Reopening the Case: Bottom-Up Validation of Renal scRNA-seq Cluster Annotations

Allison Wivagg[13], Amy Ji[2], Qinglin Kong[2], Vania Halim[13]

[1]School of Computer Science, [2]Mellon College of Science, [3]Ray and Stephanie Lane Computational Biology Department
Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

Although the number of single-cell RNA sequencing (scRNA-seq) studies continues to grow each year, over 51% are not reproducible from publicly available data[7]. A key step in scRNA-seq analysis is the process of cluster annotation, where cells that are grouped into communities based on their gene expression profiles are mapped to a cell type. This process is highly subjective and researcher-dependent. No single standardized workflow exists, nor would it be appropriate to insist on one due to the complexity of biological datasets and the utility of scRNA-seq for discovery of novel clusters. In this work we apply a bottom-up approach to validate the Doke et al. [6] discovery of a novel profibrotic Proximal Tubule (PT) cell type in mice with simulated kidney inflammation without biological domain expertise. We analyzed and verified key pre-processing steps and performed unbiased exploration of differentially expressed genes in each cluster, targeted marker gene validation, and the use of reference databases to annotate each cluster. Although we were able to loosely replicate the main discovery of the novel cluster, the limitations of the methods we explored highlights the need for transparency when making key analytical decisions in scRNA-seq studies.

Introduction

Renal fibrosis is a hallmark of chronic kidney disease (CKD) and a common representation of end-stage renal failure. Over the last 30 years, the prevalence of CKD in adults over the age of 20 has increased by over 10%. With now an estimated 800 million people worldwide affected by it, CKD is one of the fastest-growing causes of death, yet treatment options remain limited. Understanding the mechanisms driving the progression of renal fibrosis is critical for developing new therapeutic approaches for chronic kidney disease.

Fibrosis is characterized by a buildup of scar tissue within the kidney. It arises from chronic inflammation triggered by prolonged immune responses to infection or tissue injury. Immune responses typically operate through a signaling cascade, resulting in a systemic inflammatory response. This nonresolving inflammation activates fibroblasts [1,8,12], leading to excessive activation of collagen production, which replaces functional kidney cells with permanent scar tissue, eventually causing renal fibrosis[3].

Despite the clear implication of inflammation and the presence of immune cells in renal fibrosis, the specific types of immune cells and the mechanisms of the pro-fibrotic cascade remain poorly understood. To address this gap, Doke et al. (2022) investigated the role of basophils in the activation of renal fibrosis. In mice, basophils secrete IL-6, a type of cytokine that promotes inflammatory responses by recruiting helper T cells. These properties suggest that basophils are a driver of the pro-fibrotic cascade.

To systematically characterize cellular and molecular changes and develop a single-cell atlas associated with kidney fibrosis, Doke et al. performed sham and unilateral ureter obstruction (UUO) surgeries on kidneys in wild-type mice. Later, they homogenized samples and performed single cell RNA-sequencing (scRNA-seq). During data analysis, the group isolated the proximal tubule (PT), the most abundant epithelial cell types in the kidney[5], and performed sub-clustering in silico. The analysis revealed a new subgroup of profibrotic PT cells characterized by *Pdgfb*, a marker of fibroblast activation. Subsequently, in a variety of wet-lab experiments, Doke et al. confirmed basophil recruitment by CXCL1 secretion by profibrotic PT cells.

The accuracy of these findings hinges on the rigor of cluster annotation, which is the process of assigning cell types to clusters by identifying the gene expression profile of the cells within them. Thus, to verify the cluster annotation of Doke et al.'s paper, we reproduced the key in silico analysis using the published dataset and workflow and evaluated whether profibrotic PT is a reproducible biological signal or an artifact of research-dependent choices. In particular, we focus on cluster annotation, which assigns biological identities to transcriptionally defined clusters, and cell trajectory analysis, which infers how proximal tubule cells may transition from healthy to profibrotic states during renal fibrosis. We then compared our results with the key findings in the paper.

While standard scRNA-seq workflows often rely on expert-guided heuristics, the robustness of these choices is best tested by their ability to resolve rare, transitional cell states. This study employs a first-principles approach to determine if a bottom-up framework can independently reproduce the discovery of the novel profibrotic proximal tubule (PT) cluster, a critical state in kidney injury that is often difficult to distinguish from general technical noise.

Methods

Preprocessing

Mouse kidney scRNA-seq data from the UUO model generated by Doke et al. were obtained from Gene Expression Omnibus under the acquisition number GSE182256. The dataset includes six sham controls and two UUO samples[6]. The authors published count object provided as an RDS file, cell-level metadata, and UMAP coordinates. The RDS file was loaded into R and used to create a new Seurat object with the `CreateSeuratObject` function. The corresponding metadata file was then loaded and matched to the Seurat object by cell barcode. Metadata fields for the identity of the sample and the annotation of the published cell were retained as `orig.ident` and `paper_cluster`, respectively.

Before undergoing any downstream analysis, raw scRNA-seq counts data must undergo significant pre-processing. Single-cell RNA sequencing (scRNA-seq) maps each RNA transcript to its cell of origin using droplet-based microfluidics technology. Ideally, a single cell and a single bead are encapsulated in a single oil droplet. The bead is a resin particle with millions of DNA fragments covalently attached to its surface. It carries a cell barcode identifying the cell and UMI barcodes identifying individual transcripts from each cell. In reality, however, a droplet may contain more than one cell, no cells, or dying cells.

scRNA-seq data is provided as a gene expression counts matrix where each row *i* is a cell, and each column *j* corresponds to a gene. The matrix values represent the number of RNA transcripts for gene *j* detected in cell *i*. Violin plots visualize the probability density of data points for a certain category. The width of the violin indicates where the majority of transcriptomes cluster, while the elongated tails highlight outliers that may represent technical noise or doublets (Fig. 3a). By generating violin plots for each metric, we can identify or verify numerical cutoffs for isolating high-quality biological signals from artifacts.

Doublets are characterized by abnormally high total RNA counts. On the other hand, a droplet containing no cells will have an abnormally low RNA count for each gene. Stressed or dying cells will display elevated mitochondrial gene expression. During the pre-processing steps, these low-quality cells are filtered out to preserve the fidelity of data analysis results (Fig. 1a). Then, the data is mean-centered and scaled such that the mean expression of every gene is 0 and the variance is one, using Seurat's `NormalizeData` and `ScaleData` functions. This is a critical step for downstream dimensionality reduction, because it allows the algorithm to focus on patterns of gene expression rather than the magnitude of raw counts.

Dimensionality Reduction

Cells in an scRNA-seq counts matrix initially exist in an n-dimensional space, where n is the number of genes observed in the entire dataset (typically 20,000 to 30,000). Analyzing the data in such high dimensionality is computationally intensive and may lead to meaningful biological signals being obscured by stochastic noise.

As a first step in reducing the number of dimensions to a more manageable size, only the top 2000 most highly variable genes are retained as a form of feature selection using Seurat's `FindVariableFeatures` function[17]. By identifying genes that exhibit significantly higher variance across the cell population than would be expected based on their mean expression levels, we can focus the analysis on the most informative biological signals.

Dimensionality Reduction Problem

Input: A centered and scaled gene expression matrix $X \in \mathbb{R}^{n \times d}$ and the number of desired components k.

Output: A reduced-dimension matrix $X_{proj} \in \mathbb{R}^{n \times k}$ that retains most of the variance in the original dataset.

Solution: Project X onto the top k primary axes of variance using the PCA algorithm. Generate an elbow plot to determine k, the number of PCs beyond which rate of variance diminishes.

Principal Components Analysis (PCA):
1. Construct the covariance matrix S for each pair of genes
2. Perform eigenvalue decomposition on S to identify the eigenvectors or principal components (PCs) and their eigenvalues
3. Sort eigenvectors by their corresponding eigenvalues and select the top k eigenvectors
4. Project the original data onto the top k principal components/eigenvectors

---

PCA is performed using Seurat's `RunPCA` function[17] to generate a low-dimensional representation that retains essential biological. Figure 1b illustrates the identification of PCs and the subsequent projection onto the top 2 PCs. The number of principal components (*k*) to retain is a key analytical decision. Each successive PC captures a diminishing percentage of the total dataset variance. While a heuristic of *k* =30 is frequently employed for single-cell transcriptomics[17], the optimal *k* may instead be determined via an elbow plot (Figure 3c). An elbow plot visualizes the variance explained by each PC (or its

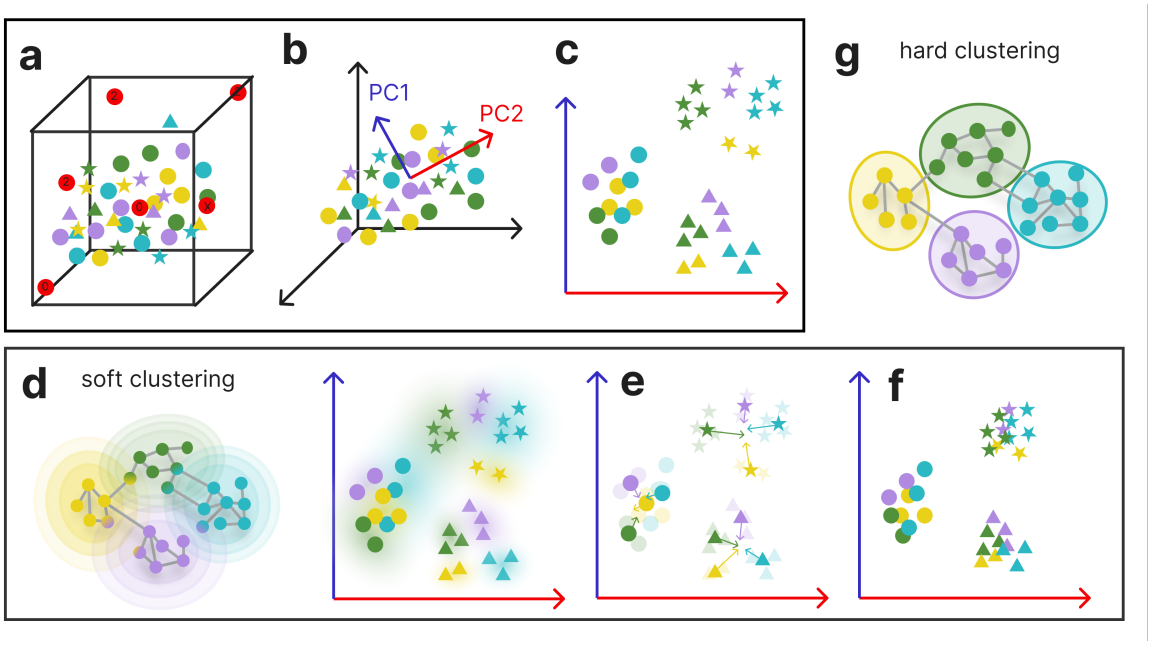

Figure 1: **Schematic of the computational pipeline for data integration and manifold learning. (a)**Low-quality barcodes are filtered from the high-dimensional expression matrix. The raw data contains biological variance (shapes) confounded by technical batch effects (colors). **(b)** PCA is applied to identify the top k principal axes of variance, reducing the feature space while preserving global structure. **(c)** Cells are projected onto the k-PC space. At this point variance due to batch effects dominates the position of each sample cell. **(d)** Harmony integration calculates probabilistic membership weights, assigning each cell probabilistically to multiple potential clusters simultaneously (soft clustering) to account for local uncertainty. **(e)** A correction vector is calculated for each cell based on its soft cluster assignments. A penalty is applied to clusters with low batch diversity, iteratively tugging cell coordinates to align the Sham and UUO batches. **(f)** The integrated PCA space represents a harmonized manifold where clusters are defined by biological cell states rather than technical batches. **(g)** In contrast to soft clustering in (d), hard clustering deterministically assigns each data point to a cluster.

corresponding eigenvalue) in descending order. The elbow represents a distinct inflection point where the magnitude of explained variance plateaus. Beyond this threshold, additional PCs may represent stochastic noise rather than biologically meaningful variation.

Removal of Batch Effects

Even after normalization, a dataset may be confounded by batch effects, which are systemic technical variations introduced by differences in sample handling, sequencing runs, or individual mouse physiology across different batches. In studies comparing healthy and diseased states, such as the Doke et al. paper, these artifacts can cause cells to cluster by their sample of origin rather than cell type.

We utilized Harmony [10] to integrate cells from all 8 batches, including cells from 6 sham and 2 UUO mouse kidneys. Soft k-means clustering is employed to identify similar cell populations or clusters across batches in the PCA space (Fig 1d). Then, a correction factor is calculated to pull batch-specific cluster centers toward one another (Fig 1e). The soft approach allows the algorithm to calculate these correction factors without over-correcting.

Quantifying Batch Integration Problem

Input: A Harmony-integrated matrix $X \in \mathbb{R}^{n \times k}$ and cluster assignments C for n cells belonging to B discrete batches.

Output: An entropy score $H(c)$ for each cluster $c \in C$, representing the degree of inter-batch mixing.

Solution: Compute Shannon Entropy for each cluster to determine if it is well-mixed or batch-biased.

Shannon Entropy:

1. For a specific cluster c, identify the set of cells n_c and their respective batch origins.
2. Calculate the probability p_i of a cell in cluster c belonging to batch i (where $i \in \{1, \ldots, B\}$):

$$p_i = \frac{\text{cells from batch } i \text{ in cluster } c}{\text{total cells in cluster } c}$$

3. Compute the Shannon Entropy $H(c)$ for the cluster:

$$H(c) = -\sum_{i=1}^{B} p_i \log_b(p_i)$$

Following the removal of technical batch effects, clusters must be validated for multi-sample representation. We utilize

Shannon Entropy to ensure that the resulting manifold is based on shared biological states rather than technical artifacts[4].

