METHOD ARTICLE

REVISED Unraveling the timeline of gene expression: A pseudotemporal trajectory analysis of single-cell RNA sequencing data [version 2; peer review: 1 approved, 2 approved with reservations]

Jinming Cheng [1,2], Gordon K. Smyth [1,3], Yunshun Chen [1,2,4]

[1]Bioinformatics Division, Walter and Eliza Hall Institute of Medical Research, Melbourne, Victoria, 3052, Australia
[2]Department of Medical Biology, The University of Melbourne, Melbourne, Victoria, 3052, Australia
[3]School of Mathematics and Statistics, The University of Melbourne, Melbourne, Victoria, 3052, Australia
[4]ACRF Cancer Biology and Stem Cells Division, Walter and Eliza Hall Institute of Medical Research, Melbourne, Victoria, 3052, Australia

v2 **First published:** 15 Jun 2023, **12**:684
https://doi.org/10.12688/f1000research.134078.1
Latest published: 10 Nov 2023, **12**:684
https://doi.org/10.12688/f1000research.134078.2

Abstract

Background

Single-cell RNA sequencing (scRNA-seq) technologies have rapidly developed in recent years. The droplet-based single cell platforms enable the profiling of gene expression in tens of thousands of cells per sample. The goal of a typical scRNA-seq analysis is to identify different cell subpopulations and their respective marker genes. Additionally, trajectory analysis can be used to infer the developmental or differentiation trajectories of cells.

Methods

This article demonstrates a comprehensive workflow for performing trajectory inference and time course analysis on a multi-sample single-cell RNA-seq experiment of the mouse mammary gland. The workflow uses open-source R software packages and covers all steps of the analysis pipeline, including quality control, doublet prediction, normalization, integration, dimension reduction, cell clustering, trajectory inference, and pseudo-bulk time course analysis. Sample integration and cell clustering follows the Seurat pipeline while the trajectory inference is conducted using the monocle3 package. The pseudo-bulk time course analysis uses the quasi-likelihood framework

Open Peer Review

Approval Status ✓ ? ?

	1	2	3
version 2 (revision) 10 Nov 2023	✓ view		
version 1 15 Jun 2023	? view	? view	? view

1. **Michael D Morgan**, University of Aberdeen, Aberdeen, UK
2. **Koen Van den berge**, Janssen R&D, Beerse, Belgium
3. **Anna Alemany**, Leiden University Medical Center, Leiden, Netherlands Antilles
Xuan Quy Nguyen, Leiden University Medical Center, Leiden, The Netherlands
Noëlle Dommann, Leiden University Medical Center, Leiden, The Netherlands

Any reports and responses or comments on the

of edgeR.

Results

Cells are ordered and positioned along a pseudotime trajectory that represented a biological process of cell differentiation and development. The study successfully identified genes that were significantly associated with pseudotime in the mouse mammary gland.

Conclusions

The demonstrated workflow provides a valuable resource for researchers conducting scRNA-seq analysis using open-source software packages. The study successfully demonstrated the usefulness of trajectory analysis for understanding the developmental or differentiation trajectories of cells. This analysis can be applied to various biological processes such as cell development or disease progression, and can help identify potential biomarkers or therapeutic targets.

Keywords
Single-cell RNA-seq, mammary gland, trajectory analysis, time course analysis, pseudo-bulk, differential expression analysis

article can be found at the end of the article.

This article is included in the Bioconductor gateway.

This article is included in the Genomics and Genetics gateway.

This article is included in the Bioinformatics gateway.

Corresponding authors: Gordon K. Smyth (smyth@wehi.edu.au), Yunshun Chen (yuchen@wehi.edu.au)

Author roles: Cheng J: Data Curation, Formal Analysis, Methodology, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Smyth GK**: Conceptualization, Funding Acquisition, Methodology, Project Administration, Software, Supervision, Writing – Review & Editing; **Chen Y**: Conceptualization, Formal Analysis, Funding Acquisition, Methodology, Project Administration, Software, Supervision, Visualization, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the Medical Research Future Fund (MRF1176199 to YC), a University of Melbourne Research Scholarship to JC, the National Health and Medical Research Council (Fellowship 1154970 to GKS) and the Chan Zuckerberg Initiative (2021-237445 to GKS and YC). The WEHI is supported by the Australian Government Independent Research Institutes Infrastructure Support Scheme and a Victorian State Government Operational Infrastructure Support Grant.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Cheng J, Smyth GK and Chen Y. **Unraveling the timeline of gene expression: A pseudotemporal trajectory analysis of single-cell RNA sequencing data [version 2; peer review: 1 approved, 2 approved with reservations]** F1000Research 2023, **12**:684 https://doi.org/10.12688/f1000research.134078.2

First published: 15 Jun 2023, **12**:684 https://doi.org/10.12688/f1000research.134078.1

REVISED **Amendments from Version 1**

In this revised version, we have incorporated additional information and provided more comprehensive explanations regarding the individual scRNA-seq analysis, the integration analysis, and the downstream edgeR time-course analysis. We have also revised the design matrix used in the downstream pseudo-bulk time-course analysis. In particular, we have introduced a sample effect into the design, which enables the incorporation of differences between the five biological samples. This substantially increases the statistical power of the time-course analysis, resulting in the detection of more genes significantly associated with pseudotime. Figure 6 has been split into two separate Figures (Figure 6 and 7). This is because users need to first generate Figure 6 to decide the starting node before they proceed to the cell ordering and pseudotime calculation. The visualization of pseudotime (Figure 7) comes after that. Consequently, all the Figure numbers after Figure 7 will change accordingly.

Any further responses from the reviewers can be found at the end of the article

Introduction

Single-cell RNA sequencing (scRNA-seq) has emerged as a popular technique for transcriptomic profiling of samples at the single-cell level. With droplet-based methods, thousands of cells can be sequenced in parallel using next-generation sequencing platforms.[1,2] One of the most widely used droplet-based scRNA-seq technologies is the 10x Genomics Chromium which enables profiling transcriptomes of tens of thousands of cells per sample.[3] A common goal of a scRNA-seq analysis is to investigate cell types and states in heterogeneous tissues. To achieve this, various pipelines have been developed, such as *Seurat*[4] and the Bioconductor's OSCA pipeline,[5] and *scanpy*.[6] A typical scRNA-seq data analysis pipeline involves quality control, normalization, dimension reduction, cell clustering, and differential expression analysis.

With the advent of single-cell multiplexing technologies, the per sample cost of scRNA-seq experiments has significantly decreased. This makes it feasible and more affordable to conduct single-cell RNA-seq profiling across a variety of biological samples within a given experimental study. In a multiple sample single-cell experiment, an integration method is required to investigate all cells across all samples simultaneously. This ensures that sample and batch effects are appropriately considered in visualizing and clustering cells. Popular integration methods include the Seurat's anchor-based integration method,[4] Harmony,[7] and the MNN.[8]

After integration and cell clustering, differential expression analysis is often performed to identify marker genes for each cell cluster. Various methods have been developed at the single-cell level for finding marker genes.[9,10] Recently, the pseudo-bulk method has become increasingly popular due to its superior computational efficiency and its ability to consider biological variation between replicate samples.[11] Under this approach, pseudo-bulk expression profiles are formed by aggregating read counts for all cells within the same group (e.g., cluster, cell type) and from the same sample.

Trajectory inference is another popular downstream analysis that aims to study cell differentiation or cell type development. Popular software tools to perform trajectory analysis include *monocle3*[12] and *slingshot*.[13] These methods learn trajectories based on the change of gene expression and order cells along a trajectory to obtain pseudotime.[14,15] This allows for pseudotime-based time course analysis in single-cell experiments, which is extremely useful for investigating specific biological questions of interest.

Here we present a new single-cell workflow that integrates trajectory analysis and pseudo-bulking to execute a single-cell pseudo time course analysis. The inputs for this workflow are single-cell count matrices, such as those generated by 10x Genomic's *cellranger*. The methods involved open source packages in R. The single-cell QC, clustering and integration analyses are performed in *Seurat*, whereas the trajectory analysis is conducted using *monocle3*. Once the pseudo-bulk samples are created and assigned pseudotime, a time course analysis is conducted in *edgeR*.[16] The analysis pipeline presented in this article can be used for examining dynamic cellular changes along a specific trajectory in any single-cell RNA-seq experiment with replicate samples.

Description of the biological experiment

The scRNA-seq data used to demonstrate this workflow consists of five mouse mammary epithelium samples at five different stages: embryonic, early postnatal, pre-puberty, puberty and adult. The puberty sample is from the study in Pal et al. 2017,[17] whereas the other samples are from Pal et al. 2021.[18] These studies examined the stage-specific single-cell profiles in order to gain insight into the early developmental stages of mammary gland epithelial lineage. The *cellranger* count matrix outputs of these five samples are available on the GEO repository as series GSE103275 and GSE164017.

Data preparation

Downloading the data

The *cellranger* output of each sample consists of three key files: a count matrix in `mtx.gz` format, barcode information in `tsv.gz` format and feature (or gene) information in `tsv.gz` format.

The outputs of the mouse mammary epithelium at embryonic stage (E18.5), post-natal 5 days (P5), 2.5 weeks (Pre-puberty), and 10 weeks (Adult) can be downloaded from GSE164017,[18] whereas the output of mouse mammary epithelium at 5 weeks (Puberty) can be downloaded from GSE103275.[17]

We first create a `data` directory to store all the data files.

```
> data_dir <- "data"
> if(!dir.exists(data_dir)){dir.create(data_dir, recursive=TRUE)}
```

We then download the barcode and count matrix files of the five samples.

```
> accessions <-c("GSM4994960","GSM4994962","GSM4994963","GSM2759554","GSM4994967")
> stages <- c("E18-ME", "Pre-D5-BL6", "Pre-BL6", "5wk-1", "Adult-BL6")
> file_suffixes <- c("barcodes.tsv.gz", "matrix.mtx.gz")
> for ( i in 1:length(accessions) ) {
+   for (file_suffix in file_suffixes) {
+     filename <- paste0(accessions[i],"_",stages[i],"-",file_suffix)
+     url <- paste0("http://www.ncbi.nlm.nih.gov/geo/download/?acc=",
+                   accessions[i],"&","format=file&","file=",filename)
+     download.file(url=url,destfile=paste0(data_dir,"/",filename))
+   }
+ }
```

Since the five samples in this workflow are from two separate studies and were processed using different *cellranger* references but built from the same mouse genome (mm10), the feature information is slightly different between the two runs. In general, the same *cellranger* reference build is preferred for the sake of consistency, although the effect on the downstream analysis is negligible. Here, we download the feature information of both runs. The `GSM2759554_5wk-1-genes.tsv.gz` file contains the feature information for the `5wk-1` sample, whereas `GSE164017_features.tsv.gz` contains the feature information for the other four samples.

```
> GSE <- c("GSE164017", "GSM2759554")
> feature_filenames <- c("GSE164017_features.tsv.gz",
+                        "GSM2759554_5wk-1-genes.tsv.gz")
> for (i in 1:length(GSE) ) {
+   url <- paste0("http://www.ncbi.nlm.nih.gov/geo/download/?acc=",
+                 GSE[i],"&","format=file&","file=",feature_filenames[i])
+   download.file(url=url,destfile=paste0(data_dir,"/",feature_filenames[i]))
+ }
```

A target information file is created to store all the sample and file information.

```
> samples <- c("E18.5-epi", "P5", "Pre-puberty", "Puberty", "Adult")
> targets <- data.frame(
+     samples=samples,
+     stages=stages,
+     accessions=accessions,
+     matrix.file = paste0("data/",accessions[1:5],"_",stages[1:5],"-","matrix.
                           mtx.gz"),
+     barcode.file = paste0("data/",accessions[1:5],"_",stages[1:5],"-","barcodes.
                           tsv.gz"),
+     feature.file = paste0("data/",feature_filenames[c(1,1,1,2,1)]))
> targets

      samples     stages accessions                                matrix.file
1   E18.5-epi     E18-ME GSM4994960     data/GSM4994960_E18-ME-matrix.mtx.gz
```

```
2          P5 Pre-D5-BL6 GSM4994962 data/GSM4994962_Pre-D5-BL6-matrix.mtx.gz
3 Pre-puberty    Pre-BL6 GSM4994963    data/GSM4994963_Pre-BL6-matrix.mtx.gz
4     Puberty      5wk-1 GSM2759554      data/GSM2759554_5wk-1-matrix.mtx.gz
5       Adult  Adult-BL6 GSM4994967  data/GSM4994967_Adult-BL6-matrix.mtx.gz
                              barcode.file                         feature.file
1     data/GSM4994960_E18-ME-barcodes.tsv.gz     data/GSE164017_features.tsv.gz
2 data/GSM4994962_Pre-D5-BL6-barcodes.tsv.gz     data/GSE164017_features.tsv.gz
3    data/GSM4994963_Pre-BL6-barcodes.tsv.gz     data/GSE164017_features.tsv.gz
4      data/GSM2759554_5wk-1-barcodes.tsv.gz data/GSM2759554_5wk-1-genes.tsv.gz
5  data/GSM4994967_Adult-BL6-barcodes.tsv.gz     data/GSE164017_features.tsv.gz
```

