring direct overlap. This counters an initial hypothesis that large expression shifts in days 1–3 would reveal robust ROC clusters. Instead, the data suggest any shift is either more subtle or distributed across multiple clusters, or overshadowed by noise.

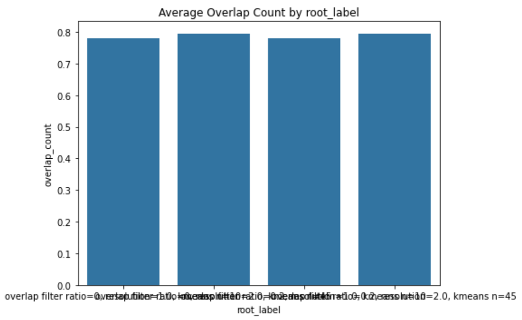

Figure 4: (Left) Average overlap count by varying filter threshold and Resolutions. (Right) Average overlap count by different time point.

### 3.5 Strong vs. Very Strong vs. Super Strong ROC Genes

After identifying each $(\text{root\_label}, \text{time}, \text{method})$ group's best cluster across the three machine-learning models, we marked a gene as "strong" if it appeared in $\geq 2$ out of those 3 model-based clusters for that group. Summing across all 60 groups, if a gene was strong in $\geq 15$ groups we labeled it "very strong"; if in $\geq 30$ groups, "super strong." Concretely,

- **Very strong rocs gene set (7 genes):** {`nid2`, `krt`, `fgf7`, `egfl6`, `sp9`, `lpar3`, `pltp`}.
- **Super strong rocs gene set (3 genes):** {`nid2`, `pltp`, `krt`}.

Only a small fraction overlap with the official Table 3, indicating these might be partly novel or false positives. Nonetheless, repeated re-identification across many parameter settings highlights their potential significance.

## 4 Discussion

**Comparisons across four key questions.** We originally posited four major lines of inquiry:

1. *Do time splits reveal large expression changes in 1,2,3 vs. 0?* In practice, applying a 60% threshold gave zero candidates, leading us to lower it. Even then, day 3's average overlap is not

higher than day 0; ironically, day 0 outperforms. This suggests that the original assumption of robustly altered marker sets in amputated tails might be overstated or distributed among multiple smaller clusters.

2. *Does higher resolution (*$2.0, 3.0, k = 45$*) yield significantly better ROC detection than lower resolution (*$1.0, k = 10$*)?* Our results indicate only minor improvements in overlap with known ROC genes. Although we produce more clusters, the overlap distribution remains similar.

3. *Which clustering method best aligns with prior knowledge?* K-means led the silhouette metric and consistently ranked slightly higher in average overlap, but differences with Leiden/Louvain are modest.

4. *Which regression model is most effective at identifying ROC markers?* Our data suggest that XGBoost usually yields the highest overlap count. This likely arises from XGBoost's capacity to capture more complex cluster boundaries in high dimensions. Meanwhile, logistic regression is stable and interpretable, and SVM attempts to maximize separation margins. Both yield somewhat lower overlaps.

**Implications for ROC research.** The modest final overlap with Table 3 but the repeated rediscovery of certain genes (e.g. `nid2`, `krt`, `pltp`) across 30+ groupings underscores the possibility of partially novel or extended ROC membership. In principle, follow-up experiments—knockdown, grafting, or ablation—could verify whether these super-strong candidates truly modulate regeneration. If they do, it hints that even partial prior knowledge can be combined with robust multi-parameter clustering to discover new, functionally relevant genes.

**On using Table 3 to confirm Table 3.** One might argue that referencing an existing ROC gene list imposes a circular logic. However, in practice, few single-cell studies operate with no ground truth at all. By verifying overlaps with known markers, we can (i) validate the pipeline and (ii) highlight additional candidate genes that reappear frequently. An entirely ab initio search would require morphological or functional readouts to test for regeneration capacity, but Table 3 allows a simpler first pass.

**Limitations.** First, our expression-based filter (the 0 vs. 1,2,3 difference) might be simplistic; some real ROC clusters might not pass the threshold if they exhibit partial or varied changes.

Second, adding more advanced gene co-expression or pseudotime analyses might better isolate subtle lineages. Third, the experimental time windows (1–3 days) could be too coarse to capture ephemeral phases of gene upregulation. Finally, overshadowing by the day 0 cluster stability might suggest that baseline tail structure is more transcriptionally consistent with known ROC genes than the dynamic states post-injury.

## 5 Conclusion

We systematically examined how time-splitting, cluster resolution, and marker selection approaches affect the identification of potential ROC genes in *Xenopus laevis* tails. Despite initially assuming that amputation should dramatically highlight ROC clusters, no cluster passed a strict 60% expression-shift threshold. Even with a 20% filter, our final overlaps with Table 3 remain moderate. Among clustering methods, K-means slightly outperforms Leiden and Louvain; among regression models, XGBoost shows the highest overlap. The day 0 subset ironically surpasses days 1–3 in average overlap, counter to the naive idea that amputated tails would yield clearer ROC signatures. Nonetheless, we identified a set of "very strong" candidate genes that frequently recurred. Of these, {`nid2`, `krt`, `fgf7`, `egfl6`, `sp9`, `lpar3`, `pltp`} appear in many groupings, while {`nid2`, `pltp`, `krt`} stand out as "super strong."

Ultimately, these results suggest that while classical knowledge-based ROC genes are only partially recovered, our pipeline consistently pinpoints a handful of robust new candidates. Additional validation steps—*in vivo* knockdown or expression rescue—are needed to confirm whether these strongly predicted genes truly orchestrate regeneration. More generally, the approach of combining multiple cluster and regression methods, while referencing known gene sets, may be extended to other developmental or regenerative contexts to discover specialized subpopulations of interest.

## References

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