Unsupervised Clustering

Once batch effects and technical variance have been minimized, we can cluster cells based on the similarity of their global gene expression profiles. The goal of unsupervised clustering is to partition the cells in the PCA space into distinct biological groupings, without prior knowledge of cell-type labels.

First, a Shared Nearest Neighbor (SNN) graph is constructed to represent the underlying topology of the dataset. In this framework, cells are represented as nodes, and the algorithm initially identifies the k-nearest neighbors (kNN) for each cell in the low-dimensional batch-corrected PCA space. The distance between two cells is represented by the Euclidean distance of gene expression between cell coordinates.

Problem: Defining Similarity of Cells

Input: A set of k-nearest neighbor sets $N = \{N_1, N_2, \ldots, N_n\}$ for n cells, derived from Euclidean distances in PCA space.

Output: A Shared Nearest Neighbor (SNN) graph G where edge weights w_{ij} represent topological similarity.

Solution: Transform spatial proximity into neighborhood overlap by calculating the Jaccard coefficient for every pair of cells that share at least one neighbor.

The Jaccard Similarity Coefficient:

For any two cells i and j, let N_i and N_j represent the sets of their respective k-nearest neighbors. The edge weight w_{ij} is defined as:

$$J(N_i, N_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

Logic and Noise Suppression:

1. **Intersection ($|N_i \cap N_j|$):** Counts the number of specific neighbors shared by both cells.
2. **Union ($|N_i \cup N_j|$):** The total number of unique neighbors across both sets.
3. **Weighting:** An edge weight of 1 indicates identical neighborhoods, while 0 indicates no shared neighbors.

The SNN is constructed by connecting two cells with an edge if they share a significant number of these neighbors in common (Fig. 2a). The strength of the connections is determined by the Jaccard similarity coefficient. Weighting edges based on shared context rather than simple Euclidean distance suppresses technical noise and ensures that connections are only maintained between cells residing in high-density biological manifolds.

Problem: Finding Modular Neighborhoods

Input: A weighted adjacency graph $G = (V, E)$, where V represents n cells and E represents the edge weights from the fuzzy simplicial set.

Output: A partition of cells into c communities such that the network modularity Q is maximized.

Solution: Utilize the Louvain algorithm: an iterative, two-phase heuristic that greedily moves nodes into communities to maximize local modularity gain, followed by the aggregation of nodes into supernodes to resolve global structure.

The Modularity Objective Function:

The algorithm seeks to maximize Q, defined as the difference between the actual number of edges within communities and the expected number of edges in a random null model:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Variable Definitions:

- A_{ij}: The weight of the edge between cell i and cell j.
- k_i, k_j: The sum of weights of the edges attached to nodes i and j (degree).
- m: The total sum of all edge weights in the graph ($m = \frac{1}{2} \sum A_{ij}$).
- $\delta(c_i, c_j)$: Kronecker delta (1 if cells i, j are in the same community, 0 otherwise).

The Louvain algorithm[2] begins in a local moving phase , where each node is assigned to its own community. Nodes are then iteratively reassigned to neighboring communities that yield the maximum increase in modularity (Fig 2b). This local optimization is repeated until the graph structure stabilizes and no further node movements result in a modularity gain. Then the nodes in each cluster are aggregated into a supernode (Fig 2c), and the process repeats until the global structure converges.

Unlike the k-means algorithm[13] which is forced to discover a user-specified k clusters, Louvain is able to learn the groupings inherent in the data. This flexibility makes the clustering highly robust and allows for the discovery of novel cell types such as the profibrotic PT subcluster.

Visualizing clusters through UMAP

PCA typically reduces dimensionality from 20,000 to 30 principal components. However, this level of dimensionality is still too complex to be manually interpreted and visualized. In order to visualize the results of clustering, we must project the data onto a two-dimensions. While PCA is excellent for identifying the primary axes of variance, it is a linear technique that fails to retain the non-linear relationships present in biological systems.

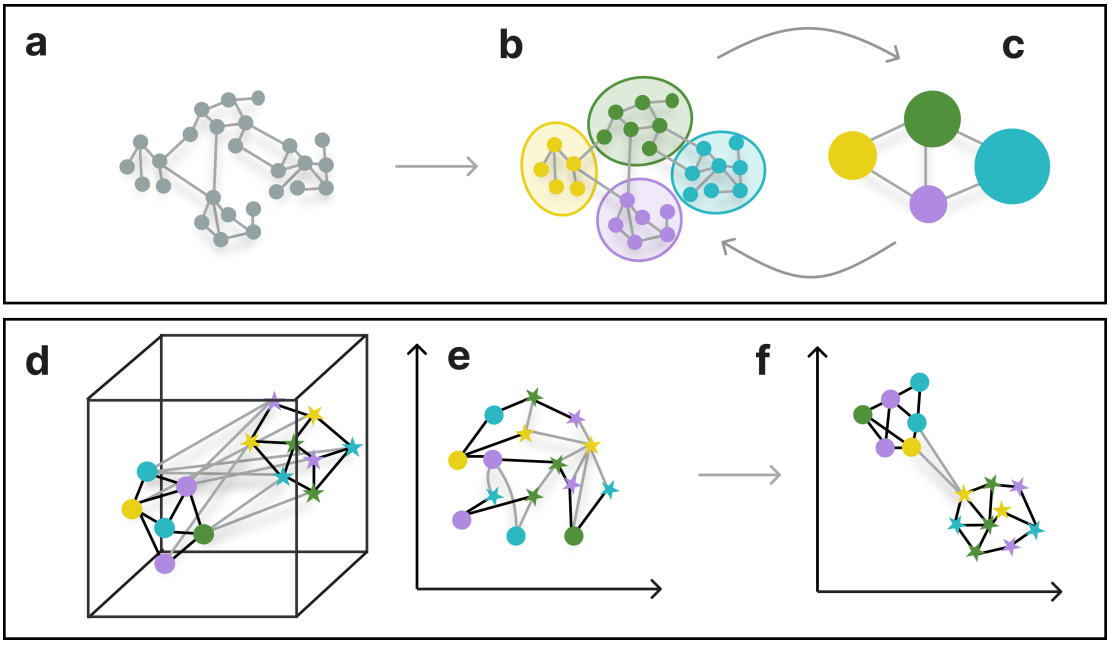

Figure 2: **Graph-based clustering and non-linear manifold embedding. (a)** In a Shared Nearest Neighbor (SNN) graph cells are represented as nodes in the k-dimensional PCA space, with edge weights determined by the Jaccard similarity of their local neighborhoods. This graph serves as the topological input for community detection through the Louvain algorithm. **(b)** In the local move nodes phase of Louvain, nodes are initially treated as individual communities and iteratively reassigned to neighboring communities to maximize local modularity gain (Q). Nodes are colored by cluster identity. **(c)** Communities are aggregated into supernodes and the optimization repeats to resolve hierarchical global structure. **(d)** In the PC-reduced space, similar cell types are connected by strong edges, and connected loosely (gray) to a cluster of different cell types. **(e)** An initial 2D projection where edge weights do not yet reflect the high-dimensional topology, resulting in high cross-entropy between the 2D layout and the reference manifold. **(f)** Optimized UMAP Embedding: The final 2D coordinates are learned by minimizing the cross-entropy. The resulting layout successfully reproduces the high-dimensional connectivity, where strong internal cluster edges (black) and loose inter-cluster bridges (gray) are preserved for biological interpretation.

To visualize the results for further interpretation, we employ the Uniform Manifold Approximation and Projection (UMAP) algorithm as presented by McInnes et al. [15]. UMAP is a non-linear manifold learning technique that operates on the assumption that the data is distributed along a low-dimensional manifold that can be approximated as a fuzzy graph.

Problem: Non-Linear 2D Cluster Embedding

Input: A high-dimensional fuzzy simplicial set with edge weights $\mu_{ij} \in [0, 1]$ representing the probability that an edge exists between cells i and j.

Output: A low-dimensional (2D) embedding $Y = \{y_1, y_2, \ldots, y_n\}$ that preserves the global and local topology of the dataset.

Solution: Optimize the positions of points in 2D space by minimizing the fuzzy set cross-entropy between the high-dimensional weights μ_{ij} and low-dimensional weights ν_{ij}.

The Cross-Entropy Objective Function:

UMAP utilizes a force-directed layout optimization. For all pairs of points (i, j), the algorithm minimizes:

$$C = \sum_{i \neq j} \left[\underbrace{\mu_{ij} \log\left(\frac{\mu_{ij}}{\nu_{ij}}\right)}_{\text{Attractive Force}} + \underbrace{(1-\mu_{ij}) \log\left(\frac{1-\mu_{ij}}{1-\nu_{ij}}\right)}_{\text{Repulsive Force}} \right]$$

The Mathematical Mechanism:

1. **High-D Weights (μ_{ij}):** Derived from the UMAP graph (SNN-like structure).
2. **Low-D Weights (ν_{ij}):** Usually modeled as a heavy-tailed distribution, $\nu_{ij} \approx (1 + a\|y_i - y_j\|^{2b})^{-1}$.
3. **Optimization:** The first term (Attractive) pulls connected points together. The second term (Repulsive) pushes disconnected points apart to prevent "clumping."

Similar to the SNN approach, UMAP first builds a weighted graph representation of the high-dimensional data, focusing on the local neighborhood of each cell (Fig 2d). Then, the algorithm finds a 2D layout that minimizes the cross-entropy of edge weights across all pairs of points (Fig 2f).

The cross-entropy model measures the disagreement between the 2D representation and the higher dimensional space. A cost function is penalized when high-dimensional neighbors are placed far apart in 2D space, or when distant points in the higher-dimensional space are placed too close together.

The resulting UMAP plot serves as the primary visual interface for the dataset. However, UMAP should not be used in isolation to identify clusters; it is highly sensitive to hyperparameter choice and must always be paired with a dedicated clustering algorithm.

Differentially Expressed Gene (DEG) Analysis

Now that we have generated a visual representation of the clustering analysis, we can move on to map each grouping to a biological function or cell type. A key step in the paper is the exclusion of proximal tubule (PT) cells for further subclustering to reveal the profibrotic PT cell cluster, which expresses markers for inflammation but also expresses genes related to fibroblast activation. Understanding how this cell type impacts kidney fibrosis is a main focus of the paper and the foundation of all other downstream studies.

Cluster annotation is performed using a multi-pronged approach that combines statistical discovery, biological validation, and comparative transcriptomics. One common way of identifying the cell type of a given cluster is Differential Expression Analysis to find differentially expressed genes (DEGs). Automated methods to find DEGs exist, such as the `FindAllMarkers` function in the Seurat package. This algorithm performs a non-parametric Wilcoxon Rank Sum test to identify genes that are significantly upregulated in one cluster compared to the weighted average of all others. The results are quantified by log2-fold change (LFC) and adjusted p-values. A cluster is statistically defined as a "cell type" if its top DEGs include canonical markers associated with known kidney physiology.

Problem: Identifying Differentially Expressed Genes

Input: A normalized expression matrix M and a partition of cells into C clusters. A target cluster $c_i \in C$ to be annotated.

Output: A ranked list of genes significantly upregulated in c_i relative to the weighted average of all other clusters $C \setminus \{c_i\}$.

Solution: Perform a non-parametric Wilcoxon Rank Sum test for each gene to assess the probability that a randomly selected cell from c_i has higher expression than one from the background population.

Wilcoxon Rank Sum Test:

1. For each gene g, calculate the $\log_2$ Fold Change (LFC_g) between cluster c_i and the rest of the dataset:

$$\text{LFC}_g = \log_2 \left(\frac{\text{mean}(M_{g,c_i}) + \epsilon}{\text{mean}(M_{g,C\setminus\{c_i\}}) + \epsilon} \right)$$

2. Compute the Wilcoxon Rank Sum W to determine the significance of the shift in the expression distribution:

$$W = \sum_{j=1}^{n_i} R_j - \frac{n_i(n_i+1)}{2}$$

where n_i is the number of cells in c_i and R_j is the rank of the j-th cell's expression in the combined pool.

Targeted marker gene validation

To assign UMAP clusters to their corresponding cell types, we developed a targeted cluster-labeling strategy guided by prior biological knowledge rather than relying solely on the top-ranked differentially expressed genes (DEGs). In this approach, we first defined curated marker-gene sets for candidate cell types from existing database and then compared their expression patterns across all clusters[9].

Problem: Targeted Cluster Annotation via Standardized Marker Enrichment

---

Input: A gene-by-cluster average expression matrix $A \in \mathbb{R}^{m\times c}$, a set of canonical marker genes G_T for cell type T, and differential expression results (LFC and p_{adj}).

Output: A prioritized ranking of candidate cell-type identities for each cluster based on a composite evidence score.

Solution: Transform absolute expression into relative enrichment via Z-score normalization and aggregate statistical evidence to validate lineage-specific signatures.