Reading in the data

The downloaded *cellranger* outputs of all the samples can be read in one-by-one using the `read10X` function in the *edgeR* package. First, a `DGElist` object is created for each sample, which is then consolidated into a single `DGElist` object by merging them altogether.

```
> library(edgeR)
> dge_all <- list()
> for ( i in 1:5 ) {
+   y <- read10X(mtx = targets$matrix.file[i],
+       barcodes = targets$barcode.file[i], genes = targets$feature.file[i])
+   y$samples$group <- targets$samples[i]
+   colnames(y) <- paste0(targets$accessions[i],"-",y$samples$Barcode)
+   y$genes$Ensembl_geneid <- rownames(y)
+   y$genes <- y$genes[,c("Ensembl_geneid","Symbol")]
+   y <- y[!duplicated(y$genes$Symbol),]
+   rownames(y) <- y$genes$Symbol
+   dge_all[[i]] <- y
+ }
> rm(y)
> common.genes <- Reduce(intersect, lapply(dge_all, rownames))
> for(i in 1:5) dge_all[[i]] <- dge_all[[i]][common.genes, ]
> dge_merged <- do.call("cbind", dge_all)
```

The levels of `group` in the sample information data frame are reordered and renamed from the early embryonic stage to the late adult stage.

```
> dge_merged$samples$group <- factor(dge_merged$samples$group, levels=samples)
```

The number of genes, the total number of cells, and the number of cells in each sample are shown below.

```
> dim(dge_merged)

[1] 26589 33735

> table(dge_merged$samples$group)

  E18.5-epi         P5 Pre-puberty      Puberty      Adult
       6969       3886        4183         5428      13269
```

Single-cell RNA-seq analysis

Quality control

Quality control (QC) is essential for single-cell RNA-seq data analysis. Common choices of QC metrics include number of expressed genes or features, total number of reads, and proportion of reads mapped to mitochondrial genes in each cell. The number of expressed genes and mitochondria read percentage in each cell can be calculated as follows.

```
> dge_merged$samples$num_exp_gene <- colSums(dge_merged$counts>0)
> mito_genes <- rownames(dge_merged)[grep("^mt-",rownames(dge_merged))]
```

```
> dge_merged$samples$mito_percentage <-
+   colSums(dge_merged$counts[mito_genes,])/
+   colSums(dge_merged$counts)*100
```

These QC metrics can be visualized in the following scatter plots (Figure 1).

```
> library(ggplot2)
> my_theme_ggplot <- theme_classic() +
+   theme(axis.text=element_text(size=12),
+         axis.title=element_text(size=15,face="bold"),
+         plot.title=element_text(size=15,face="bold",hjust=0.5),
+         plot.margin=margin(0.5, 0.5, 0.5, 0.5, "cm"))
> my_theme_facet <-
+   theme(strip.background=element_rect(colour="white",fill="white"),
+         strip.text=element_text(size=15, face="bold",color="black"))
> my_colors_15 <- c("cornflowerblue", "darkorchid1", "firebrick1", "gold",
+                   "greenyellow", "mediumspringgreen", "mediumturquoise",
+                   "orange1", "pink", "deeppink3", "violet", "magenta",
+                   "goldenrod4", "cyan", "gray90")
```

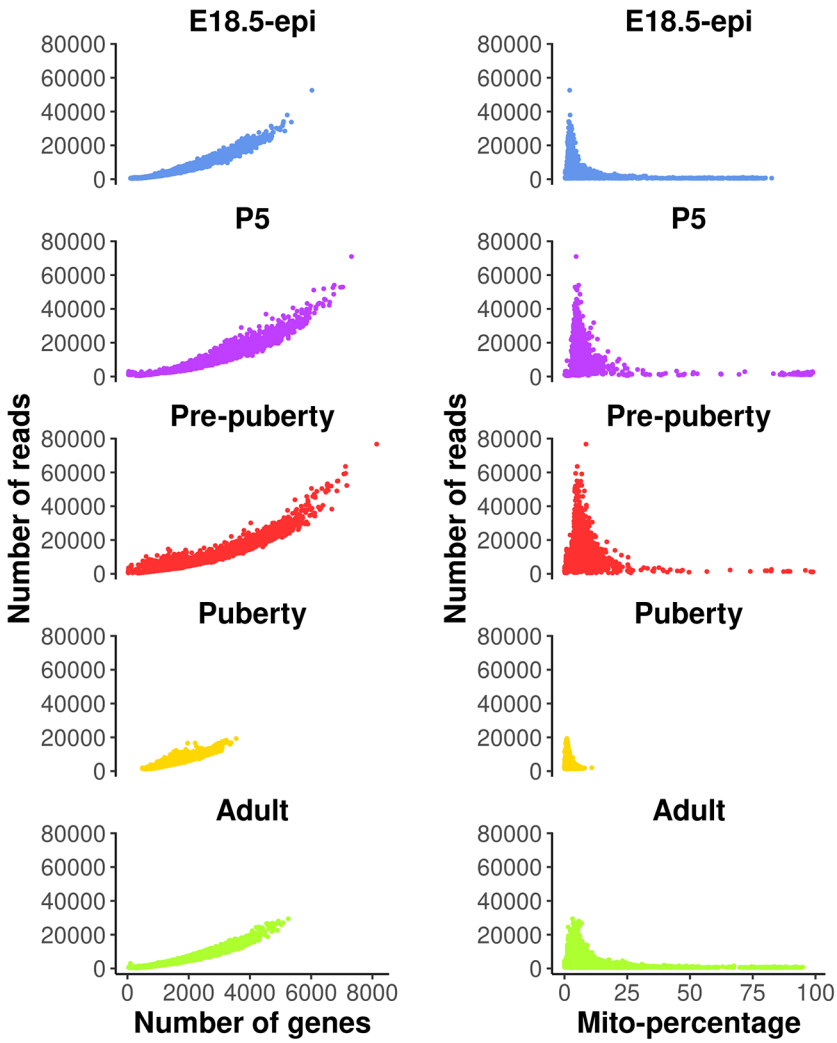

Figure 1. Scatter plots of quality control metrics across all the samples. Each dot represents a cell. The plots on the left show number of reads vs number of genes detected, whereas those on the right show number of reads vs mitochondria read percentage.

```
> p1 <- ggplot(data = dge_merged$samples,
+              aes(x=num_exp_gene, y=lib.size, color = group ) ) +
+       geom_point(size=0.5, show.legend=FALSE) +
+       facet_wrap(group~., ncol=1) +
+       scale_color_manual(values=my_colors_15 ) +
+       labs(x="Number of genes", y="Number of reads") +
+       my_theme_ggplot + my_theme_facet
> p2 <- ggplot(data = dge_merged$samples,
+              aes(x = mito_percentage, y=lib.size, color = group ) ) +
+       geom_point(size = 0.5, show.legend = FALSE) +
+       facet_wrap(group~., ncol=1) +
+       scale_color_manual(values=my_colors_15) +
+       labs(x="Mito-percentage", y="Number of reads") +
+       my_theme_ggplot + my_theme_facet
> patchwork::wrap_plots(p1, p2, ncol=2)
```

Cells with a very low number of genes (<500) are considered of low quality and hence are removed from the analysis. Cells with high mitochondria read percentage (>10%) are also removed as high expression level of mitochondrial genes indicate damaged or dead cells. In general, these QC thresholds are dependent on the study data and hence should be considered carefully. For example, quiescent cells may normally have low RNA expression levels, and metabolically active cells may have a high mitochondrial content. Cells expressing a large number of genes are also removed as they are likely to be doublets. Even though a separate doublet detection analysis is performed later on, we notice from our own practice that the combination of both doublet detection and the removal of cells with large counts works the best. Different thresholds are selected for different samples based on the distribution of the number of genes expressed. Here, we choose 5000, 6000, 6000, 3000, 4000 for E18.5-epi, P5, pre-puberty, puberty and adult samples, respectively. In this workflow, most of the single-cell analysis is conducted using the *Seurat* package. We begin by reading in the scRNA-seq data from the five samples together with an initial QC process. Specifically, we filter out genes expressed in fewer than 3 cells and cells expressing fewer than 200 genes for each sample. Then the abovementioned QC thresholds are applied in order to further remove cells of low quality in each sample. The data after QC are stored as a list of five Seurat objects.

```
> library(Seurat)
> n_genes_max <- c(5000, 6000, 6000, 3000, 4000)
> data_seurat <- list()
> for (i in 1:5) {
+   sel <- dge_merged$samples$group == samples[i]
+   y <- dge_merged[, sel]
+   data_seurat[[i]] <- CreateSeuratObject( counts=y$counts,
+       meta.data=y$samples, min.cells=3, min.features=200,
+       project=samples[i] )
+   data_seurat[[i]] <- subset( data_seurat[[i]],
+       subset = (nFeature_RNA > 500) & (nFeature_RNA < n_genes_max[i]) &
+                  (mito_percentage < 10) )
+ }
> names(data_seurat) <- samples
```

Standard Seurat analysis of individual sample

A standard Seurat analysis is performed for each individual sample. This would provide us some general information on how each individual sample looks like and what cell types present within them. More details on how to perform a scRNA-seq analysis can be found in Seurat online vignettes.

For each individual sample analysis, the default log normalization method in `NormalizeData` is applied to each sample. The top 2000 highly variable genes (HVGs) are identified by `FindVariableFeatures`. The normalized data of the 2000 highly variable genes are scaled by `ScaleData` to have a mean of 0 and a variance of 1. The principal component analysis (PCA) dimension reduction is performed on the highly variable genes by `RunPCA`. Uniform manifold approximation and projection (UMAP) dimension reduction is performed on the first 30 PCs by `RunUMAP`. Here we use the first 30 PCs to be consistent with the analysis in Pal *et al.* 2021.[18] Based on the Seurat vignette and our own practice, the number of PCs chosen would not change the results dramatically if it is large enough (> 10). Cell clustering is performed individually for each sample by `FindNeighbors` and `FindClusters`, which by default uses the Louvain algorithm. Cell clustering resolution is carefully chosen for each sample so that distinct cell types are grouped

into separate clusters. For this dataset, the cell clustering resolution is set at 0.1, 0.1, 0.2, 0.2 and 0.2 for E18.5-epi, P5, pre-puberty, puberty and adult, respectively.

```
> data_seurat <- lapply(data_seurat, NormalizeData)
> data_seurat <- lapply(data_seurat, FindVariableFeatures, nfeatures=2000)
> data_seurat <- lapply(data_seurat, ScaleData)
> data_seurat <- lapply(data_seurat, RunPCA, verbose = FALSE)
> data_seurat <- lapply(data_seurat, RunUMAP, reduction = "pca", dims = 1:30)
> data_seurat <- lapply(data_seurat, FindNeighbors, reduction="pca", dims=1:30)
> resolutions <- c(0.1, 0.1, 0.2, 0.2, 0.2)
> for(i in 1:5)
+     data_seurat[[i]] <- FindClusters(data_seurat[[i]],
+         resolution=resolutions[i], verbose=FALSE)
```

Removing potential doublets and non-epithelial cells

Although high-throughput droplet-based single-cell technologies can accurately capture individual cells, there are instances where a single droplet may contain two or more cells, which are known as doublets or multiplets. Here we use the *scDblFinder* package[19] to further remove potential doublets. To do that, each Seurat object in the list is first converted into a `SingleCellExperiment` object using the `as.SingleCellExperiment` function in *Seurat.* Then the `scDblFinder` function in the *scDblFinder* package is called to predict potential doublets on each `SingleCellExperiment` object. The `scDblFinder` output for each sample is stored in the corresponding Seurat object.

```
> library(scDblFinder)
> for (i in 1:5) {
+   sce <- as.SingleCellExperiment(DietSeurat(data_seurat[[i]],
+       graphs=c("pca","umap")) )
+   set.seed(42)
+   sce <- scDblFinder(sce)
+   data_seurat[[i]]$db_score <- sce$scDblFinder.score
+   data_seurat[[i]]$db_type <- factor( sce$scDblFinder.class,
+       levels=c("singlet", "doublet") )
+ }
```

The main object of this single-cell experiment is to examine the early developmental stages of the mouse epithelial mammary gland. Hence, for the rest of the analysis we will mainly focus on the epithelial cell population which is typically marked by the *Epcam* gene. The cell clustering, the expression level of *Epcam* and doublet prediction results of each sample are shown below (Figure 2).

```
> p1 <- lapply(data_seurat,function(x){DimPlot(x, pt.size=0.1, cols=my_colors_15) +
+                 ggtitle(x$group[1]) + theme(plot.title=element_text(hjust=0.5))})
> p2 <- lapply(data_seurat, FeaturePlot, feature="Epcam", pt.size=0.1)
> p3 <- lapply(data_seurat, DimPlot, group.by="db_type", pt.size=0.1,
+              cols=c("gray90", "firebrick1"))
> patchwork::wrap_plots(c(p1,p2,p3), nrow=5, byrow=FALSE)
```

By examining the expression level of the *Epcam* gene, together with some other known marker genes of basal, LP and ML, we select the following clusters in each sample as the epithelial cell population.

```
> epi_clusters <- list(
+   "E18.5-epi" = 0,
+   "P5" = c(1,3),
+   "Pre-puberty" = c(0:2, 5),
+   "Puberty" = 0:6,
+   "Adult" = 0:3
+ )
```

Cells that are non-epithelial and those identified as potential doublets by *scDblFinder* are excluded from the subsequent analysis. The cellular barcodes of the remaining epithelial cells from each sample are stored in the list object called `epi_cells`. The respective number of epithelial cells that are retained for each sample is shown below.