The Targeted Scoring Procedure:

1. **Standardization (Z-score):** For each marker gene g, calculate the mean μ_g and standard deviation σ_g of its average expression across all C clusters. Compute the Z-score for cluster j:

$$Z_{g,j} = \frac{A_{g,j} - \mu_g}{\sigma_g}$$

2. **Evidence Integration:** A marker gene $g \in G_T$ provides positive support for identity T in cluster j if it satisfies the composite criteria:

$$E_{g,j} = \begin{cases} 1 & \text{if } Z_{g,j} > 0, \text{ LFC}_{g,j} > 0, \text{ and } p_{adj} < 0.05 \\ 0 & \text{otherwise} \end{cases}$$

3. **Identity Ranking:** Aggregate the evidence across the marker set G_T to calculate the identity score S for cluster j:

$$S(T, j) = \sum_{g \in G_T} Z_{g,j} \cdot E_{g,j}$$

4. Rank the top candidate identities $\{T_1, T_2, T_3\}$ by S for manual review.

Specifically, we used Seurat's `AverageExpression` function to calculate the mean expression of each marker gene in each cluster, thereby generating a gene-by-cluster expression matrix. Because different genes naturally span different expression ranges, these raw averages are not directly comparable across markers. To address this, we standardized each gene across clusters by converting its values to z-scores. Under this transformation, a positive z-score indicates that a given cluster expresses that gene above the gene's mean level across all clusters, whereas a negative z-score indicates below-average expression. For example, if a marker gene has average expression values of 8, 2, and 1 across three clusters, then the first cluster would receive a positive z-score because its expression is above the gene-wide mean, while the other two clusters would receive negative z-scores. This step ensures that each marker contributes according to its relative enrichment pattern rather than its absolute expression magnitude.

To further strengthen the annotation, we combined this z-score-based ranking with differential expression evidence. For each cluster, we tested whether the selected marker genes were more highly expressed in that cluster than in the remaining clusters, and we retained support from markers with positive log fold-change and significant adjusted p-values. We then aggregated this evidence within each candidate marker set to generate a cluster-level score and ranked the top three candidate cell-type identities accordingly, followed by manual review. As a simple illustration, if a DCT marker such as Slc12a3 showed a positive z-score, a positive log fold-change, and an adjusted p-value below 0.05 in cluster 6, then it contributed positive evidence toward assigning that cluster a DCT identity. This strategy is more reliable than using DEGs alone, because canonical lineage markers are not always among the top-ranked differentially expressed genes, even when they are the most biologically informative features for annotation.

Results

Seurat preprocessing and Harmony integration preserved the global transcriptomic structure.

To assess sample and data quality in the available dataset, we plotted the distributions of `nFeature_RNA`, `nCount_RNA`, and `percent.mt` directly from the published Seurat object (Figure 3a). Because the original FASTQ files were not made available by Doke et al., we relied on the published preprocessed object for downstream inspection and replication. These distributions were broadly consistent with the filtering strategy described in the original study, suggesting that quality-control filtering had already been applied before the object was released. However, the mitochondrial threshold appeared comparatively permissive. In many single-cell RNA-seq workflows, a `percent.mt` threshold around 5–10% is commonly used to remove dead or stressed cells[14,16], whereas the published object contained cells with mitochondrial read fractions approaching 50%.

To identify genes contributing most strongly to cell-to-cell variation, we applied Seurat's `FindVariableFeatures` function with default parameters and selected the top 2000 variable features from a total of 16 779 genes. As expected, most genes showed low standardized variance, while only a relatively small subset showed strong variability between cells (Figure 3b). We then performed principal component analysis (PCA) using these top 2000 variable features and retained 30 principal components (PCs), consistent with the approach used by Doke et al.. To evaluate this choice, we examined an elbow plot of the standard deviation explained by the first 30 PCs (Figure 3c). The plot showed a steep decline in the first 6 PCs, followed by a more gradual taper, indicating that the strongest structure is captured early.

However, we retained 30 PCs for the main replication analysis to remain consistent with the workflow of Doke et al. To validate the number of PCs used, we performed a heatmap analysis of PC loadings (Appendix B.2). In the early components we see very distinct block-like patterns that represent coordinated gene expression across specific groups of cells. However, as PCs increase, the blocks become less defined, reflecting a transition from broad cell-type differences to more subtle variations within each cluster that contribute to the fine-grained resolution of the final UMAP.

To assess the impact of dimensionality on our results, we compared UMAP embeddings when only 10 PCs are used (Appendix B.3), compared to the 30 shown in the default paper and Figure 5 of our analysis. In the 10 PC UMAP, 5 fewer clusters are identified, and distinct clusters appear projected onto one another. This suggests that the lower-dimensional space lost information about the unique genetic signatures of the cells, forcing the algorithm to group them together incorrectly. Our decision to proceed with 30 Principal Components (PCs) for the subsequent analysis is supported by the collective evidence from the elbow plot, heatmaps, and UMAP embeddings. This choice also maintains consistency with the original publication.

The projection of cells onto the first two PCs showed that the data set retained a clear large-scale structure after the reduction in dimensionality (Figure 3d). The cells formed a continuous organized distribution, suggesting that PCA captured the major patterns of transcriptional variation in the dataset. After PCA, we applied Harmony to reduce sample-associated batch effects before downstream visualization and clustering.

The Harmony-corrected embedding remained broadly consistent with the global structure observed in PCA (Figure 4d), indicating that the integration successfully mitigated technical batch effects without collapsing distinct biological states. To quantify this, we evaluated the Shannon entropy of batch distributions within each cluster (Fig 4e). While the theoretical maximum entropy for eight batches is $lo_2(8) = 3$, observed values ranged from 1.6 to 2.8. Notably, Clusters 7, 15, 18,

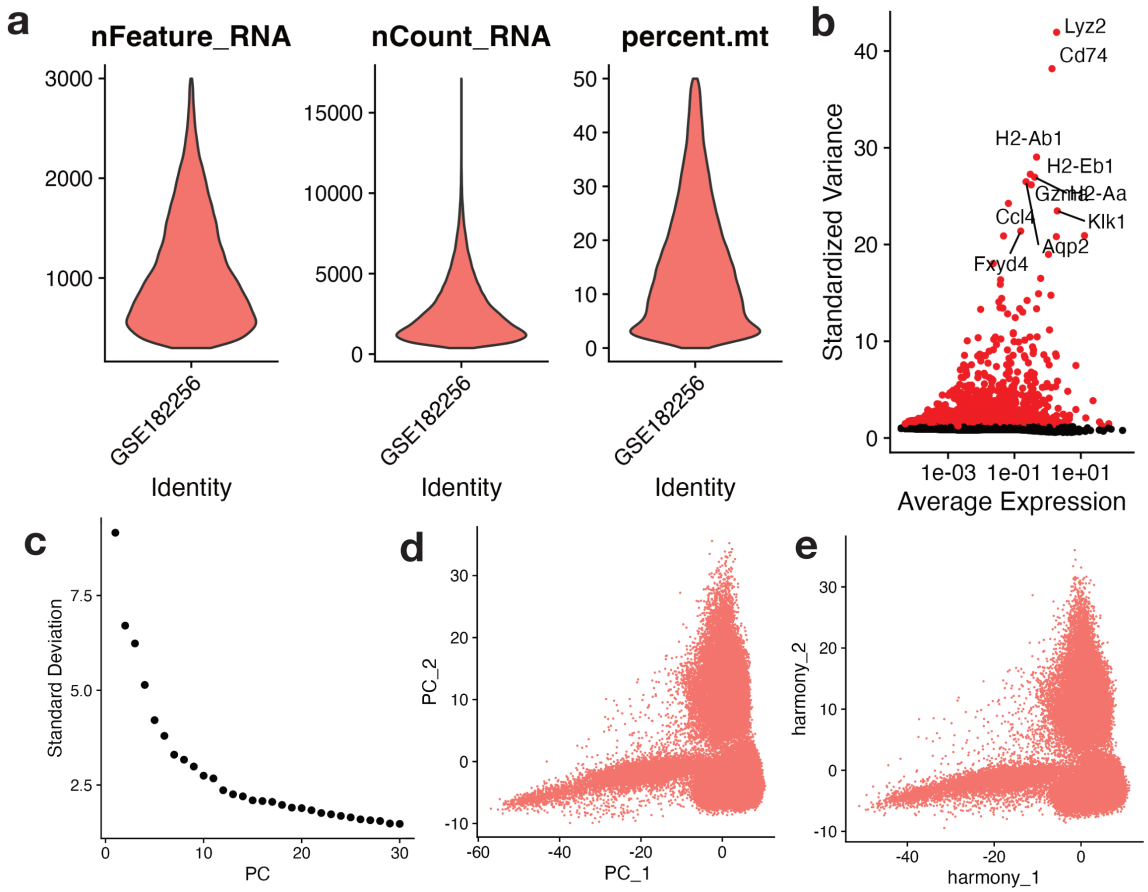

Figure 3: **Preprocessing and dimensionality reduction of the published scRNA-seq dataset.** a) Violin plots of nFeature_RNA, nCount_RNA, and percent.mt. b) Variable feature plot showing standardized gene variance versus average expression, with the top variable genes labeled in red. c) Elbow plot showing the standard deviation of the first 30 PCs. d) Projection of cells onto PC1 and PC2, showing broad structure captured by PCA. e) Projection of cells onto the first two Harmony dimensions after batch correction.

20, and 22 (Fig 4f) exhibited lower entropy, primarily driven by a high concentration of cells from UUO batches and the `Control5` batch.

In a study design involving both healthy and diseased (UUO) models, such variation in entropy is expected. While homeostatic cell populations should be evenly distributed across all batches, cell states unique to the injury response should naturally be enriched in the UUO samples. These results suggest that the integration achieved a balance where technical noise was removed, yet the distinct transcriptional signatures associated with the experimental conditions were preserved for further downstream exploration.

Global clustering and UMAP identified the major cellular structure of the dataset

To characterize the global structure of the dataset after preprocessing and batch correction, we performed UMAP visualization and Louvain clustering on Harmony-corrected embedding using Seurat. Specifically, we ran `RunUMAP`, `FindNeighbors`, and `FindClusters` on the first 30 Harmony dimensions. We identified 25 clusters in the full dataset (Figure 4a).

To assess consistency with the published annotation, we compared our Seurat-derived UMAP with the published embedding (Figure 4b). At the level of broad cell populations, the two representations showed a similar overall structure, with one large dominant region and several smaller, spatially separated populations. However, to our surprise, the agreement was weaker at the level of fine subtype resolution. For example, several of our Seurat clusters (0, 1, 2, 3, 4, 6, 7, 15, 19) occupied regions corresponding to the PT-labeled cluster (in orange) in the published embedding (Figure 4a, b). This suggests that our global clustering recovered the major organization of the dataset, but did not fully reproduce the finer published PT subtypes.

Initially, we hypothesized that the representation of PT cells as a singular, unified group in the global UMAP might be an intentional simplification by the researchers. To determine if the observed internal divisions were biologically robust or merely an artifact of hyperparameter selection, we increased the number of nearest neighbor k used in Seurat's FindNeighbors function from the default of 20 to 30 and 120 (Appendix C.4). While increasing k usually forces a more global, consol-

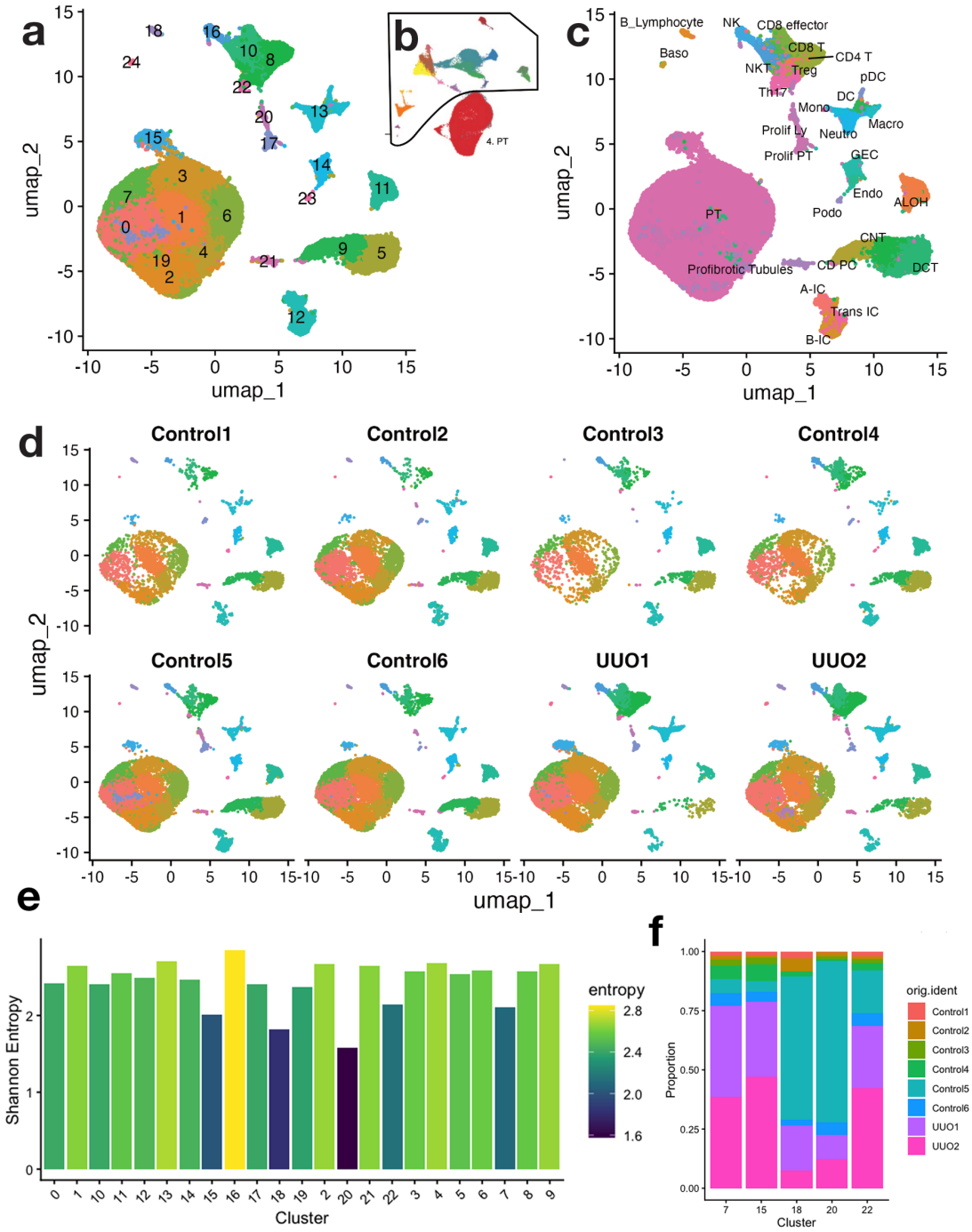

Figure 4: **Global clustering of the Harmony-corrected dataset.** a) UMAP colored by Seurat cluster identity. b) Published UMAP from Doke et al. c) UMAP colored by the published reference labels provided in the metadata. d) UMAP split by sample and colored by Seurat cluster identity. e) Shannon Entropy values for each cluster, with values ranging from 1.6 to 2.8. The maximum entropy value for 8 batches is 3 f) Proportion of each batch in Clusters 7,15,18,20,22 which displayed a lower Shannon entropy compared to other clusters. These clusters display a higher proportion of UUO1, UUO2, and Control5

idated structure, the PT sub-clusters remained distinct and failed to merge into a single cluster. This persistence suggests that the transcriptomic differences between the subclusters are not merely an artifact of visualization or clustering parameters.

To further compare our clustering results with the published annotation, we overlaid the reference labels provided in the metadata onto our UMAP embedding (Figure 4c). This comparison further highlighted differences in resolution between our clustering and the published labels. In particular, several regions that appeared as a small number of Seurat clusters in our embedding were subdivided into more specific immune populations in the published labels. For example, areas cor-

responding to our clusters 8, 10, 12, and 22 were assigned to multiple immune cell states in the reference annotation, including NK, CD8 effector, NKT, CD8 T, CD4 T, Treg, and Th17 cells (Figure 4a, c). At the same time, our clustering also appeared more resolved than the published embedding itself in some regions of the UMAP. For example, these same clusters (8, 10, 12, and 22) appeared to correspond broadly to a single orange cluster in the published embedding (Figure 4a, b). These observations support the description by Doke et al. that their analysis used a broad global embedding followed by a more detailed downstream subdivision of the main PT and non-PT cell populations. At the same time, this discrepancy was also practically important for our downstream analysis because the mismatch between clustering structures meant that the identities of the PT cells could not be transferred directly from the published clusters to our own.

As an additional check on the robustness of our clustering, we visualized the UMAP separately for each set of the experimental group (Figure 4d). The major regions of embedding were represented in both control and UUO samples, indicating that the overall clustering pattern was stable between groups. Taken together, these results suggest that Doke et al. combined a comparatively coarse global embedding with a more detailed downstream annotation, while our standard workflow recovered additional substructure at the clustering level, but did not fully reproduce the same published subtype assignments.

Marker-based approach identified PT cell populations

To obtain an initial estimate of where proximal tubule (PT) cells were located in the global embedding, we first performed a visual inspection of canonical PT marker expression on the UMAP. Specifically, we assessed cells co-expressing *Lrp2* and *Slc34a1*, two well-established PT markers[6,11]. We classified cells as PT-like if both genes had detectable expression and overlaid these double-positive cells onto the UMAP (Figure 5a). This visualization showed that double-positive cells were concentrated primarily within the large dominant region of the UMAP embedding, especially in clusters 0, 1, 2, 3, 4, 6, 7, 15, 19, with only a smaller number of positive cells scattered across other clusters, such as clusters 17 and 20. This pattern suggested that the major PT population was located within the large central cluster group, yielding a similar result as published by Doke et al.. However, given smaller clusters it also indicates that visual inspection alone may be insufficient.

To define PT clusters more rigorously, we next applied a marker-based statistical approach at the cluster level. We focused again on genes *Lrp2* and *Slc34a1*, two canonical PT markers. Using Seurat's `AddModuleScore` function, we first calculated a PT module score for each cell based on the combined expression of these PT markers. We then averaged this score across cells within each cluster to obtain a mean PT module score per cluster (Figure 5b). Based on positive mean PT module scores alone, clusters 0, 1, 2, 3, 4, 6, 7, 15, 19, 17, and 20 all showed PT-like signal. We also summarized the fraction of cells expressing *Lrp2* and *Slc34a1* in each cluster (Figure 5c,d), which showed that clusters 17 and 20 retained moderate proportions of marker-positive cells despite weaker overall PT signal than the main PT clusters.

Because the PT clusters were large and potentially heterogeneous, we also complemented module scoring with a marker-enrichment analysis. For each cluster, we used Seurat's `FindMarkers` function to test whether each of these two genes was significantly enriched relative to the rest of the dataset, and extracted the corresponding adjusted p-values. Clusters were then classified as PT only if their mean PT module score was positive and the adjusted p-values are less than 0.05. Using these combined criteria, we identified clusters 0, 1, 2, 3, 4, 6, 7, 15, and 19 as PT clusters. Interestingly, although clusters 17 and 20 showed positive PT module scores and moderate fractions of cells expressing *Lrp2* and *Slc34a1*, they were not significantly enriched for either marker relative to the rest of the dataset and therefore were not retained as PT clusters. This suggests that clusters 17 and 20 carried some PT-like signal, but did not exhibit the cluster-specific enrichment expected of canonical PT populations.

Taken together, these results allowed us to seperate the dataset into PT and non-PT populations with a marker-guided strategy. We defined a PT subset consisting of clusters 0, 1, 2, 3, 4, 6, 7, 15, and 19, while excluding clusters 17 and 20 despite their partial PT-like signal.

Identification of non-PT subclusters

To identify non-PT cell clusters, we first attempted to perform an unbiased annotation using differentially-expressed gene analysis (DEG). Using the `FindAllMarkers` function in Seurat, we computed and found the top-10 DEGs for each cluster(Figure A.1). This method aligned with the paper's. However, Doke et al. did not explain how clusters were annotated after finding the DEGs for each cluster, except by mentioning the use of the "previously reported marker gene". Furthermore, of the top 10 DEGs we found, none of them matches known marker genes for any non-PT cell types (Figure A.1). Using our own annotation algorithm, we were able to reproduce cluster annotations that partially align with the paper's result. The cell markers obtained from the reference paper and database are organized in Figure 6b. The UMAP algorithm identified 26 clusters from the PCA reduced data, which were then annotated with the predicted label (Figure 6a).

To verify that cluster annotations are accurate, we created a bubble plot with the same list of marker genes used in the reference paper, and examined the expression level of each gene across all annotated cell clusters (Figure 6c). Most clusters show high expression of the canonical marker genes expected for their assigned identities, indicating that the annotation is broadly consistent with known biology. Notably, the basophil cluster is supported by strong expression of canonical basophil markers such as Fcer1a and Mcpt8, which is particularly important because basophils are among the key inflammatory cell popula-

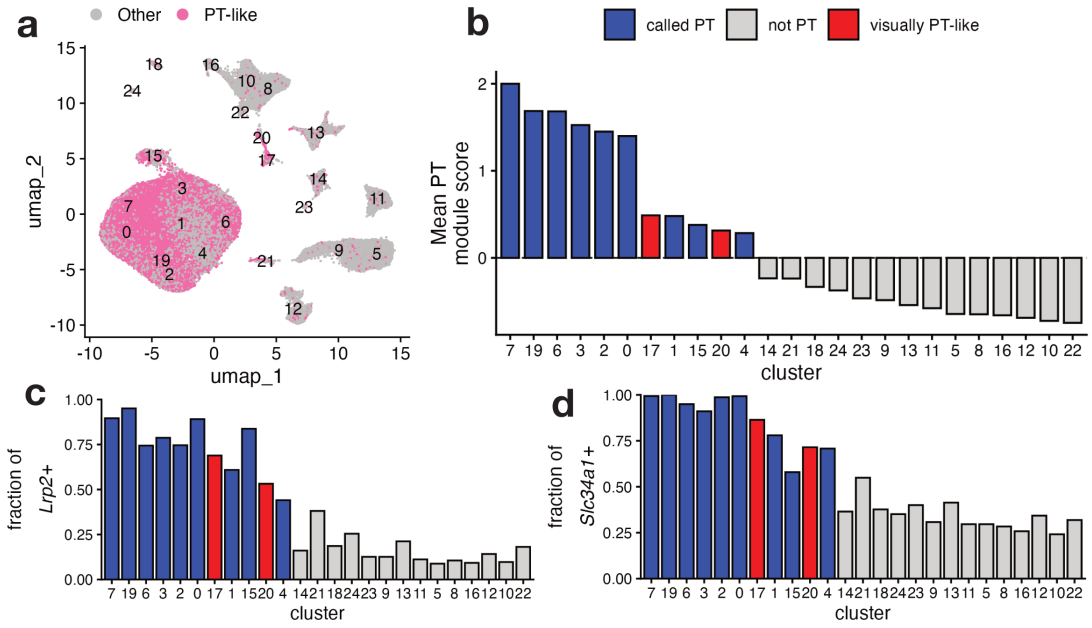

Figure 5: **Marker-based identification of proximal tubule-like clusters.** a) UMAP of the global dataset showing cells with detectable co-expression of the canonical PT markers *Lrp2* and *Slc34a1* in pink. b) Mean PT module score for each Seurat cluster, calculated using *Lrp2* and *Slc34a1*. Blue: clusters called as PT by the combined marker-based criteria; Gray: clusters not called as PT; Red: appeared PT-like by visual inspection in panel a but not retained after statistical filtering. c) Fraction of cells in each cluster with detectable *Lrp2* expression. d) Fraction of cells in each cluster with detectable *Slc34a1* expression.

tions of interest in the UUO kidney. Although some marker genes are expressed across more than one cluster and not every cluster matches the reference paper exactly, the overall pattern suggests that the predicted labels capture the major cell populations present in the non-PT compartment. These results support the conclusion that our annotation strategy can recover biologically meaningful cluster identities even when top-ranked DEGs alone are insufficient for direct cell-type assignment.

Identification of PT subclusters

To identify the PT subclusters matching those reported in Doke et al.'s paper, we repeated the downstream Seurat workflow on the UUO PT cells. This replication was necessary because the original paper reports PT-specific subclustering rather than simply relying on the global kidney cluster embeddings alone. Unlike the paper, even with the same parameters, we obtained 13 clusters instead of 8 (Figure 7a). Although this differed from the 8 PT subclusters reported in the original study, the difference in cluster number was not unexpected, because clustering resolution and dataset-specific variation can split a single biological state into multiple computational clusters.

To reproduce the PT states shown in the paper, we first attempted a direct marker-based approach using representative genes from the reference figure. Using a cluster level bubble plot, we visualized canonical PT and subtype-associated markers including *Apom*, *Slc5a12*, *Gsta4*, *Slc6a13*, *Pdgfb*, *Wfdc15b*, *Mki67*, and *Cd52*, which represents precursor, S1, S2, S3, profibrotic, transient, proliferating, immune PT cell types (Figure 7b). Similar to the paper, we also included *Slc34a1* and *Slc13a3*, which are canonical PT cell markers. In the bubble plot, we observed similar patterns across our unlabeled clusters compared to the papers. However, unlike the clean separation shown in the paper, our bubble plot showed a much less discrete structure. Several markers appeared more diffuse across clusters, such as *Slc6a13* and *Apom*, and the expected subtype boundaries were not immediately obvious.

To specifically identify the profibrotic PT population, we then moved to a more targeted analysis centered on the genes emphasized by the original study. We examined the co-expression *Lrp2* and *Slc34a1* with *Pdgfb* and other inflammatory markers mentioned in the paper, such as *Cd74*, *Tnfrsf12a*, *Cxcl1*, *Cxcl10*, and *Cxcl16* (Figure 7c, d). Among the PT subclusters, one cluster consistently stood out by showing the strongest enrichment for *Pdgfb*, while still retaining canonical PT markers, and displaying inflammatory markers that shift cells to a profibrotic inflammation state mentioned in the paper. Based on both approaches, we considered cluster 10 to be the strongest candidate for the profibrotic PT-like cluster in our dataset. Interestingly, our UMAP embedding shows a vaguely similar pattern as the papers, albeit our plot appears to be rotated 180°. Despite this, the profibrotic PT clusters are located at a similar position within the entire plot across both our UMAP and the papers.