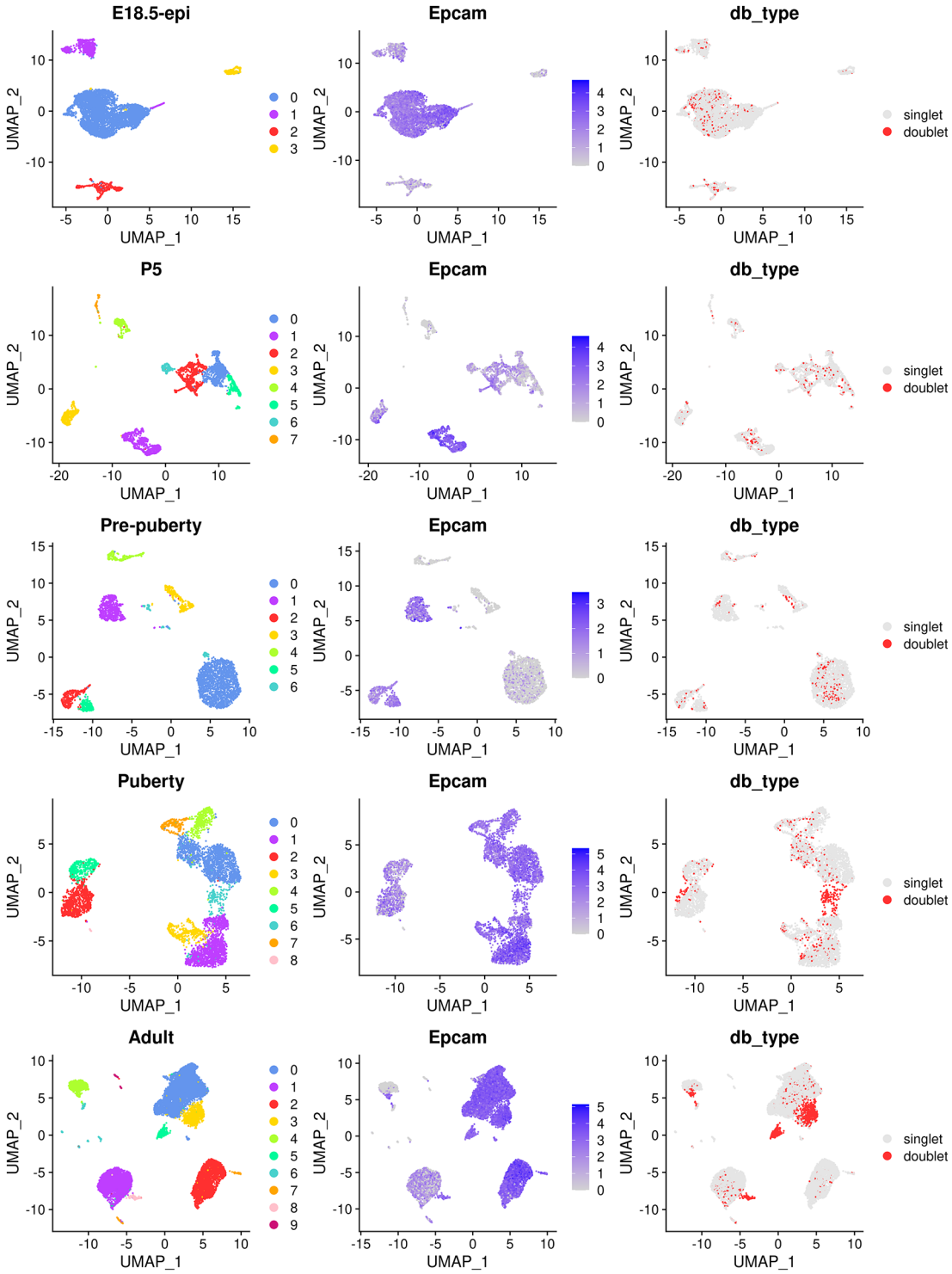

Figure 2. UMAP visualization of each individual samples. The UMAP plots, in sequence from the top row to the bottom row, correspond to E18.5-epi, P5, Pre-puberty, Puberty, and Adult, respectively. In each row, cells are coloured by cluster on the left, by Epcam expression level in the middle, and by doublet prediction on the right.

```
> epi_cells <- list()
> for (i in samples) {
+   epi_cells[[i]] <- rownames(
+     subset(data_seurat[[i]]@meta.data,
+       (db_type == "singlet") & (seurat_clusters %in% epi_clusters[[i]])))
+ }
> do.call(c, lapply(epi_cells, length))

  E18.5-epi          P5 Pre-puberty      Puberty      Adult
       4343        1140        2546         4706       9341
```

Data integration

Integrating epithelial cells of five samples

Since we have five individual scRNA-seq samples, conducting an integration analysis is necessary to explore all cells across these samples simultaneously. In this workflow, we use the default anchor-based integration method of the *Seurat* package. Depending on the single-cell analysis workflow, users are free to use other integration methods they may prefer (e.g., Harmony and MNN). A Seurat object is first created from the merged `DGEList` object of epithelial cells using `CreateSeuratObject` function without filtering any cells (`min.features` is set to 0). Lowly expressed genes are removed as they are not of any biological interest here. Here we keep genes expressed in at least 3 cells in each sample (`min.cells` is set to 3) although different thresholds can be adopted in general depending on the data.

```
> epi_cells <- do.call(c, epi_cells)
> dge_merged_epi <- dge_merged[, epi_cells]
> seurat_merged <- CreateSeuratObject(counts = dge_merged_epi$counts,
+                       meta.data = dge_merged_epi$samples,
+                       min.cells = 3, min.features = 0, project = "mammary_epi")
```

Then the Seurat object is split into a list of five Seurat objects, where each object corresponds to one of the five samples. For each sample, the log normalization method is applied to normalize the raw count by `NormalizeData`, and highly variable genes are identified by `FindVariableFeatures`.

```
> seurat_epi <- SplitObject(seurat_merged, split.by = "group")
> seurat_epi <- lapply(seurat_epi, NormalizeData)
> seurat_epi <- lapply(seurat_epi, FindVariableFeatures, nfeatures = 2000)
```

The feature genes used for integration are chosen by `SelectIntegrationFeatures`, and these genes are used to identify anchors for integration by `FindIntegrationAnchors`. The integration process is performed by `IntegrateData` based on the identified anchors. Please note that the integration step is computationally intensive and might take a substantial amount of time to complete (20-40 minutes depending on the computational resource).

```
> anchor_features <- SelectIntegrationFeatures(seurat_epi,
+                       nfeatures = 2000, verbose = FALSE)
> anchors <- FindIntegrationAnchors(seurat_epi, verbose = FALSE,
+                       anchor.features = anchor_features)
> seurat_int <- IntegrateData(anchors, verbose = FALSE)
```

The integrated data are then scaled to have a mean of 0 and a variance of 1 by `ScaleData`. PCA is performed on the scaled data using `RunPCA`, followed by UMAP using `RunUMAP`. Same as before, we use 30 PCs for the sake of consistency and the results would not change dramatically provided a good amount of PCs (>10) are used. Cell clusters of the integrated data are identified by using `FindNeighbors` and `FindClusters`. We choose 0.2 as the cell clustering resolution after experimenting with different resolution parameters. This is because under this resolution the three major epithelial subpopulations, two intermediate cell clusters, and a small group of stroma cells can be clearly separated in distinct cell clusters.

```
> DefaultAssay(seurat_int) <- "integrated"
> seurat_int <- ScaleData(seurat_int, verbose = FALSE)
> seurat_int <- RunPCA(seurat_int, npcs = 30, verbose = FALSE)
> seurat_int <- RunUMAP(seurat_int, reduction = "pca",
+                   dims = 1:30, verbose = FALSE)
> seurat_int <- FindNeighbors(seurat_int, dims = 1:30, verbose = FALSE)
> seurat_int <- FindClusters(seurat_int, resolution = 0.2, verbose = FALSE)
```

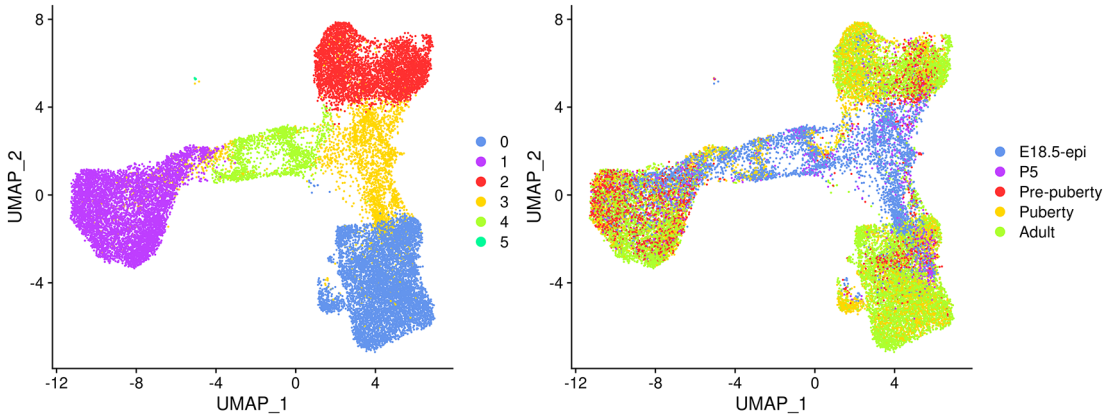

Figure 3. UMAP visualization of the integrated data. Cells are coloured by cluster on the left and by original sample on the right.

UMAP plots are generated to visualize the integration and cell clustering results (Figure 3). The UMAP plot indicates the presence of three major cell clusters (cluster 0, 1, and 2), which are bridged by intermediate clusters located in between them. Cells at the later stages largely dominate the three major cell clusters, while cells at the earlier stages are predominantly present in the intermediate clusters in the middle.

```
> seurat_int$group <- factor(seurat_int$group, levels = samples)
> p1 <- DimPlot(seurat_int, pt.size = 0.1, cols = my_colors_15)
> p2 <- DimPlot(seurat_int, pt.size = 0.1, group.by = "group",
+           shuffle = TRUE, cols = my_colors_15) + labs(title="")
> p1 | p2
```

Cell type identification

The mammary gland epithelium consists of three major cell types: basal myoepithelial cells, luminal progenitor (LP) cells and mature luminal (ML) cells. These three major epithelial cell populations have been well studied in the literature. By examining the classic marker genes of the three cell types, we are able to identify basal, LP and ML cell populations in the integrated data (Figure 4). Here we use *Krt14* and *Acta2* for basal, *Csn3* and *Elf5* for LP, and *Prlr* and *Areg* for ML. We also examine the expression level of *Hmgb2* and *Mki67* as they are typical markers for cycling cells and the expression level of *Igfbp7* and *Fabp4* as they are marker genes for stromal cells.

```
> markers <- c("Krt14", "Acta2", "Csn3","Elf5", "Prlr","Areg",
+              "Hmgb2", "Mki67", "Igfbp7","Fabp4")
> DefaultAssay(seurat_int) <- "RNA"
> FeaturePlot(seurat_int, order = TRUE, pt.size = 0.1, features = markers, ncol = 2)
```

Based on the feature plots, cluster 1, cluster 2 and cluster 0 represent the basal, LP and ML cell populations, respectively. Cluster 4 mainly consists of cycling cells, whereas cluster 3 seems to be a luminal intermediate cell cluster expressing both LP and ML markers. Cluster 5 consists of a few non-epithelial (stromal) cells that have not been filtered out previously.