To move beyond visual inspection alone, we then attempted a more systematic cluster-labeling strategy using curated marker sets for each proposed PT subtype. These marker sets were

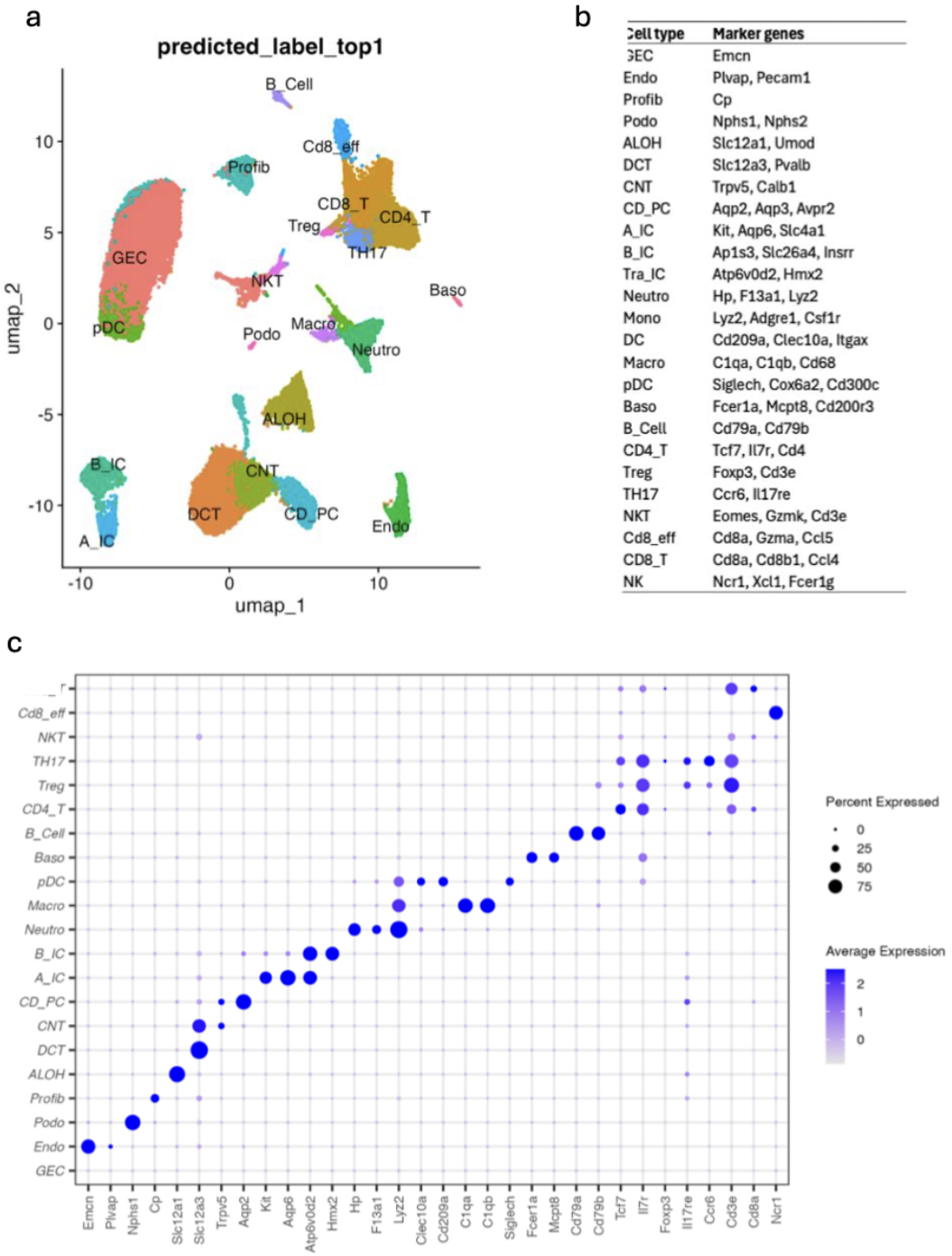

Figure 6: **non-PT cell cluster annotation and marker gene expression validation.** a) Annotated UMAP embedding. b) A list summarizing the marker genes used for each cell type. These are obtained from reference paper, literature and databases[6,9]. c) Expression of known marker genes in each predicted cell cluster.

compiled from the reference study and related kidney single cell literature, and included signatures for cell types such as S1, S2, S3 PT cells. For each marker set, we again used Seurat's `AddModuleScore` function to calculate a per-cell score representing how strongly that cell expressed the corresponding gene program. These scores were averaged across all cells in each cluster, giving us a top scoring and second-highest scoring label for every PT subcluster. This method was useful because it provided a more quantitative alternative to manually reading bubble plots. However, although the method was able to assign plausible labels to many clusters, the results were never fully consistent with the expected biology and outcome. For example, some marker gene sets overlapped substantially, and some markers were not strongly specific in our data, and the

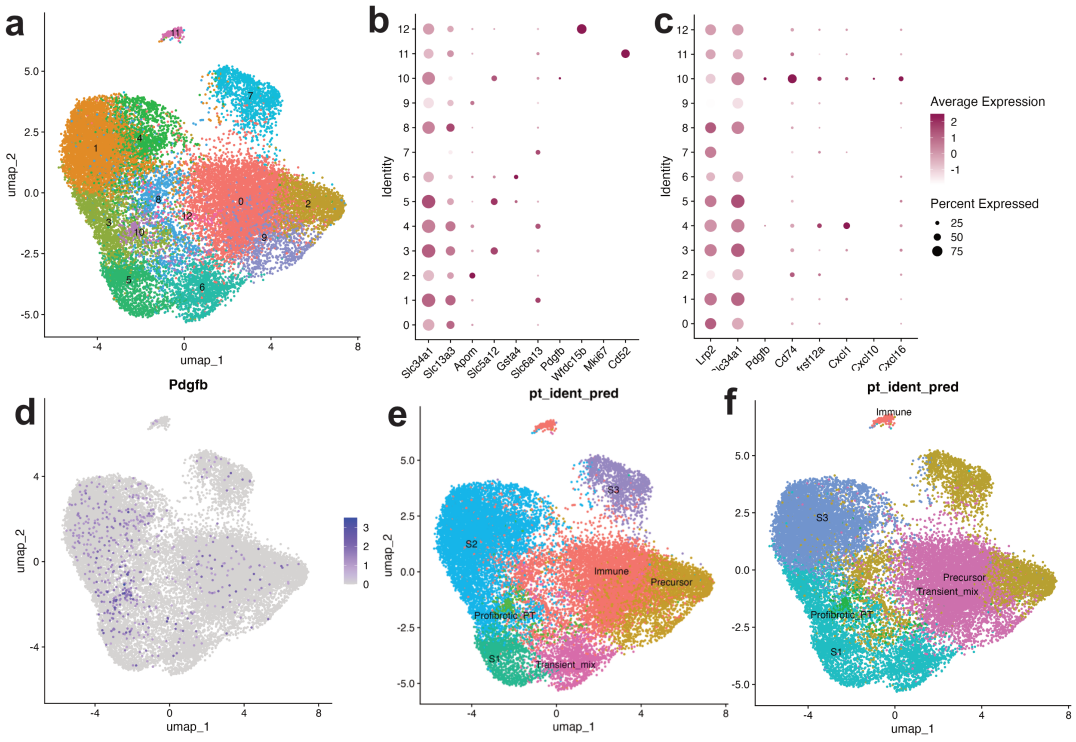

Figure 7: **Identification and annotation of PT subclusters in UUO kidneys. (a)** UMAP of PT cells after PT-specific subclustering, showing 13 computational clusters. **(b)** Dot plot of canonical PT markers and subtype-associated markers used to compare PT subclusters with the PT states reported by Doke et al.. **(c)** Dot plot of canonical PT markers, *Pdgfb*, and inflammatory markers used to identify profibrotic PT-like cells. **(d)** Feature plot showing *Pdgfb* expression across PT subclusters. **(e)** UMAP annotation based on manual marker inspection, identifying candidate PT states including precursor, S1, S2, S3, transient-like, immune-like, and profibrotic PT-like populations. **(f)** UMAP annotation based on module-score-assisted subtype assignment.

highest-scoring label for a cluster was not always the one that best matched the visual expression pattern of the key reference genes. Most importantly, the module score approach helped organize the clusters into candidate PT states, but it failed to resolve the ambiguity between closely related subtypes such as S1, S2, and S3 PT cells. One difficulty with this approach is the availability of existing data and marker gene sets. In contrast to identifying and subsetting PT cells, where we were much more certain about which canonical genes to use, here we were much less sure which genes should define each PT subtype. After extensive literature search, we found that different sources reported different markers for precursor, S1, S2, and S3 states, and these marker sets did not always agree with one another (Kirita et al. 2020, Hu et al. 2022). In addition, one major roadblock for this approach is the lack of explanation for the immune-like subcluster, which is characterized in the paper mainly by *Cd52*. Because *Cd52* is essentially the only specific information available for this cluster, we were unable to accurately capture its identity using a module-score method, as AddmoduleScore is much more informative when given a broader gene program rather than one single marker (Figure 7e). As a result, while this method was useful for generating provisional subtype labels, the output should be interpreted with caution.

To address this, we next turned to Doke et al.'s supplementary table 3, which reports marker genes for PT subclusters in UUO kidneys, and selected some genes with low p-values for each labeled cluster randomly. This gave us a more data-driven way to define subtype-specific marker sets, rather than relying on markers gathered from the literature. However, even this approach did not yield fully reproducible results (Figure 7f). Although some of the selected genes appeared enriched in the expected clusters, the overall subtype assignments were still unstable and did not cleanly reproduce the PT subcluster structure reported in the paper. These results suggest that marker-based PT subtype annotation is highly sensitive to marker selection, and that exact subtype labels in this dataset cannot be recovered using a simple framework alone.

Together, these analyses show that PT subtype annotation was one of the most challenging parts of the replication. Across direct marker inspection, targeted profibrotic marker analysis, literature-based module scoring, and marker sets derived from Supplementary Table 3 in Doke et al., we consistently recovered a plausible profibrotic PT-like population. However, the exact fine-grained PT subtype labels reported in the original study were not robustly recovered using simple marker-based or module-score-based annotation alone.

Cell Trajectory Analysis

Cell trajectory is a computational method that positions single cells along a trajectory that represents their progress in a biological process such as differentiation or transcription. In order to perform this analysis, we used the Python package scVelo which is designed to perform RNA velocity analysis in single cell data, which estimates the future transcriptional state of each cell based on the ratio of unspliced to spliced mRNA. If the expected amount of RNA is lower than the actual value, then the gene is being actively up-regulated. This analysis allows us to see which cell states are transitioning into which other states, which cell populations are transcriptionally early or late, and which cell populations are transcriptionally stable endpoints. This is important for identifying which PT subcluster the profibrotic PT cells originate from and how.

We performed RNA velocity analysis using combined data from kidney samples UUO1 and UUO2 in the form of loom files, which store high-dimensional matrices of spliced and unspliced mRNA transripts. Loom files were not provided, so we created our own starting from raw SRA files. Because the original FASTQ files were unavailable, we used fasterq-dump from the SRA toolkit to create our own. Then, we ran Cellranger to align the FASTQ file to a mouse reference geneome which resulted in a BAM file containing all aligned reads as well as cell barcodes and UMI tags. Then, velocyto was used on the BAM files to quantify spliced, unspliced, and ambiguous counts resulting in a loom file that can be used in scVelo.

From the Doke et. al. paper, we see a differentiation pattern starting with precursor PT cells and ending at profibrotic PT cells. We also see evidence of a differentiation path from S2 to S3 with an endpoint around the S1 subcluster. As we can see in Figure 8a, we see evidence that transcription originates in the precursor cells and that these cells differentiate into the profibrotic cluster along with the S1 subcluster. Interestingly, the S3 subcluster in our figure appears to rather isolated. These results are further supported by Table 1, where we can see that precursor cells have a low pseudotime of 0.669, indicating that they are transcriptionally early in the PT differentiation pipeline. The S3 subcluster has a similar pseudotime of 0.652, but does not appear to transition to other cell types like the percursor cells. Also in Table 1, we see that the profibrotic PT cells have the highest pseudotime of 0.785, which indicates that they occur transcriptionally latest, as expected. Termination at the profibrotic PT subcluster is not obvious in Figure 8a due to the small number (315) of profibrotic PT cells, though the velocity pseudotime statistics provide us with additional information.

Monocle3 Trajectory and Pseudotime Analysis

We used Monocle3 to create trajectory plots detailing the structural topology of the cell differentiation path. These paths help verify the results of the RNA velocity plot by detailing which cell types are connected to each other.It should be noted that while Doke et. al. used Monocle2 to plot cell trajectory, we used Monocle3, which unlike Monocle2, plots the trajectory directly onto the UMAP and is designed for large, complex, single cell datasets. Monocle3 is also able to learn complex disjoint trajectories and doesn't assume a single structure. Additionally, as of the time of writing this, Monocle2 has been deprecated and the Trapnell Lab, the developrs of Monocle, recommend using Monocle3 (Trapnell et. al). The scVelo analysis was performed on 20,540 cells following barcode matching between the Seurat object and velocyto loom files (90.6% match rate), while the Monocle3 trajectory analysis was performed on the full dataset of 15,630 PT cells (excluding immune cell contamination) from the Seurat object. The slight difference in cell counts between the two analyses reflects the 9.4% of cells whose barcodes could not be matched to the loom files and were therefore excluded from the scVelo analysis only.

Based on Figure 8b, we can assume that the origin occurs in the precursor cells. From there, the trajectory appears to diverge. One direction leads into the S3 subcluster, while the other passes through the transient mix and into the S1 subcluster. From the S1 subcluster, we then reach the S2 and profibrotic subclusters. While we cannot infer directionality from the Monocle3 trajectory plots, these results do support our conclusions from the RNA velocity plot. The S3 subcluster is fairly isolated with its closest connection being to the precursor cells, indicating that it is likely a distinct PT subtype that does not primarily contribute to the profibrotic population. Additionally, the precursor cells are connected to the S1 subcluster, which is subsequently connected to the S2 and profibrotic clusters.

In order to verify the cell trajectory path, we also plotted the pseudotimes for each of the PT subclusters. Pseudotime is a measure that represents each cell's progress along a differentiation trajectory. Darker values represent earlier transcriptional states while lighter values represent later states. Calculating pseudotime values using Monocle3 helps validate our RNA trajectory plot and gives us a holistic view of the differentiation pathway. As we can see in Figure 8c, the precursor cells have a dark blue value representing the lowest pseudotime and the origin of the differentiation trajectory. The subclusters with the next lowest times are the transient mix and S3 subclusters. From the transient mix, which is likely an intermediate stage, the next subcluster is the S1 subcluster and profibrotic cluster. This validates our result that precursor cells transition into profibrotic cells. These results are further validated by Table 2, which details the exact pseudotime values for each cluster. Based on this table, we can see that precursor cells are transcriptionally earliest, which is consistent with them being the origin point in the RNA velocity figure. One unexpected result is that the pseudotime values suggests that the S2 subcluster is the terminal state of the graph due to having the highest pseudotime of 24.962, and not profibrotic subcluster which has the second highest pseudotime of 19.182. This differs from the RNA velocity plot, and scVelo analysis which show the profibrotic subcluster as the terminal state. There are a couple possible reasons for the discrepancy. First, the pseudotime plot

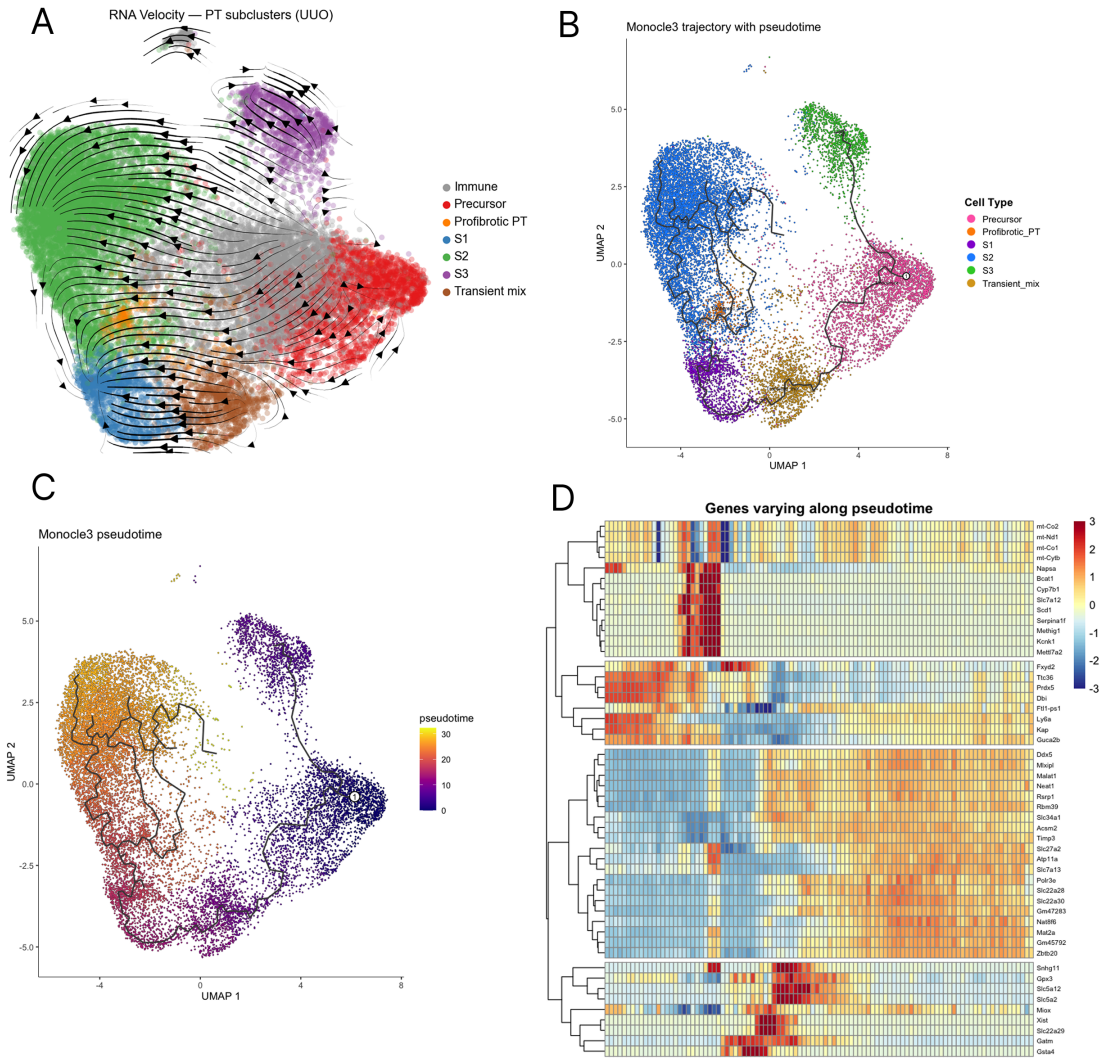

Figure 8: **Cell trajectory plot overlaid on the UMAP predicting subclustering of PT cells.** a) Cell trajectory plot overlaid on the UMAP predicting subclustering of PT cells. b) Monocle3 trajectory plots overlaid on PT subclustering UMAP. Immune cells are exlucde to focus on PT cell interactions c) Monocle3 pseudotime plot overlaid on PT subclustering UMAP d) Heatmap depicting top 50 genes that vary most significantly along the Monocle3 pseudotime trajectory. Rows represent genes, and columns represent pseudotime bins where cells with earlier pseudotimes are towards the left.

does not show directionality like the RNA velocity plot. It correctly identifies precursor cells as the origin, but may also be identifying the S2 subcluster as the most transcriptionally different cluster compared to the precursor cells. The scVelo and Monocle3 calculations are different, as scVelo focuses on how far along the velocity trajectory is based on the current transcriptional dynamics of each cell while Monocle3 focuses on how different the transcriptional similarity is between cells based on how far they are from the root. The S2 cells being the most different from the precursor cells does not necessarily mean that S2 cells are the true terminal point in the graph.

From the heatmap in Figure 8d, we can observe which specific genes are more highly expressed during different stages in the cell trajectory. As seen above, there are 4 relatively distinct gene clusters. The 2nd cluster of genes is highly expressed early on, and is followed by high expression in cluster 1, then cluster 4, and lastly cluster 3. Using both the pseudotime plot and UMAP, we can infer that cluster 2 likely corresponds to precursor cells. We see a relatively smooth transition into the 1st cluster from the 2nd cluster, which suggests that the 1st cluster is representative of the transient mix intermediate steps. The 4th cluster likely represents the S2 subcluster, as we see expression of Gsta4, which is shown to be expressed in S2 cells in Figure 2b of the Doke et. al. paper. This implies that the S2 cluster is not the endpoint of the trajectory, which is what we expect. Cluster 3 likely represents the S1 and profibrotic subclusters, which are expected to be towards the end of the trajectory.

Table 1: Velocity pseudotime distribution among different PT subclusters. A Kruskal-Wallis test confirmed highly significant differences in pseudotime distributions across all clusters ($\chi^2 = 12{,}339.58$, df = 5, $p < 10^{-300}$).

Cluster	Mean	SD	Median	Count
Immune	0.719	0.031	0.721	5666
Precursor	0.669	0.031	0.675	2617
Profibrotic PT	0.785	0.053	0.797	315
S1	0.707	0.129	0.681	1384
S2	0.762	0.020	0.757	8168
S3	0.652	0.037	0.645	1135
Transient Mix	0.463	0.252	0.490	1255

Table 2: Monocle3 pseudotime statistics per cluster. A Kruskal-Wallis test confirmed highly significant differences in pseudotime distributions across all clusters ($\chi^2 = 12{,}339.58$, df = 5, $p < 10^{-300}$).

Cluster	Mean	SD	Median	Count
Precursor	2.673	3.85	2.113	2933
Profibrotic PT	19.182	4.66	20.339	328
S1	15.743	1.93	16.157	1394
S2	24.962	3.04	25.840	8247
S3	6.590	1.23	6.660	1244
Transient Mix	11.018	3.92	10.010	1484

Discussion

Limitation of the cluster label prediction strategy

A related challenge is that clustering annotation is often more difficult than it first appears. Even when UMAP shows visually distinct groups, assigning biologically accurate labels to these clusters is not straightforward. In practice, cell identity is rarely determined by a single marker or even a small fixed set of markers. Many genes are commonly expressed across related cell types, especially when these cell types are from the same tissue. For example, Slc12a1, a marker gene for the ascending loop of Henle (ALOH), is among the top-differentially expressed genes in many clusters. As a result, cluster annotation often involves interpreting patterns of relative enrichment rather than identifying a single definitive signature.

In our own PT subclustering analysis, this difficulty became very apparent. We were able to recover broad PT-associated structure and identify plausible profibrotic PT-like clusters through targeted inspection of Pdgfb together with inflammatory markers. However, when we systematically assigned subtype labels such as precursor, S1, S2, S3, profibrotic, transient, proliferating, and immune-like PT cells using fixed marker sets and modules scores, the results were not reproducible. Different marker lists taken from literature often gave different subtype assignments, and even marker genes selected from Supplementary Table 3 of the reference paper did not yield clean and reproducible substyle structure. This suggests that the boundaries between these PT states are not as discrete as the paper implied. Cells may lie along differentiation trajectories, activation gradients, or stress-response programs, so that the boundaries between groups are not always sharp. However, a clustering algorithm may still partition this continuum into discrete clusters. In such cases, an apparent cluster may represent a set of intermediate or transitional cells rather than a pure cell type with a uniform identity.

One highly likely possible explanation is that the Doke et al. did not derive these PT subtype labels from a single explicit rule based annotation pipeline. Instead, we think it is likely that the authors first ran a general differential expression analysis on the PT subclusters and then assigned labels semi-manually using prior biological knowledge about PT segment identity, injury states, and known marker genes. This knowledge may have come from previous published studies, internal preliminary experiments for grant applications, or even learned during a spirited discussion that happened between an eager PhD student and their PI. In any case, the final labels may reflect a combination of statistical output and expert interpretation rather than a strictly reproducible scoring framework. This would explain why we were able to identify some of the major markers like Pdfgb, but were unable to reproduce the exact same subtype labels using a fully systematic module-score approach alone.

UMAP visualization generates hypothesis, not evidence.

A limitation of our analysis is that visualization can easily create a false sense of certainty. UMAP and related cluster plots are useful for summarizing high-dimensional data in two dimen-

sions, but they do not provide direct proof that each visible cluster corresponds to a biologically discrete cell type. The apparent separation of clusters can be influenced by preprocessing choices such as number of pcs, dimensionality reduction, clustering resolution, and the selected marker genes used for annotation. As a result, visual separation on the embedding may overstate how confidently cells can actually be assigned.

This issue is especially relevant in our cluster annotation strategy. Because our goal was to recover a predefined set of biologically significant cell types, we used curated set of marker-gene and ranked the most likely identities for each group. However, in practice, many clusters contained cells whose identities were not strongly supported by multiple markers and would more appropriately be considered uncertain (Appendix D.5 & D.6). Despite this, a final label was still assigned because the predicted identity matched one of the target cell types in our annotation scheme. This means that the resulting visualization is best interpreted as a practical approximation rather than a definitive classification.

It is also true that the in silico annotation in Doke et al.'s work was used primarily as a discovery step rather than as an endpoint analysis. In this case, the researchers may only have needed to identify a plausible candidate cell population, profibrotic PT, with strong enough evidence to motivate downstream experimental validation. Once such a cell type was found in silico, the focus of these excited scientists would then shift from computational refinement to wet-lab confirmation, such as validating the biological relevance of the identified marker genes and its cell type. Doke et al. performed in situ hybridization to validate several of their single-cell findings. For example, they showed that Pdgfb is indeed expressed in UUO kidneys within Hnf4a-positive proximal tubule cells, confirming the presence of a profibrotic PT population in vivo; they also used in situ hybridization to verify Cxcl1 expression in PT cells and Cxcr2 expression in Mcpt8-positive basophils, supporting the proposed interaction between profibrotic tubules and recruited basophils.

From this perspective, the purpose of clustering and annotation pipeline might have been less about generating reproducibility in terms of subtype-calling, and more about narrowing the search space to a biologically meaningful candidate population.

Comparison of Replicated Results with the Original Study

One of the clearest differences between our replication and the published study emerged in the PT subclustering analysis. Although we followed the same general downstream Seurat workflow, we obtained 13 PT clusters rather than the 8 PT subclusters reported by[6] This discrepancy is not trivial because it suggests that the PT landscape is highly sensitive to analytical choices and may not partition into a single stable set of discrete subtypes. In our data, the overall PT-associated structure was still evident, and we were able to identify a plausible profibrotic PT-like population, but the finer subdivision into precursor, S1, S2, S3, transient, proliferating, immune-like, and profibrotic states was much less clean than in the published figures.

When performing the cell trajectory analysis on the PT subclusters we did find that precursor cells were at the transcriptional origin and eventually differentiated into profibrotic cells, which are transcriptionally later. This is an expected result. However, so of the specific trajectories differ from the Doke et. al. paper. For example, in our results we found that the S3 subcluster appeared to be isolated from the other clusters, indicating that it may be separate from the transcriptional pathway of the other PT clusters. In the paper, we see that the S3 subcluster appears to be an intermediary stage from the S2 to S1 subcluster. Additionally, due to the low amount of profibrotic cells observed in our data, the Monocle3 heatmap did not seem to count profibrotic marker genes as highly significant when choosing the top 50 genes varying along pseudotime trajectory. This made it difficult to conclude which cell cluster is transcriptionally latest based on the heatmap alone. In fact, very little of the genes in our heatmap actually match up with those mentioned in the Doke et. al. paper, likely due to limitations of our initial clustering.