The number of cells in each cluster for each sample is shown below.

```
> tab_number <- table(seurat_int$group, seurat_int$seurat_clusters)
> tab_number

               0    1    2    3    4   5
  E18.5-epi  171  878   29 2341  921   3
  P5         272  347   47  351  120   3
  Pre-puberty 381 1590  482   64   23   6
  Puberty   1894  986 1535   20  265   6
  Adult     4281 2495 2362  171   32   0
```

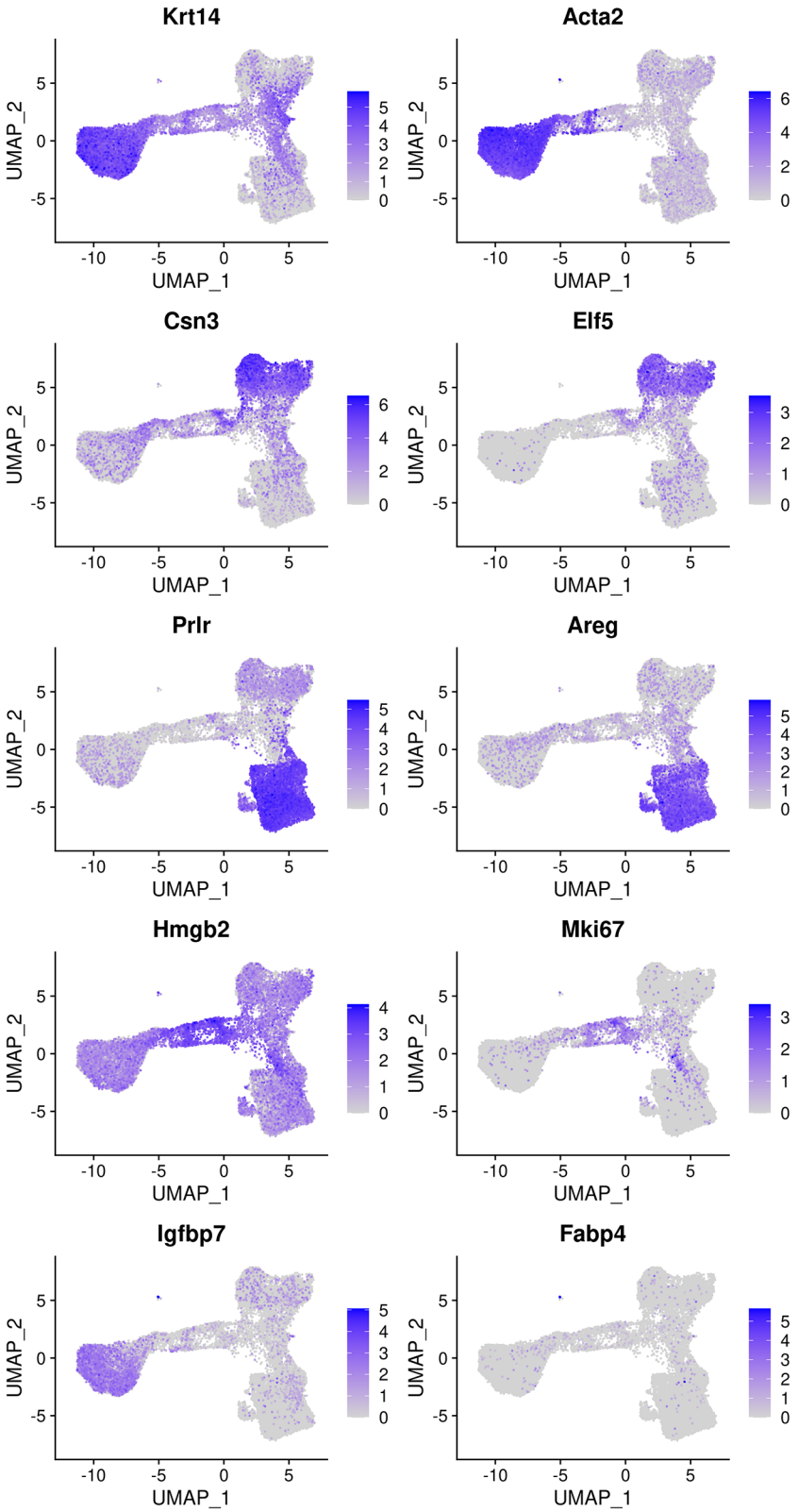

Figure 4. Feature plots of the integrated data. Genes from the top row to the bottom rows are the markers of basal, LP, ML, cycling, and stromal cells, respectively.

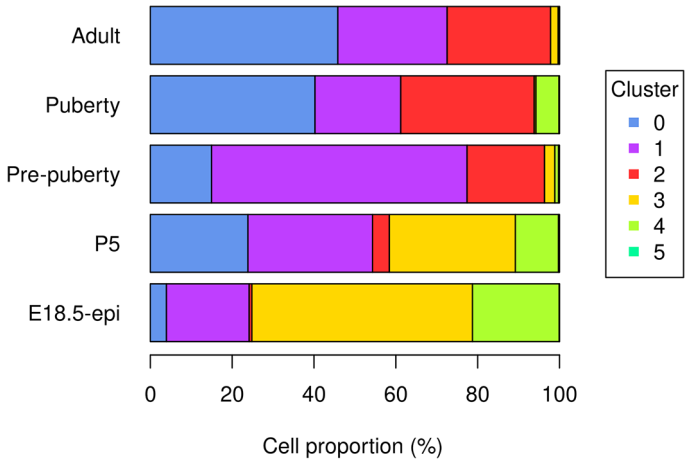

Figure 5. Bar plot of cell proportion of each cluster in each sample.

The proportion of cells in each cluster is calculated for each sample to compare the variation in cell composition across different stages.

```
> tab_ratio <- round(100*tab_number/rowSums(tab_number), 2)
> tab_ratio <- as.data.frame.matrix(tab_ratio)
> tab_ratio

                0    1     2     3     4    5
E18.5-epi    3.94 20.2  0.67 53.90 21.21 0.07
P5          23.86 30.4  4.12 30.79 10.53 0.26
Pre-puberty 14.96 62.5 18.93  2.51  0.90 0.24
Puberty     40.25 20.9 32.62  0.42  5.63 0.13
Adult       45.83 26.7 25.29  1.83  0.34 0.00
```

The bar plot (Figure 5) shows the proportion of different cell types in samples at different developmental stages. Specifically, the proportion of basal cells (purple) demonstrates an ascending trend from E18.5 to pre-puberty stage, after which it declines towards adult stage. The LP cell proportion (red) rises from E18.5 to puberty stage, followed by a slight dip at adult stage. Although the proportion of ML cells (blue) is higher at P5 than pre-puberty stage, it shows an overall increasing trend. Cycling cells (green) constitute the highest proportion at E18.5 stage, but decrease to a smaller proportion at pre-puberty stage, with a slight increase at puberty stage, and subsequently, they reduce to a negligible proportion at adult stage. The augmented cycling cell proportion at puberty stage aligns with the ductal morphogenesis characteristics of the mammary gland. The luminal intermediate cell proportion (yellow) displays a decreasing trend from E18.5 stage to adult stage.

```
> par(mar=c(5, 7, 1, 7), xpd=TRUE)
> barplot(t(tab_ratio), col=my_colors_15, xlab="Cell proportion (%)",
+     horiz = TRUE, las=1)
> legend("right", inset = c(-0.3,0), legend = 0:5, pch = 15,
+     col=my_colors_15, title="Cluster")
```

Trajectory analysis with monocle3

Constructing trajectories and pseudotime

Many biological processes manifest as a dynamic sequence of alterations in the cellular state, which can be estimated through a "trajectory" analysis. Such analysis is instrumental in detecting the shifts between different cell identities and modeling gene expression dynamics. By treating single-cell data as a snapshot of an uninterrupted process, the analysis establishes the sequence of cellular states that forms the process trajectory. The arrangement of cells along these trajectories can be interpreted as pseudotime.

Here, we use the *monocle3* package to infer the development trajectory in the mouse mammary gland epithelial cell population. The Seurat object of the integrated data is first converted into a `cell_data_set` object to be used in *monocle3*.

```
> library(monocle3)
> cds_obj <- SeuratWrappers::as.cell_data_set(seurat_int)
```

monocle3 re-clusters cells to assign them to specific clusters and partitions, which are subsequently leveraged to construct trajectories. If multiple partitions are used, each partition will represent a distinct trajectory. The calculation of pseudotime, which indicates the distance between a cell and the starting cell in a trajectory, is conducted during the trajectory learning process. These are done using the `cluster_cells` and `learn_graph` functions. To obtain a single trajectory and avoid a loop structure, both `use_partition` and `close_loop` are turned off in `learn_graph`.

```
> set.seed(42)
> cds_obj <- cluster_cells(cds_obj)
> cds_obj <- learn_graph(cds_obj, use_partition=FALSE, close_loop=FALSE)
```

Visualizing trajectories and pseudotime

The `plot_cells` function of *monocle3* is used to generate a trajectory plot that superimposes the trajectory information onto the UMAP representation of the integrated data. By adjusting the `label_principal_points` parameter, the names of roots, leaves, and branch points can be displayed. Cells in the trajectory UMAP plot (Figure 6) are coloured by cell cluster identified in the previous Seurat integration analysis.

```
> p1 <- plot_cells(cds_obj, color_cells_by="seurat_clusters",
+                  group_label_size=4, graph_label_size=3,
+                  label_cell_groups=FALSE, label_principal_points=TRUE,
+                  label_groups_by_cluster=FALSE) +
+       scale_color_manual(values = my_colors_15)
> p1
```

Along the *monocle3* trajectory analysis, several nodes are identified and marked with black circular dots on the resulting plot, representing principal nodes along the trajectories. To establish the order of cells and calculate their corresponding pseudotime, it is necessary to select a starting node among the identified principal nodes. For this analysis, node `"Y_65"` in the basal population (cluster 1) was selected as the starting node, as mammary stem cells are known to be enriched in the basal population and give rise to LP and ML cells in the epithelial lineage.[20] It should be noted that node numbers may vary depending on the version of monocle3 used.

```
> cds_obj <- order_cells(cds_obj, root_pr_nodes="Y_65")
```

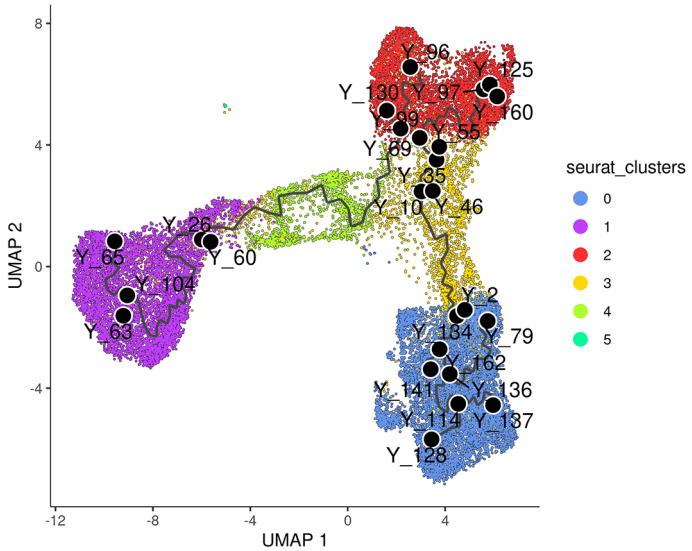

Figure 6. UMAP visualization of trajectory inferred by monocle3. Cells are coloured by cluster.

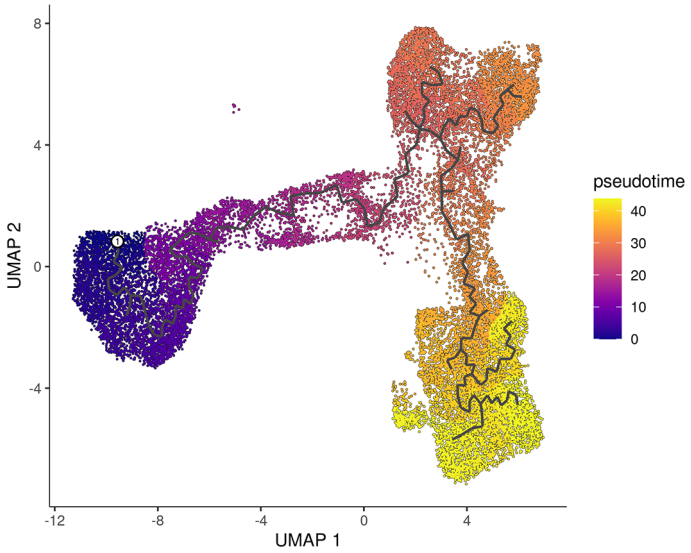

Figure 7. UMAP visualization of pseudotime computed by monocle3. Cells are coloured by pseudotime.

The cells are then ordered and assigned pseudotime values by the `order_cells` function in *monocle3*. The resulting pseudotime information can be visualized on the UMAP plot by using the `plot_cells` function, as demonstrated in the UMAP plot (Figure 7).

```
> p2 <- plot_cells(cds_obj, color_cells_by="pseudotime",
+                  label_groups_by_cluster=FALSE, label_leaves=FALSE,
+                  label_branch_points=FALSE)
> p2
```

The `pseudotime` function in *monocle3* allows users to extract the pseudotime values of the cells from a `cell_data_set` object. This information can then be stored in the metadata of the Seurat object for further analysis.

```
> seurat_int$pseudotime <- pseudotime(cds_obj)
```

Pseudo-bulk time course analysis with edgeR

Constructing pseudo-bulk profiles

After obtaining the pseudotime of each cell, we proceed to a time course analysis to identify genes that change significantly along the pseudotime. Our approach involves creating pseudo-bulk samples using a pseudo-bulking approach and performing an *edgeR*-style time course analysis.

To create the pseudo-bulk samples, read counts are aggregated for all cells with the same combination of sample and cluster. The number of cells used to construct each pseudo-bulk sample is added to the sample metadata. For simplicity, the average pseudotime of all cells in each pseudo-bulk sample is used as the pseudotime for that sample. One could also use the median of the cellwise pseudotime instead of the mean, but the results will not change dramatically.

```
> y <- dge_merged[, colnames(seurat_int)]
> y$samples <- cbind(y$samples[, 1:3],
+   seurat_int@meta.data[, c("seurat_clusters", "pseudotime")])
> sample_cluster <- paste0(y$samples$group, "_C", y$samples$seurat_clusters)
> avg_pseudotime <- tapply(y$samples$pseudotime, sample_cluster, mean)
> cell_number <- table(sample_cluster)
> y <- sumTechReps(y, ID = sample_cluster)
> y$samples$pseudotime <- avg_pseudotime[colnames(y)]
> y$samples$cell_number <- cell_number[colnames(y)]
```

The Entrez gene IDs are added to the gene information. Genes with no valid Entrez gene IDs are removed from the downstream analysis.

```
> library(org.Mm.eg.db)
> entrez_id <- select(org.Mm.eg.db, keys = y$genes$Symbol,
+                     columns = c("ENTREZID", "SYMBOL"), keytype = "SYMBOL")
> y$genes$ENTREZID <- entrez_id$ENTREZID
> y <- y[!is.na(y$genes$ENTREZID), ]
```

The samples are ordered by average pseudotime for the following analysis.

```
> y <- y[, order(y$samples$pseudotime)]
```

Filtering and normalization

We now proceed to the standard *edgeR* analysis pipeline, which starts with filtering and normalization. The sample information, such as library sizes, average pseudotime and cell numbers, are shown below.

```
> y$samples[, c("lib.size", "pseudotime", "cell_number")]

               lib.size pseudotime cell_number
Pre-puberty_C1 11886898       4.65        1590
Adult_C1        9285265       4.77        2495
P5_C1           2680089       6.41         347
Puberty_C1      3112796       6.48         986
E18.5-epi_C1    8084434      10.16         878
E18.5-epi_C5       5179      15.61           3
P5_C5             24491      15.61           3
Pre-puberty_C5    57834      15.61           6
Puberty_C5        12278      15.61           6
Adult_C4         204212      19.25          32
E18.5-epi_C4   11725731      19.31         921
P5_C4           1917228      19.68         120
Puberty_C4      1860770      19.72         265
Pre-puberty_C4   289370      22.10          23
E18.5-epi_C3   15806768      28.03        2341
Puberty_C2      5841364      28.62        1535
P5_C3           3862152      28.94         351
E18.5-epi_C2     167242      29.14          29
Adult_C2        8320630      29.66        2362
P5_C2            347365      29.67          47
Pre-puberty_C2  4625160      30.51         482
Puberty_C3        63722      31.73          20
Pre-puberty_C3  1432336      32.64          64
Adult_C3         997670      33.99         171
Pre-puberty_C0  6024872      38.90         381
E18.5-epi_C0     990670      39.64         171
Adult_C0       27621462      40.44        4281
P5_C0           3113053      40.68         272
Puberty_C0      8806924      41.09        1894
```

To ensure the reliability of the analysis, it is recommended to remove pseudo-bulk samples that are constructed from a small number of cells. We suggest each pseudo-bulk sample should contain at least 30 cells. In this analysis, we identified seven pseudo-bulk samples that were constructed with less than 30 cells and removed them form the analysis.

```
> keep_samples <- y$samples$cell_number > 30
> y <- y[, keep_samples]
```

Genes with very low count number are also removed from the analysis. This is performed by the `filterByExpr` function in *edgeR*.

```
> keep_genes <- filterByExpr(y)
> y <- y[keep_genes, , keep.lib.sizes=FALSE]
```

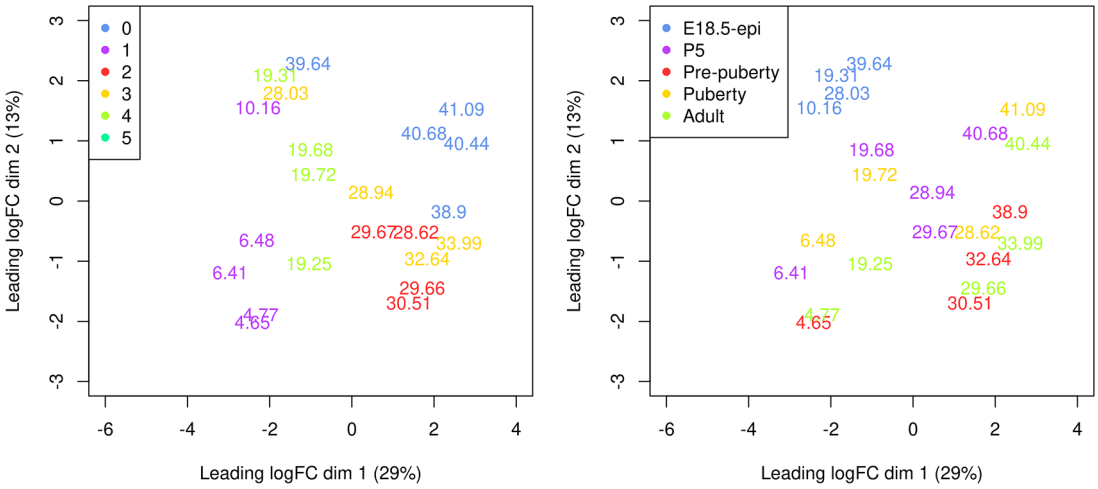

Figure 8. Multi-dimensional scaling (MDS) plot of the pseudo-bulk samples labelled by pseudotime. Samples are coloured by original cell cluster on the left and by developmental stage on the right.

The number of genes and samples after filtering are shown below.

```
> dim(y)
[1] 11550    22
```

Normalization is performed by the trimmed mean of M values (TMM) method[21] implemented in the `calcNormFactors` function in *edgeR*.

```
> y <- calcNormFactors(y)
```

A Multi-dimensional scaling (MDS) plot serves as a valuable diagnostic tool for investigating the relationship among samples. MDS plots are produced using the `plotMDS` function in *edgeR* (Figure 8).

```
> par(mar = c(5.1, 5.1, 2.1, 2.1), mfrow=c(1,2))
> cluster <- y$samples$seurat_clusters
> group <- y$samples$group
> plotMDS(y, labels = round(y$samples$pseudotime, 2),
+     xlim=c(-6,4), ylim=c(-3,3), col=my_colors_15[cluster])
> legend("topleft", legend=levels(cluster), col=my_colors_15, pch=16)
> plotMDS(y, labels = round(y$samples$pseudotime, 2),
+     xlim=c(-6,4), ylim=c(-3,3), col=my_colors_15[group])
> legend("topleft", legend=levels(group), col=my_colors_15, pch=16)
```

On the MDS plot, pseudo-bulk samples derived from the same cell cluster are close to each other. The samples are positioned in ascending order of pseudotime from left to right, suggesting a continuous shift in the gene expression profile throughout the pseudotime.

Design matrix

The aim of a time course experiment is to examine the relationship between gene abundances and time points. Assuming gene expression changes smoothly over time, we use a natural cubic spline with degrees of freedom of 3 to model gene expression along the pseudotime. In general, any degrees of freedom in range of 3 to 5 is reasonable provided there are sufficient time points for the degrees of freedom of the residuals.

The spline design matrix is generated by `ns` function in *splines*. The three spline coefficients of the design matrix (i.e., Z1, Z2 and Z3) do not have any particular meaning in general. However, we can re-parametrize the design matrix using QR decomposition so that the first coefficient Z1 represents the linear trend in pseudotime.

```
> t1 <- y$samples$pseudotime
> X <- splines::ns(as.numeric(t1),df = 3)
```

```
> A <- cbind(1,t1,X)
> QR <- qr(A)
> r <- QR$rank
> R_rank <- QR$qr[1:r,1:r]
> Z <- t(backsolve(R_rank,t(A),transpose=TRUE))
> Z <- Z[,-1]
> design <- model.matrix(~ Z)
```

Since the five samples are from different timepoints, the pseudo-bulk samples derived from these five samples are not independent replicates. The sample effect at the pseudo-bulk level can also be seen from the MDS plot (Figure 8 right). Hence, we add the sample effect to the design in addition to the re-parametrized spline coefficients. The full design matrix is shown below.

```
> group <- y$samples$group
> design <- model.matrix(~ Z + group)
> colnames(design) <- gsub("group", "", colnames(design))
> design
   (Intercept)      Z1      Z2      Z3 P5 Pre-puberty Puberty Adult
1            1 -0.3593 -0.0550 -0.3206  0           1       0     0
2            1 -0.3572 -0.0572 -0.3116  0           0       0     1
3            1 -0.3285 -0.0887 -0.1837  1           0       0     0
4            1 -0.3271 -0.0901 -0.1780  0           0       1     0
5            1 -0.2626 -0.1489  0.0887  0           0       0     0
6            1 -0.1034 -0.0918  0.4047  0           0       0     1
7            1 -0.1024 -0.0900  0.4042  0           0       0     0
8            1 -0.0958 -0.0775  0.4002  1           0       0     0
9            1 -0.0951 -0.0761  0.3997  0           0       1     0
10           1  0.0505  0.2625  0.0541  0           0       0     0
11           1  0.0609  0.2746  0.0262  0           0       1     0
12           1  0.0666  0.2796  0.0118  1           0       0     0
13           1  0.0790  0.2862 -0.0180  0           0       0     1
14           1  0.0792  0.2863 -0.0185  1           0       0     0
15           1  0.0940  0.2854 -0.0491  0           1       0     0
16           1  0.1313  0.2389 -0.1021  0           1       0     0
17           1  0.1551  0.1804 -0.1195  0           0       0     1
18           1  0.2410 -0.1584 -0.1106  0           1       0     0
19           1  0.2540 -0.2199 -0.1034  0           0       0     0
20           1  0.2682 -0.2885 -0.0947  0           0       0     1
21           1  0.2722 -0.3084 -0.0922  1           0       0     0
22           1  0.2794 -0.3435 -0.0876  0           0       1     0
attr(,"assign")
[1] 0 1 1 1 2 2 2 2
attr(,"contrasts")
attr(,"contrasts")$group
[1] "contr.treatment"
```

Dispersion estimation

The *edgeR* package uses negative binomial (NB) distribution to model read counts of each gene across all the sample. The NB dispersions are estimated by the `estimateDisp` function. The estimated common, trended and gene-specific dispersions can be visualized by `plotBCV` (Figure 9).

```
> y <- estimateDisp(y, design)
> sqrt(y$common.dispersion)

[1] 0.588

> plotBCV(y)
```

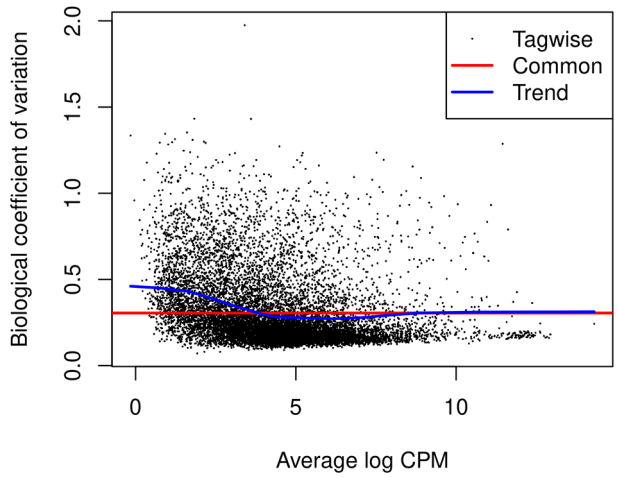

Figure 9. A scatter plot of the biological coefficient of variation (BCV) against the average abundance of each gene in log2 count-per-million (CPM). The square-root estimates of the common, trended and gene-wise NB dispersions are shown.