The immune-like PT cluster was particularly difficult to reproduce in a systematic way. In the published study, this cluster is represented mainly by *Cd52* in the summary bubble plot, which makes it appear as though the subtype is defined by a single marker[6]. However, Supplementary Table 3 indicates that this population was associated with multiple statistically significant differentially expressed genes rather than *Cd52* alone. This suggests that the authors likely selected *Cd52* as a representative marker for visualization because it was biologically interpretable based on prior knowledge, not because it was the only supporting DEG. In our replication, this distinction mattered. A module-score or fixed-marker approach is much less reliable when the published figure emphasizes only one marker, while the broader DEG program underlying that cluster is not made explicit in the main text. As a result, the immune-like PT state was harder to recover reproducibly, and our difficulty in identifying it likely reflects both the dependence of annotation on expert judgment and the fact that the final published label was shaped by biological interpretation as much as by a fully specified computational rule.

Conclusion

The exploration of novel subclusters necessitates a deep dive into the choices made by scientists, a process facilitated by sensitivity analysis, parameter validation, and cell trajectory analysis. Crucially, this work underscores the importance of reproducibility in the discovery of new subclusters. While we were not able to exactly replicate this study, our analysis provides a basis for the experimental verification of the profibrotic PT cluster, which the original researchers carry out to supplement their computational analysis results.

While not a deterministic statistical method, UMAP remains a valuable tool for exploration and hypothesis generation. It offers an intuitive visual framework for identifying targets that warrant further investigation and experimental verification. Although the results of our analysis complement the visual message suggested by the UMAP embeddings, this is not always the case. Further analysis such as the validation tests and sensitivity analysis we performed should always accompany any interpretation from UMAP.

Despite the inherent subjectivity in interpreting single-cell RNA sequencing (scRNA-seq) studies, their value to the scientific community is undeniable as it has led to discoveries of cellular pathways and biological mechanisms that allow for new therapies to be developed. Although automated computational methods are continually advancing in complexity and accuracy, scRNA-seq studies consistently highlight the essential need for biological intuition when navigating this unknown space.

At the end of the day, the search for therapeutics is not dependent on a perfectly elegant or robust computational method, although that is of course the ideal scenario. In the case of scRNA-seq studies, we often weave together many different tools to form a coherent picture that must still be experimentally verified. These limitations are a testament to the human condition that, despite our limited understanding and finite tools, we persist in what often appears to be a losing battle against biological complexity. Yet, through this perseverance, we uncover the fundamental truths necessary to transform an impossible struggle into a tangible path toward healing.

References

[1] M. Anderson-Crannage, A. M. Ascensión, O. Ibanez-Solé, H. Zhu, E. Schaefer, D. Ottomanelli, B. Hochberg, J. Pan, W. Luo, M. Tian, Y. Chu, M. S. Cairo, A. Izeta, and Y. Liao. Inflammation-mediated fibroblast activation and immune dysregulation in collagen VII-deficient skin. 14:1211505, 2023. ISSN 1664-3224. doi: 10.3389/fimmu.2023.1211505. URL https://www.frontiersin.org/articles/10.3389/fimmu.2023.1211505/full.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008. doi: 10.1088/1742-5468/2008/10/P10008. URL https://doi.org/10.1088/1742-5468/2008/10/P10008.

[3] R. D. Bülow and P. Boor. Extracellular matrix in kidney fibrosis: More than just a scaffold. 67(9):643–661, 2019. ISSN 0022-1554, 1551-5044. doi: 10.1369/0022155419849388. URL https://journals.sagepub.com/doi/10.1369/0022155419849388.

[4] R. Chazarra-Gil, S. van Dongen, V. Y. Kiselev, and M. Hemberg. Flexible comparison of batch correction methods for single-cell rna-seq using batchbench. *Nucleic Acids Research*, 49(7), Feb 2021. doi: 10.1093/nar/gkab004.

[5] J. Z. Clark, L. Chen, C.-L. Chou, H. J. Jung, J. W. Lee, and M. A. Knepper. Representation and relative abundance of cell-type selective markers in whole-kidney RNA-seq data. 95(4):787–796. ISSN 00852538. doi: 10.1016/j.kint.2018.11.028. URL https://linkinghub.elsevier.com/retrieve/pii/S0085253818309128.

[6] T. Doke, A. Abedini, D. L. Aldridge, Y.-W. Yang, J. Park, C. M. Hernandez, M. S. Balzer, R. Shrestra, G. Coppock, J. M. I. Rico, S. Y. Han, J. Kim, S. Xin, A. M. Piliponsky, M. Angelozzi, V. Lefebvre, M. C. Siracusa, C. A. Hunter, and K. Susztak. Single-cell analysis identifies the interaction of altered renal tubules with basophils orchestrating kidney fibrosis. 23(6):947–959, 2022. ISSN 1529-2908, 1529-2916. doi: 10.1038/s41590-022-01200-7. URL https://www.nature.com/articles/s41590-022-01200-7.

[7] G. Gibson. Perspectives on rigor and reproducibility in single cell genomics. *PLOS Genetics*, 18:e1010210, 2022. doi: 10.1371/journal.pgen.1010210. URL https://doi.org/10.1371/journal.pgen.1010210.

[8] J. Hartupee and D. L. Mann. Role of inflammatory cells in fibroblast activation. 93:143–148, 2016. ISSN 00222828. doi: 10.1016/j.yjmcc.2015.11.016. URL https://linkinghub.elsevier.com/retrieve/pii/S002228281530122X.

[9] C. Hu, T. Li, Y. Xu, X. Zhang, F. Li, J. Bai, J. Chen, W. Jiang, K. Yang, Q. Ou, X. Li, P. Wang, and Y. Zhang. CellMarker 2.0: an updated database of manually curated cell markers in human/mouse and web tools based on scRNA-seq data. 51:D870–D876. ISSN 0305-1048. doi: 10.1093/nar/gkac947. URL https://doi.org/10.1093/nar/gkac947. _eprint: https://academic.oup.com/nar/article-pdf/51/D1/D870/48440910/gkac947.pdf.

[10] I. Korsunsky, 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. *bioRxiv*, 2018. doi: 10.1101/461954. URL https://www.biorxiv.org/content/early/2018/11/05/461954.

[11] T. Kusaba, M. Lalli, R. Kramann, A. Kobayashi, and B. D. Humphreys. Differentiated kidney epithelial cells repair injured proximal tubule. 111(4):1527–1532. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1310653110. URL https://pnas.org/doi/full/10.1073/pnas.1310653110.

[12] Y. Liu. Cellular and molecular mechanisms of renal fibrosis. 7(12):684–696, 2011. ISSN 1759-5061, 1759-507X. doi: 10.1038/nrneph.2011.149. URL https://www.nature.com/articles/nrneph.2011.149.

[13] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar 1982. doi: 10.1109/tit.1982.1056489.

[14] M. D. Luecken and F. J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. 15(6):e8746, 2019. ISSN 1744-4292, 1744-4292. doi: 10.15252/msb.20188746. URL https://link.springer.com/article/10.15252/msb.20188746.

[15] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. 2018. URL http://arxiv.org/abs/1802.03426. cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap.

[16] D. Osorio and J. J. Cai. Systematic determination of the mitochondrial proportion in human and mice tissues for single-cell RNA-sequencing data quality control. 37(7):963–967, 2021. ISSN 1367-4803, 1367-4811. doi: 10.1093/bioinformatics/btaa751. URL https://academic.oup.com/bioinformatics/article/37/7/963/5896986.

[17] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33:495–502, 2015. doi: 10.1038/nbt.3192. URL https://doi.org/10.1038/nbt.3192.

Appendix

Appendix A Top DEGs Heatmap

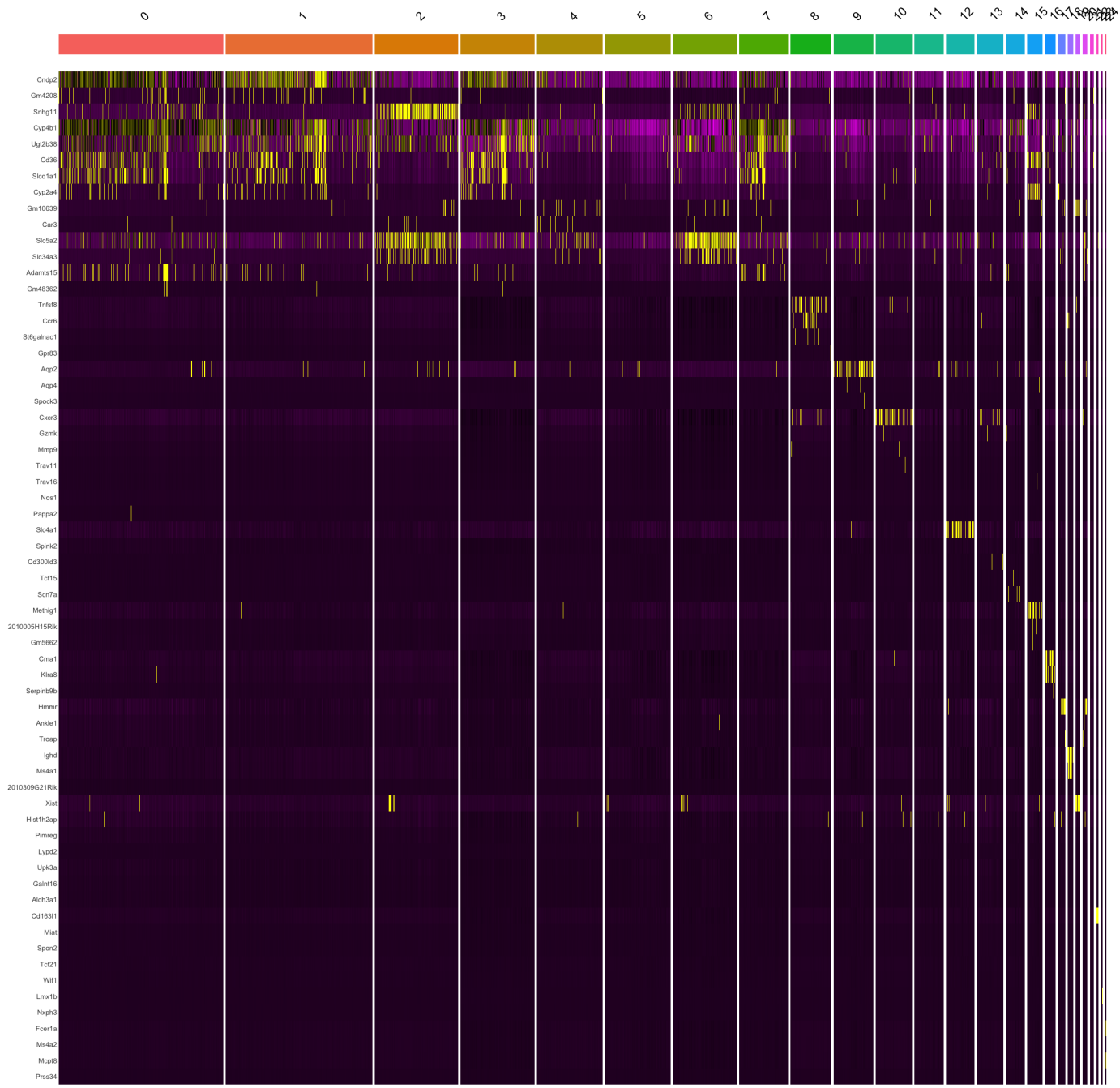

Figure A.1: Heatmap of top DEGs for each cluster.