The NB model can be extended with quasi-likelihood (QL) methods to account for gene-specific variability from both biological and technical sources.[22,23] Note that only the trended NB dispersion is used in the QL method. The gene-specific variability is captured by the QL dispersion, which is the dispersion parameter of the negative binomial QL generalized linear model.

The `glmQLFit` function is used to fit a QL model and estimate QL dispersions. The QL dispersion estimates can be visualized by `plotQLDisp` (Figure 10).

```
> fit <- glmQLFit(y, design, robust=TRUE)
> plotQLDisp(fit)
```

Time course trend analysis

The QL F-tests are performed by `glmQLFTest` in *edgeR* to identify genes that change significantly along the pseudotime. The tests are conducted on all three covariates of the spline model matrix. This is because the significance of any of the three coefficients would indicate a strong correlation between gene expression and pseudotime.

```
> res <- glmQLFTest(fit, coef=2:4)
```

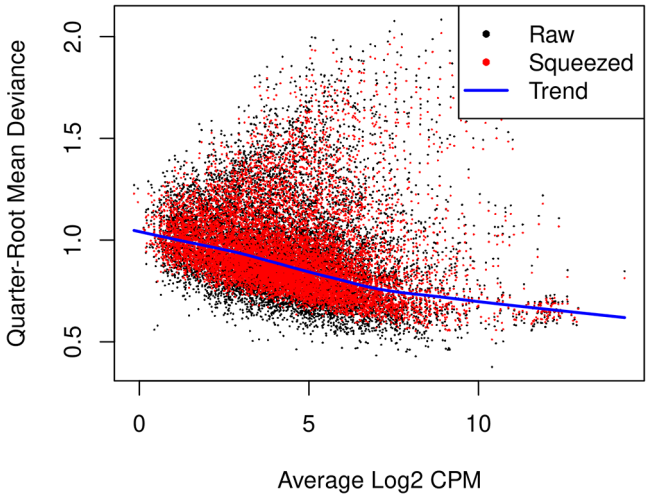

Figure 10. A scatter plot of the quarter-root QL dispersion against the average abundance of each gene in log2 count-per-million (CPM). Estimates are shown for the raw, trended and squeezed dispersions.

The number of genes significantly associated with pseudotime (FDR < 0.05) are shown below.

```
> summary(decideTests(res))

       Z3-Z2-Z1
NotSig     4843
Sig        6707
```

Top significant genes can be viewed by `topTags`.

```
> topTags(res, n=10L)

Coefficient:  Z1 Z2 Z3
           Ensembl_geneid  Symbol ENTREZID logFC.Z1 logFC.Z2 logFC.Z3 logCPM
Fhod3    ENSMUSG00000034295   Fhod3   225288   -13.88   1.1400    1.201   4.73
Mlph     ENSMUSG00000026303    Mlph   171531    10.83   1.5074    2.449   5.98
Luzp2    ENSMUSG00000063297   Luzp2   233271   -13.93   0.0755    2.317   2.16
Ptpre    ENSMUSG00000041836   Ptpre    19267    -8.28   1.9119    0.113   5.45
Aoc1     ENSMUSG00000029811    Aoc1    76507    16.04  -3.3051  -18.932   5.02
Col27a1 ENSMUSG00000045672 Col27a1   373864    -8.71   1.0071   -2.231   4.02
Jph2     ENSMUSG00000017817    Jph2    59091   -15.64  -5.8376    3.242   2.20
Popdc2   ENSMUSG00000022803  Popdc2    64082   -17.44  -7.3846    3.814   2.32
Myh11    ENSMUSG00000018830   Myh11    17880   -16.38  -3.4680    2.343   7.94
Tns1     ENSMUSG00000055322    Tns1    21961    -9.56   1.1794   -3.453   3.87
            F   PValue      FDR
Fhod3   331.3 4.26e-16 4.92e-12
Mlph    136.9 1.07e-12 6.17e-09
Luzp2   121.9 2.92e-12 9.08e-09
Ptpre   118.6 3.72e-12 9.08e-09
Aoc1    469.7 4.27e-12 9.08e-09
Col27a1 115.4 4.72e-12 9.08e-09
Jph2    104.5 1.11e-11 1.83e-08
Popdc2  101.0 1.49e-11 2.15e-08
Myh11   121.6 2.03e-11 2.60e-08
Tns1     90.7 3.72e-11 4.25e-08
```

The `logFC.Z1`, `logFC.Z2`, and `logFC.Z3` values in the table above denote the estimated coefficients of Z1, Z2, and Z3 for each gene. It should be noted that these values do not carry the same interpretation as log-fold changes in traditional RNA-seq differential expression analysis. For each gene, the sign of the coefficient `logFC.Z1` indicates whether the expression level of that gene increases or decreases along pseudotime in general. The top increasing and the top decreasing genes are listed below.

```
> tab <- topTags(res, n=Inf)$table
> tab$trend <- ifelse(tab$logFC.Z1 > 0, "Up", "Down")
> tab.up <- tab[tab$trend == "Up", ]
> tab.down <- tab[tab$trend == "Down", ]
> head(tab.up)

          Ensembl_geneid Symbol  ENTREZID logFC.Z1 logFC.Z2 logFC.Z3 logCPM     F
Mlph    ENSMUSG00000026303   Mlph   171531    10.83     1.51    2.449   5.98 136.9
Aoc1     ENSMUSG00000029811   Aoc1    76507    16.04    -3.31  -18.932   5.02 469.7
Mpzl3   ENSMUSG00000070305  Mpzl3   319742     5.37     0.92    0.723   4.90  89.8
Prr15l ENSMUSG00000047040 Prr15l   217138    13.35     2.39    3.568   4.63  87.7
Elf5    ENSMUSG00000027186   Elf5    13711     5.90     8.90    7.192   6.26  97.5
Lrrc26 ENSMUSG00000026961 Lrrc26   227618    13.99     3.05    3.276   4.68  82.9
         PValue      FDR trend
Mlph   1.07e-12 6.17e-09    Up
Aoc1   4.27e-12 9.08e-09    Up
Mpzl3  4.04e-11 4.25e-08    Up
Prr15l 4.97e-11 4.78e-08    Up
```

```
Elf5   5.80e-11 5.06e-08    Up
Lrrc26 8.00e-11 5.44e-08    Up
> head(tab.down)

            Ensembl_geneid  Symbol ENTREZID logFC.Z1 logFC.Z2 logFC.Z3 logCPM   F
Fhod3   ENSMUSG00000034295   Fhod3   225288   -13.88   1.1400    1.201   4.73 331
Luzp2   ENSMUSG00000063297   Luzp2   233271   -13.93   0.0755    2.317   2.16 122
Ptpre   ENSMUSG00000041836   Ptpre    19267    -8.28   1.9119    0.113   5.45 119
Col27a1 ENSMUSG00000045672 Col27a1   373864    -8.71   1.0071   -2.231   4.02 115
Jph2    ENSMUSG00000017817    Jph2    59091   -15.64  -5.8376    3.242   2.20 104
Popdc2  ENSMUSG00000022803  Popdc2    64082   -17.44  -7.3846    3.814   2.32 101
          PValue      FDR trend
Fhod3   4.26e-16 4.92e-12  Down
Luzp2   2.92e-12 9.08e-09  Down
Ptpre   3.72e-12 9.08e-09  Down
Col27a1 4.72e-12 9.08e-09  Down
Jph2    1.11e-11 1.83e-08  Down
Popdc2  1.49e-11 2.15e-08  Down
```

Line graphs are produced to visualize the relationship between gene expression level and pseudotime for the top 6 increasing and the top 6 decreasing genes (Figure 11).

For each gene, the expression levels (in log2-CPM) are averaged across five samples, and the line is smoothed using its predicted expression level at 100 evenly spaced pseudotime points within the pseudotime range. The smooth curves for the first 6 genes exhibit a generally increasing trend in gene expression over pseudotime, while the curves for the last 6 genes show a general decreasing trend.

```
> design2 <- model.matrix(~ X + group)
> fit2 <- glmQLFit(y, design2, robust=TRUE)
> pt <- y$samples$pseudotime
> pt_new <- round(seq(min(pt), max(pt), length.out=100), 2)
> X_new <- predict(X, newx=pt_new)
> topGenes <- c(rownames(tab.up)[1:6], rownames(tab.down)[1:6])
> par(mfrow=c(4,3))
> for(i in 1:12) {
+   Symbol <- topGenes [i]
+   beta <- coef(fit2)[Symbol,]
+   AverageIntercept <- beta[1] + mean(c(0,beta[5:8]))
+   Trend <- AverageIntercept + X_new %*% beta[2:4]
+   Trend <- (Trend + log(1e6))/log(2)
+   plot(pt_new, Trend,type="l", frame=FALSE, col="red", lwd=2,
        xlab="Pseudotime", ylab="Log2CPM", main=Symbol)
+}
```

A heatmap is generated to examine the top 20 up and top 20 down genes collectively (Figure 12). In the heatmap, pseudo-bulk samples are arranged in increasing pseudotime from left to right. The up genes are on the top half of the heatmap whereas the down genes are on the bottom half. The heatmap shows a gradual increase in expression levels of the up genes from left to right, while the down genes display the opposite trend.

```
> logCPM.obs <- edgeR::cpm(y, log=TRUE, prior.count=fit$prior.count)
> topGenes <- c(rownames(tab.up)[1:20], rownames(tab.down)[1:20])
> z <- logCPM.obs[topGenes, ]
> z <- t(scale(t(z)))
> ComplexHeatmap::Heatmap(z, name = "Z score",
+     cluster_rows = FALSE,cluster_columns = FALSE)
```

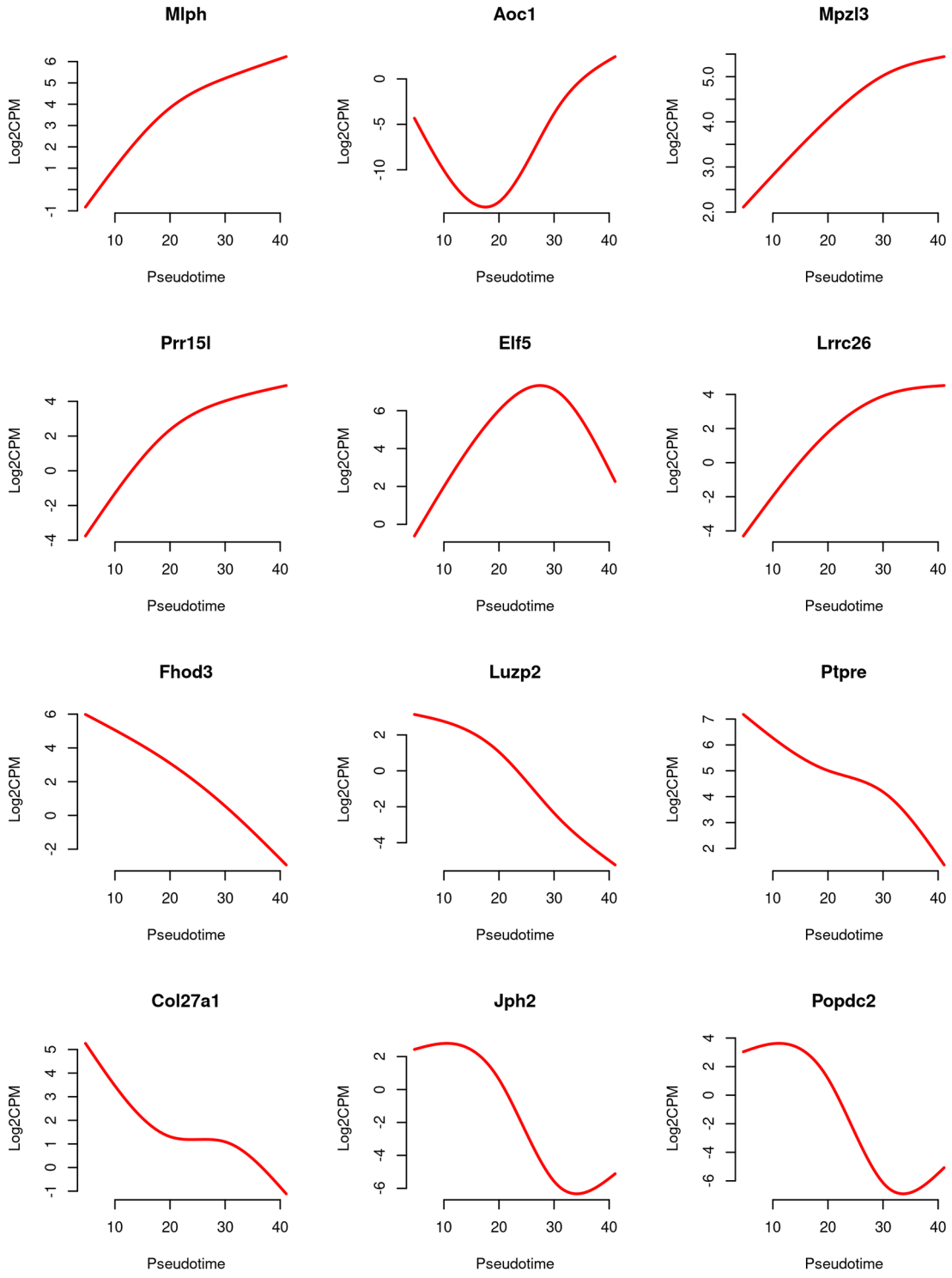

Figure 11. Line graphs of expression level of top genes along pseudotime. The red line represents the predicted expression level in log2-CPM along pseudotime.

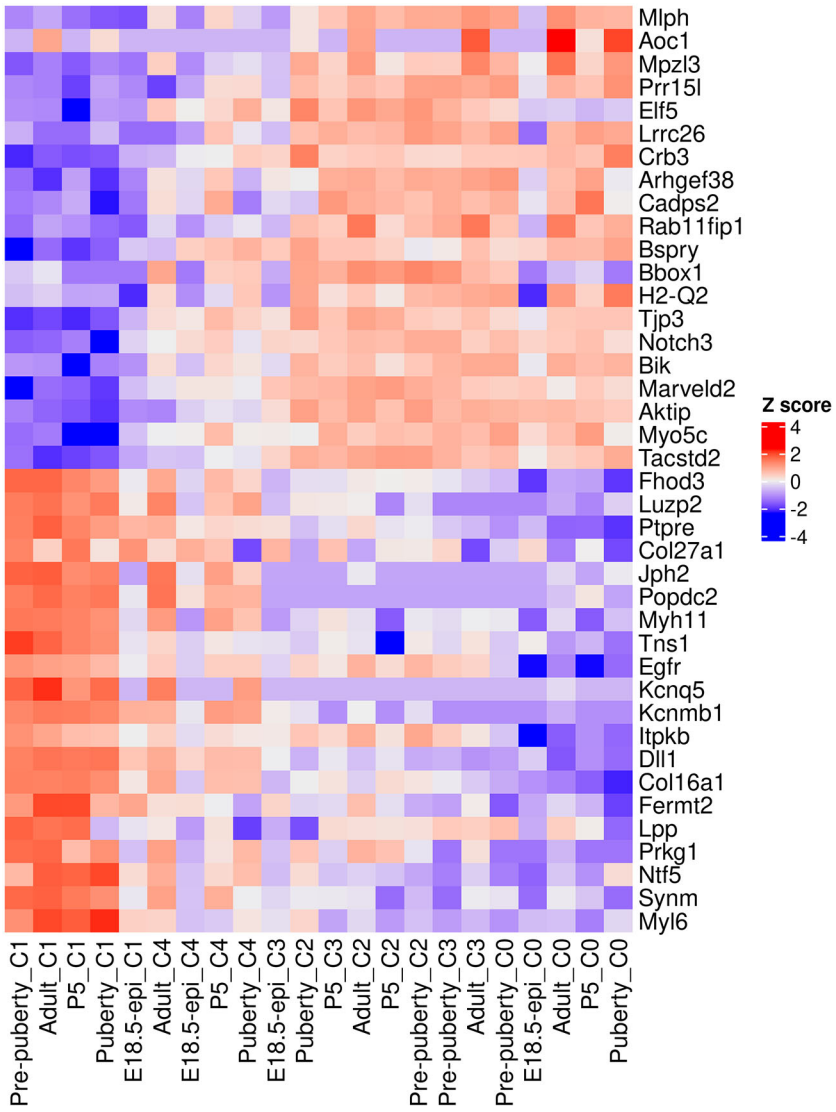

Figure 12. Heatmap of top 20 up and top 20 down genes. Rows are genes and columns are pseudo-bulk samples.

Time course functional enrichment analysis

Gene ontology analysis

To interpret the results of the time course analysis at the functional level, we perform gene set enrichment analysis. Gene ontology (GO) is one of the commonly used databases for this purpose. The GO terms in the GO databases are categorized into three classes: biological process (BP), cellular component (CC) and molecular function (MF). In a GO analysis, we are interested in finding GO terms that are over-represented or enriched with significant genes.

GO analysis is usually directional. For simplicity, we re-perform the QL F-test on the `Z1` coefficient to identify genes that exhibit a general linear increase or decrease along pseudotime. The numbers of genes with a significant increasing or decreasing linear trend are shown below.

```
> res_2 <- glmQLFTest(fit, coef=2)
> summary(decideTests(res_2))

          Z1
Down    2742
NotSig 6273
Up      2535
```

To perform a GO analysis, we apply the `goana` function to the above test results. Note that Entrez gene IDs are required for `goana`, which has been added to the `ENTREZID` column in the gene annotation. The top enriched GO terms can be viewed using `topGO` function.

```
> go <- goana(res_2, geneid="ENTREZID", species="Mm")
> topGO(go, truncate.term = 30, n=15)

                                    Term Ont    N  Up Down    P. Up   P.Down
GO:0071944                 cell periphery  CC 2928 695 1029 0.003837 4.73e-60
GO:0009653 anatomical structure morpho...  BP 1828 345  669 0.999801 6.61e-42
GO:0005576           extracellular region  CC  995 212  415 0.707860 1.19e-39
GO:0030312 external encapsulating stru...  CC  297  43  174 0.999620 9.63e-39
GO:0031012           extracellular matrix  CC  297  43  174 0.999620 9.63e-39
GO:0005886                plasma membrane  CC 2650 646  877 0.000352 2.21e-36
GO:0062023 collagen-containing extrace...  CC  240  34  145 0.999223 1.69e-34
GO:0007155                 cell adhesion  BP  833 169  350 0.894200 5.69e-34
GO:0040011                    locomotion  BP 1186 245  458 0.879526 6.97e-34
GO:0006928 movement of cell or subcell...  BP 1361 282  508 0.885421 2.59e-33
GO:0032501 multicellular organismal pr...  BP 4101 838 1238 0.998415 4.06e-33
GO:0048731             system development  BP 2741 536  886 0.999787 2.18e-32
GO:0032502          developmental process  BP 3917 776 1188 0.999971 3.33e-32
GO:0048856 anatomical structure develo...  BP 3657 724 1118 0.999941 3.45e-31
GO:0007275 multicellular organism deve...  BP 3188 615  995 0.999992 1.74e-30
```

It can be seen that most of the top GO terms are down-regulated. Here, we choose the top 10 down-regulated terms for each GO category and show the results in a bar plot (Figure 13).

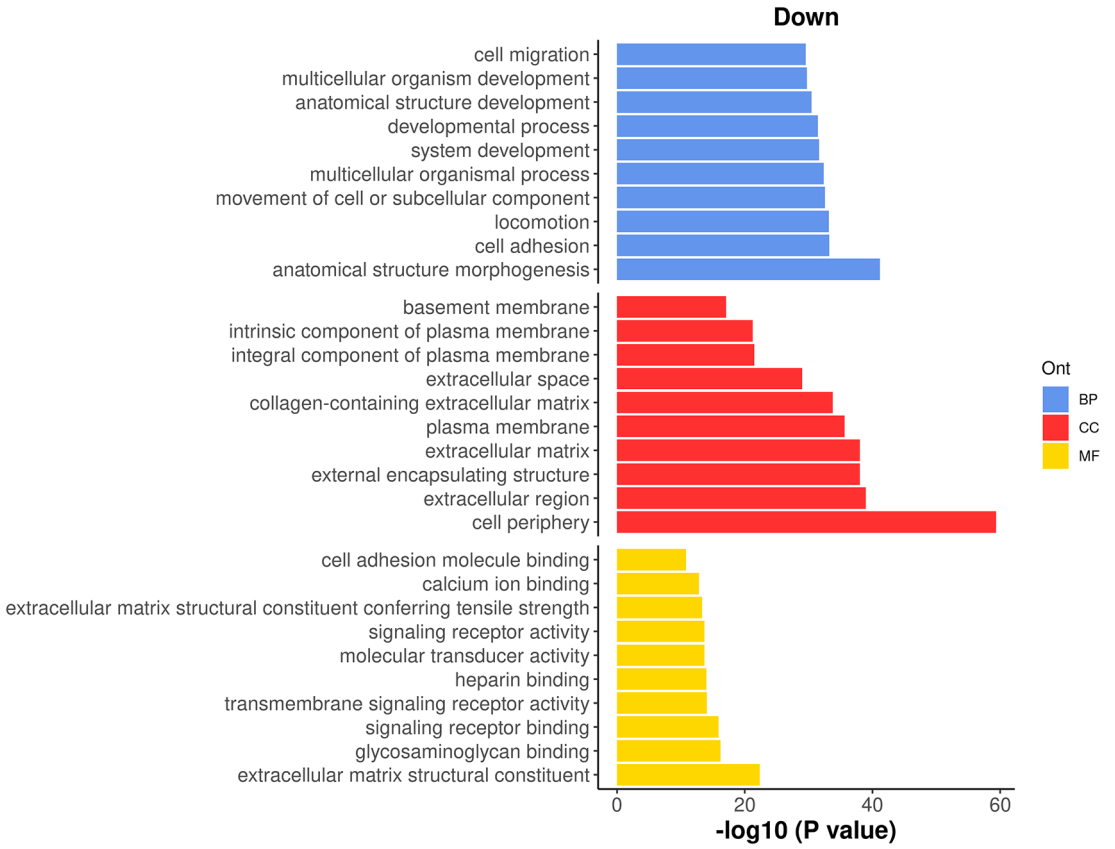

Figure 13. Bar plot of $-\log_{10}$ p-values of the top 10 down-regulated GO terms under each GO category.

```
> top_go <- rbind.data.frame(topGO(go, ont =c("BP"), sort="Down",n=10),
+                            topGO(go, ont =c("CC"), sort="Down",n=10),
+                            topGO(go, ont =c("MF"), sort="Down",n=10))
> d <- transform(top_go, P_DE = P.Down, neg_log10_P = -log10(P.Down))
> d$Term <- factor(d$Term,levels = d$Term)
> ggplot(data = d, aes(x = neg_log10_P, y = Term, fill = Ont) ) +
+   geom_bar(stat = "identity", show.legend = TRUE) +
+   labs(x="-log10 (P value)", y="", title = "Down") +
+   facet_grid(Ont~.,scales = "free",space = "free") +
+   my_theme_ggplot + my_theme_facet +
+   scale_fill_manual(values = my_colors_15[-2]) +
+   theme(strip.text = ggplot2::element_blank())
```

KEGG pathway analysis

The Kyoto Encyclopedia of Genes and Genomes[24] (KEGG) is another commonly used database for exploring signaling pathways to understand the molecular mechanism of diseases and biological processes. A KEGG analysis can be done by using `kegga` function.

The top enriched KEGG pathways can be viewed by using `topKEGG` function.

```
> kegg <- kegga(res_2, geneid="ENTREZID", species="Mm")
> topKEGG(kegg, truncate.path=40, n=15)

                                          Pathway    N   Up Down     P.Up    P.Down
mmu03010                                Ribosome  127   79    3 1.07e-22 1.00e+00
mmu05171           Coronavirus disease - COVID-19  161   92   15 1.91e-22 1.00e+00
mmu04510                          Focal adhesion  157   19   81 1.00e+00 2.79e-14
mmu04512                 ECM-receptor interaction   56    5   40 9.97e-01 5.48e-14
mmu04974         Protein digestion and absorption   49    7   35 9.36e-01 2.11e-12
mmu04015                   Rap1 signaling pathway  150   15   71 1.00e+00 1.92e-10
mmu04921                Oxytocin signaling pathway   93    9   48 9.99e-01 4.92e-09
mmu04151                PI3K-Akt signaling pathway  243   32   98 1.00e+00 4.99e-09
mmu04020                 Calcium signaling pathway  115   18   54 9.64e-01 3.98e-08
mmu05200                       Pathways in cancer  373   55  134 1.00e+00 5.16e-08
mmu05414                  Dilated cardiomyopathy   53    4   31 9.99e-01 6.08e-08
mmu01100                      Metabolic pathways 1062  302  219 1.10e-07 9.95e-01
mmu04360                           Axon guidance  143   22   62 9.81e-01 1.60e-07
mmu04024                  cAMP signaling pathway  119   19   54 9.59e-01 1.65e-07
mmu05412 Arrhythmogenic right ventricular card...   44    3   26 9.98e-01 5.38e-07
```

The results show that most of the top enriched KEGG pathways are down-regulated. Here, we select the top 15 down-regulated KEGG pathways and visualize their significance in a bar plot (Figure 14).

```
> top_path <- topKEGG(kegg,sort="Down",n=15)
> data_for_barplot <- transform(top_path, P_DE=P.Down, neg_log10_P=-log10(P.Down))
> data_for_barplot$Pathway <- factor(data_for_barplot$Pathway,
+                                    levels=data_for_barplot$Pathway)
> ggplot(data=data_for_barplot,aes(x=neg_log10_P, y=Pathway) ) +
+     geom_bar(stat="identity", show.legend=FALSE, fill=my_colors_15[1]) +
+     labs(x="-log10 (P value)", y="", title="Down" ) +
+     my_theme_ggplot
```

Among the top down-regulated pathways, the PI3K-Akt signaling pathway is noteworthy as it is typically involved in cell proliferation and plays a crucial role in mammary gland development.