Appendix B PC Loadings Heatmap

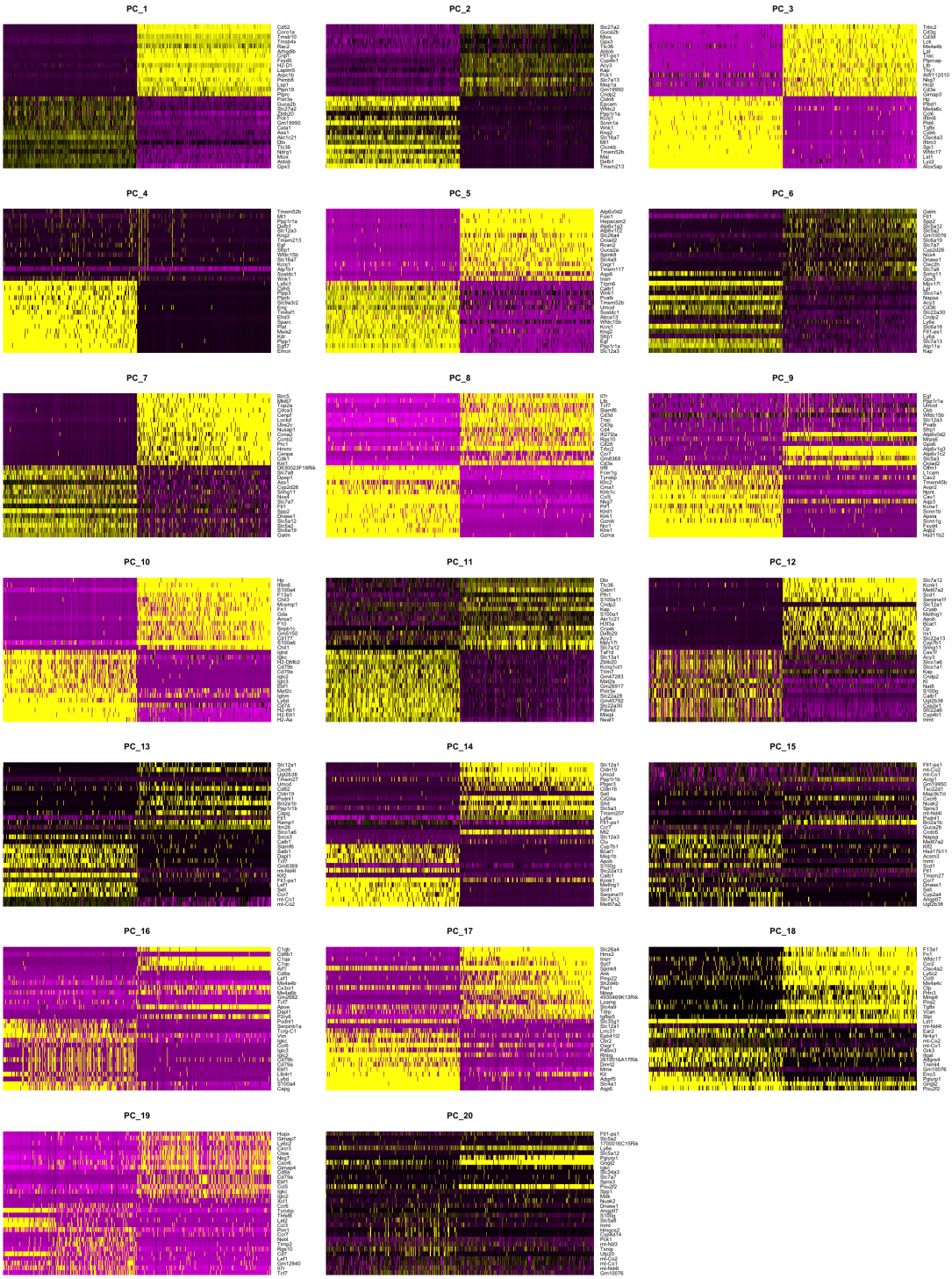

Figure B.2: Heatmap of the first 30 PC Loadings

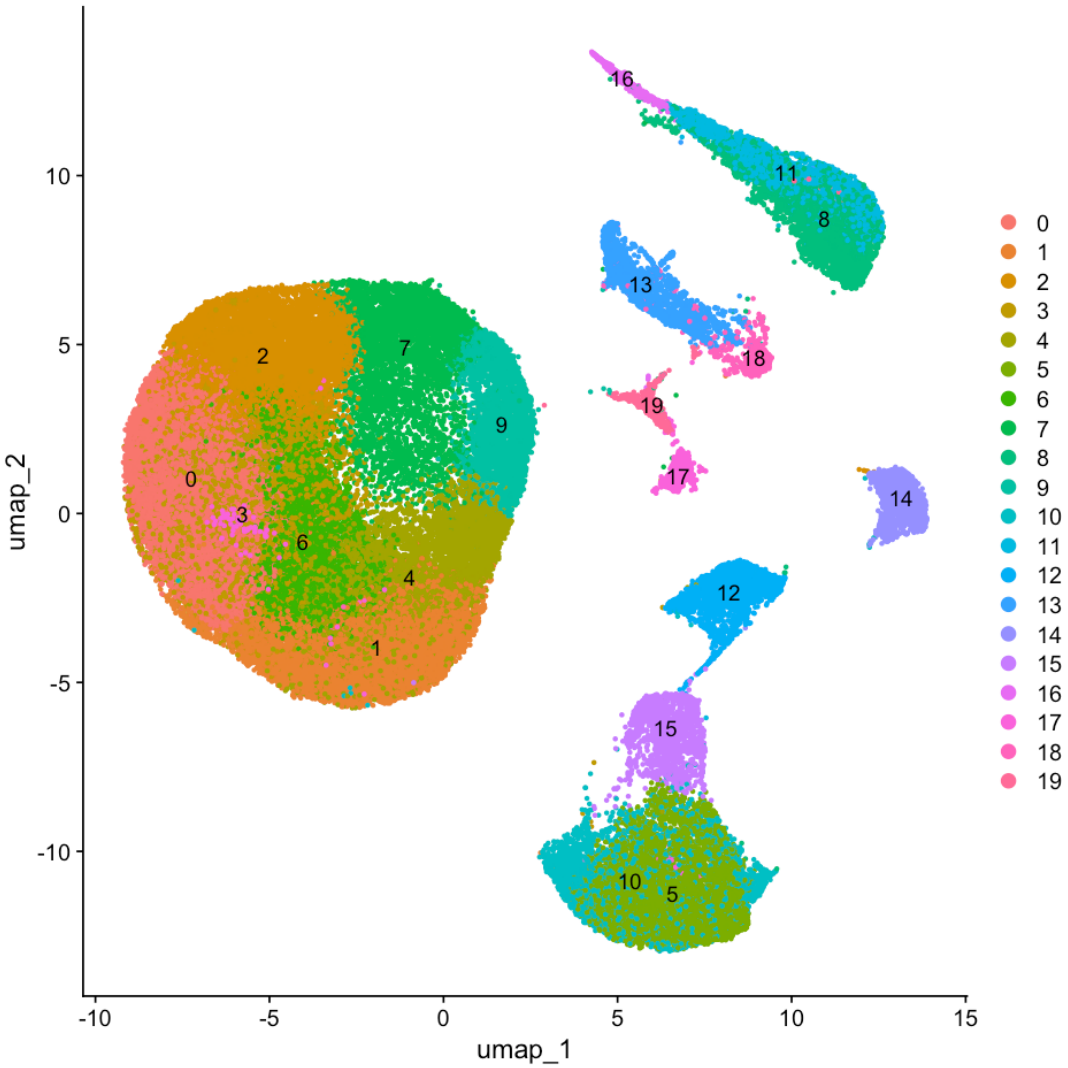

Figure B.3: Clustering with 10 PCs on the entire dataset

Appendix C Comparison of UMAP embeddings with different clustering resolutions.

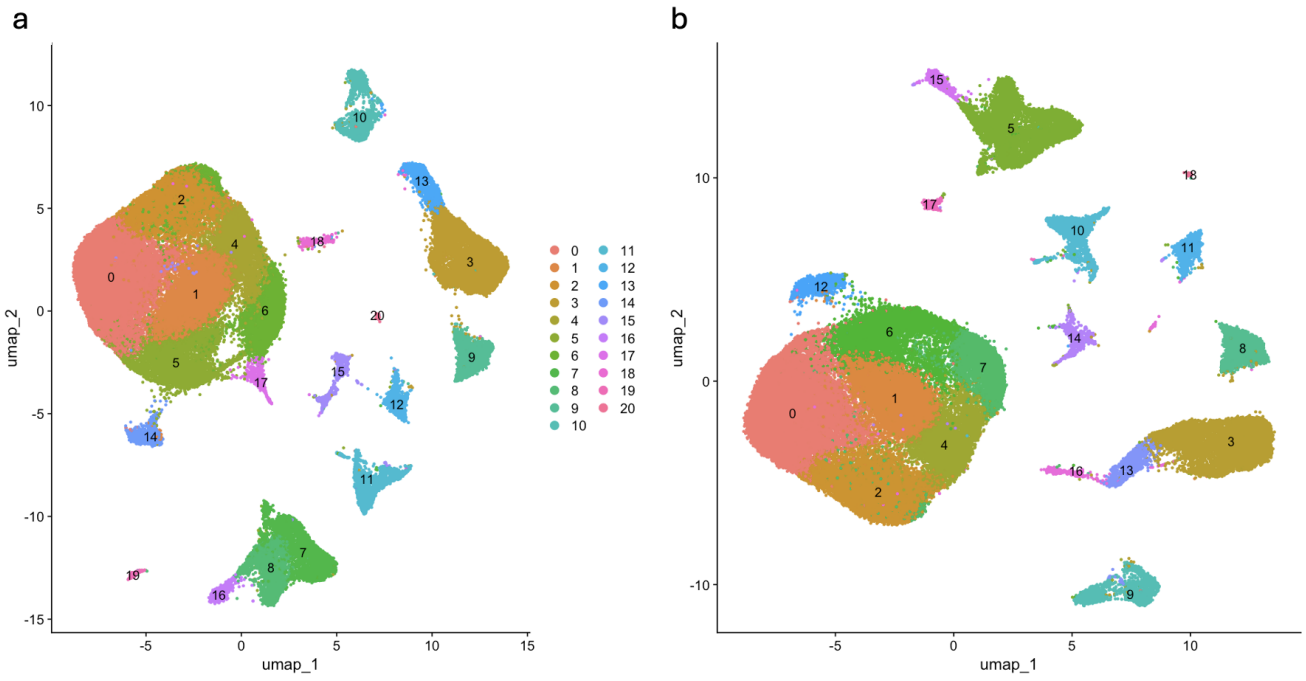

Figure C.4: a) Clustering analysis on a graph with k = 30 nearest neighbors, b) and k = 120 neighbors.

Appendix D Predicted labels with Uncertainty

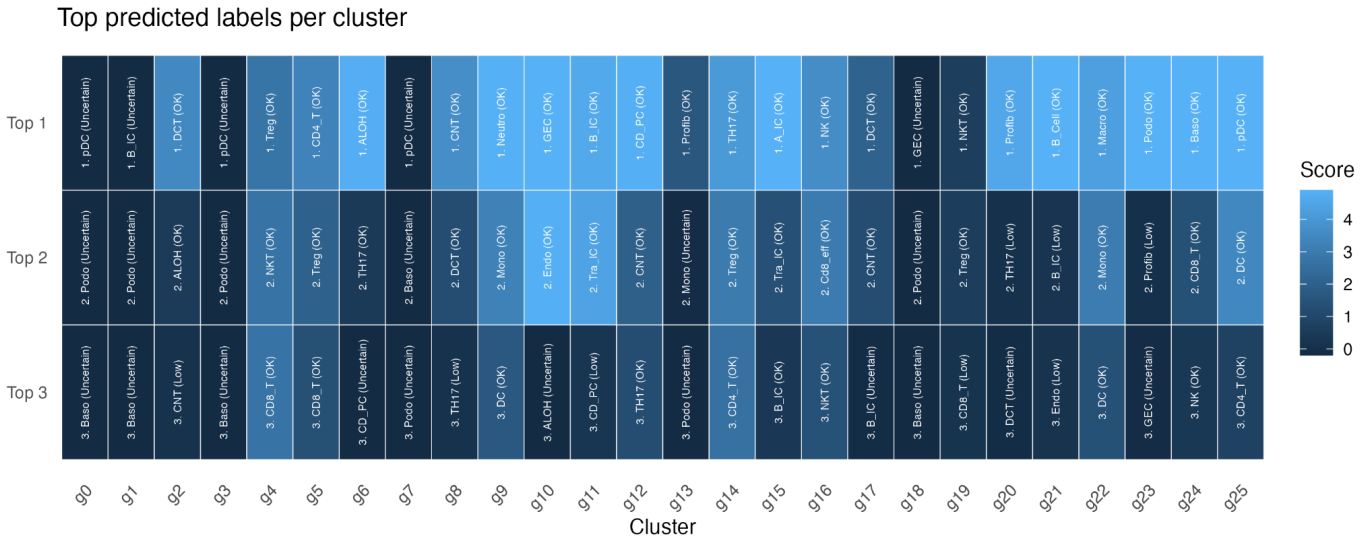

Figure D.5: Top three potential cluster label predicted for each cluster

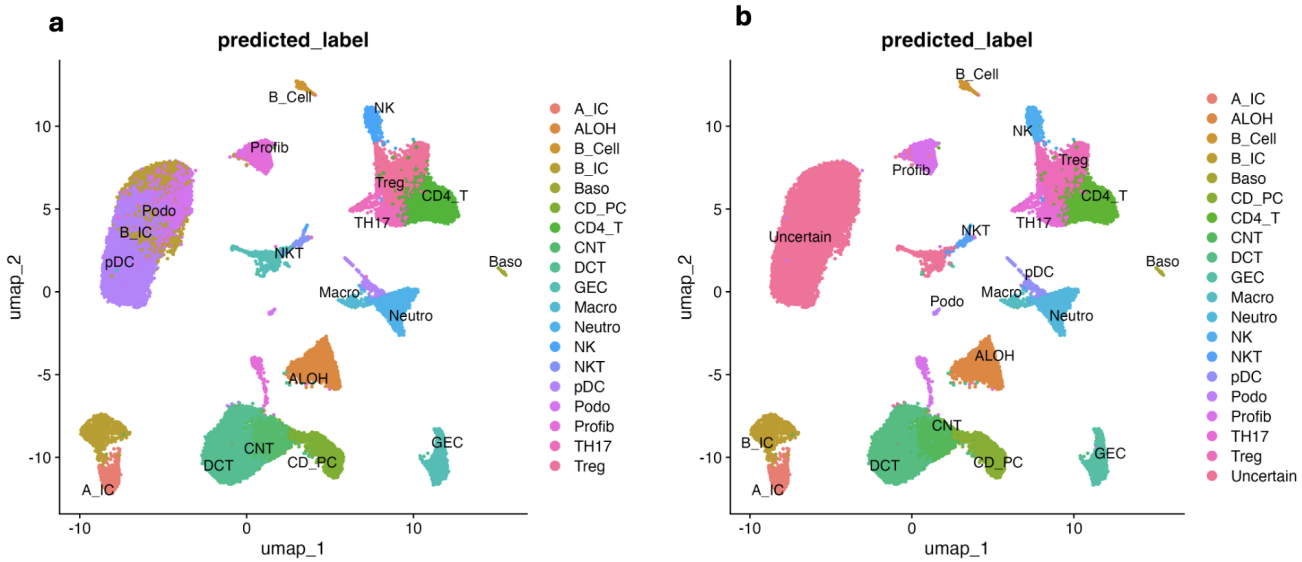

Figure D.6: Uncertainty in cluster label