To assess the overall expression level of the PI3K-Akt signaling pathway across pseudotime, a plot is generated by plotting the average expression level of all the genes in the pathway against pseudotime. The information of all the genes in the pathway can be obtained by `getGeneKEGGLinks` and `getKEGGPathwayNames`.

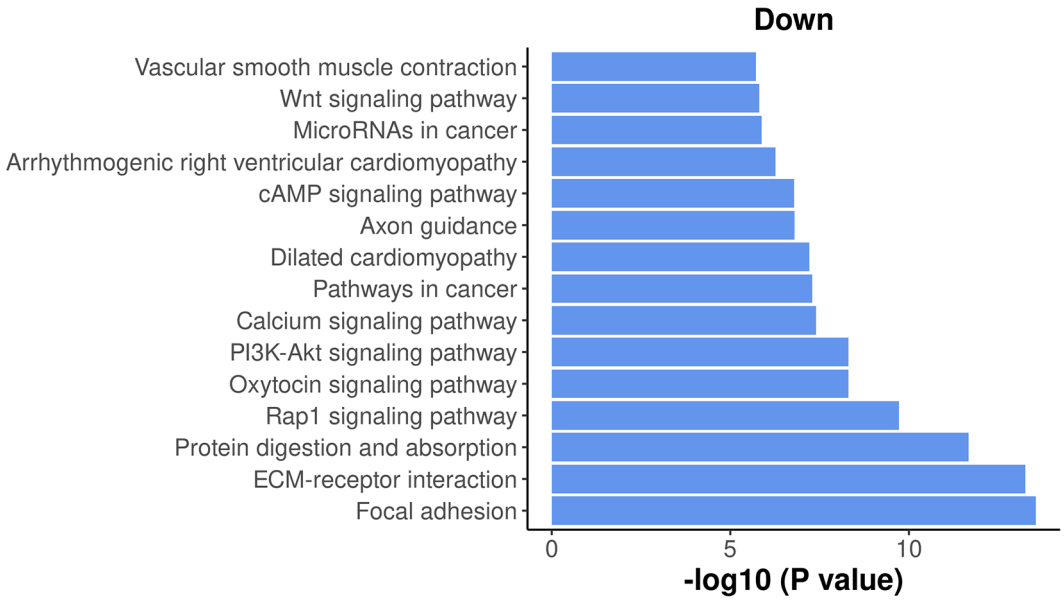

Figure 14. Bar plot of $-\log_{10}$ p-values of the top 15 down-regulated KEGG pathways.

```
> kegg_links <- getGeneKEGGLinks("mmu")
> p_names <- getKEGGPathwayNames("mmu")
> p1 <- p_names[grep("PI3K", p_names$Description), ]
> p1_GeneIDs <- subset(kegg_links, PathwayID == p1$PathwayID)$GeneID
> tab_p1 <- tab[tab$ENTREZID %in% p1_GeneIDs, ]
> d <- logCPM.obs[tab_p1$Symbol,]
> d <- apply(d, 2, mean)
> d <- data.frame(avg_logCPM = d, avg_pseudotime = y$samples$pseudotime)
> head(d)

               avg_logCPM avg_pseudotime
Pre-puberty_C1       4.43           4.65
Adult_C1             4.69           4.77
P5_C1                4.70           6.41
Puberty_C1           4.18           6.48
E18.5-epi_C1         4.46          10.16
Adult_C4             3.53          19.25
```

The plot below clearly illustrates a significant down-regulation of the PI3K-Akt pathway along pseudotime (Figure 15).

```
> ggplot(data = d,aes(x = avg_pseudotime, y = avg_logCPM) ) +
+     geom_smooth(color=my_colors_15[1],se = FALSE) +
+     labs(x="Pseudotime", y="Average log-CPM",
+          title = "PI3K-Akt signaling pathway" ) +
+     my_theme_ggplot
```

Discussion

In this article, we demonstrated a complete workflow of a pseudo-temporal trajectory analysis of scRNA-seq data. This workflow takes single-cell count matrices as input and leverages the Seurat pipeline for standard scRNA-seq analysis, including quality control, normalization, and integration. The *scDblFinder* package is utilized for doublet prediction. Trajectory inference is conducted with *monocle3*, while the *edgeR* QL framework with a pseudo-bulking strategy is applied for pseudo-time course analysis. Alternative methods and packages can be used interchangeably with the ones implemented in this study, as long as they perform equivalent functions. For instance, the Bioconductor workflow may be substituted for the Seurat pipeline in scRNA-seq analysis, whereas the *slingshot* package may replace *monocle3* for performing trajectory analysis.

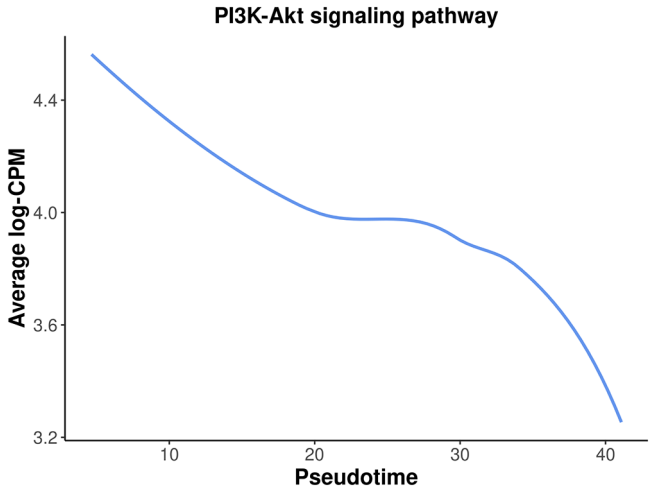

Figure 15. A smooth curve of PI3K-Akt signaling pathway expression level against pseudotime.

This workflow article utilized 10x scRNA-seq data from five distinct stages of mouse mammary gland development, with a focus on the lineage progression of epithelial cells. By performing a time course analysis based on pseudotime along the developmental trajectory, we successfully identified genes and pathways that exhibit differential expression patterns over the course of pseudotime. The results of this extensive analysis not only confirm previous findings in the literature regarding the mouse mammary gland epithelium, but also reveal genes and pathways that exhibit continuous changes along the epithelial lineage. The analytical framework presented here can be utilized for any single-cell experiments aimed at studying dynamic changes along a specific path, whether it involves cell differentiation or the development of cell types.

Packages used

This workflow depends on various packages from the Bioconductor project version 3.15 and the Comprehensive R Archive Network (CRAN), running on R version 4.2.1 or higher. The complete list of the packages used for this workflow are shown below:

```
> sessionInfo()

R version 4.2.1 (2022-06-23)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)

Matrix products: default
BLAS:   /stornext/System/data/apps/R/R-4.2.1/lib64/R/lib/libRblas.so
LAPACK: /stornext/System/data/apps/R/R-4.2.1/lib64/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_AU.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_AU.UTF-8        LC_COLLATE=en_AU.UTF-8
 [5] LC_MONETARY=en_AU.UTF-8    LC_MESSAGES=en_AU.UTF-8
 [7] LC_PAPER=en_AU.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_AU.UTF-8 LC_IDENTIFICATION=C

attached base packages:
 [1] stats4    stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] org.Mm.eg.db_3.15.0         AnnotationDbi_1.58.0
 [3] monocle3_1.2.9              SingleCellExperiment_1.18.0
 [5] SummarizedExperiment_1.26.1 GenomicRanges_1.48.0
 [7] GenomeInfoDb_1.32.2         IRanges_2.30.0
```

```
 [9] S4Vectors_0.34.0            MatrixGenerics_1.8.0
[11] matrixStats_0.62.0          Biobase_2.56.0
[13] BiocGenerics_0.42.0         scDblFinder_1.10.0
[15] sp_1.5-0                    SeuratObject_4.1.0
[17] Seurat_4.1.1                ggplot2_3.3.6
[19] edgeR_3.38.1                limma_3.55.5

loaded via a namespace (and not attached):
 [1] utf8_1.2.2              R.utils_2.11.0            reticulate_1.25
 [4] lme4_1.1-29             tidyselect_1.1.2          RSQLite_2.2.14
 [7] htmlwidgets_1.5.4       grid_4.2.1                BiocParallel_1.30.3
[10] Rtsne_0.16              munsell_0.5.0             ScaledMatrix_1.4.0
[13] codetools_0.2-18        ica_1.0-2                 statmod_1.4.36
[16] scran_1.24.0            xgboost_1.6.0.1           future_1.26.1
[19] miniUI_0.1.1.1          withr_2.5.0               spatstat.random_2.2-0
[22] colorspace_2.0-3        progressr_0.10.1          highr_0.9
[25] knitr_1.39              ROCR_1.0-11               tensor_1.5
[28] listenv_0.8.0           labeling_0.4.2            GenomeInfoDbData_1.2.8
[31] polyclip_1.10-0         bit64_4.0.5               farver_2.1.0
[34] parallelly_1.32.0       vctrs_0.4.1               generics_0.1.2
[37] xfun_0.31               doParallel_1.0.17         R6_2.5.1
[40] clue_0.3-61             ggbeeswarm_0.6.0          rsvd_1.0.5
[43] locfit_1.5-9.5          cachem_1.0.6              bitops_1.0-7
[46] spatstat.utils_2.3-1    DelayedArray_0.22.0       assertthat_0.2.1
[49] promises_1.2.0.1        BiocIO_1.6.0              scales_1.2.0
[52] rgeos_0.5-9             beeswarm_0.4.0            gtable_0.3.0
[55] beachmat_2.12.0         Cairo_1.5-15              globals_0.15.0
[58] goftest_1.2-3           rlang_1.0.2               GlobalOptions_0.1.2
[61] splines_4.2.1           rtracklayer_1.56.0        lazyeval_0.2.2
[64] spatstat.geom_2.4-0     BiocManager_1.30.18       yaml_2.3.5
[67] reshape2_1.4.4          abind_1.4-5               httpuv_1.6.5
[70] tools_4.2.1             ellipsis_0.3.2            spatstat.core_2.4-4
[73] RColorBrewer_1.1-3      proxy_0.4-27              ggridges_0.5.3
[76] Rcpp_1.0.8.3            plyr_1.8.7                sparseMatrixStats_1.8.0
[79] zlibbioc_1.42.0         purrr_0.3.4               RCurl_1.98-1.7
[82] rpart_4.1.16            deldir_1.0-6              GetoptLong_1.0.5
[85] pbapply_1.5-0           viridis_0.6.2             cowplot_1.1.1
[88] zoo_1.8-10              ggrepel_0.9.1             cluster_2.1.3
[91] magrittr_2.0.3          data.table_1.14.2         RSpectra_0.16-1
[94] scattermore_0.8         circlize_0.4.15           lmtest_0.9-40
[97] RANN_2.6.1              fitdistrplus_1.1-8        patchwork_1.1.1
[100] mime_0.12              evaluate_0.15             xtable_1.8-4
[103] XML_3.99-0.10          shape_1.4.6               gridExtra_2.3
[106] compiler_4.2.1         scater_1.24.0             tibble_3.1.7
[109] KernSmooth_2.23-20     crayon_1.5.1              R.oo_1.25.0
[112] minqa_1.2.4            htmltools_0.5.2           mgcv_1.8-40
[115] later_1.3.0            tidyr_1.2.0               DBI_1.1.3
[118] ComplexHeatmap_2.12.0  MASS_7.3-57               boot_1.3-28
[121] leidenbase_0.1.11      Matrix_1.5-3              cli_3.3.0
[124] R.methodsS3_1.8.2      parallel_4.2.1            metapod_1.4.0
[127] igraph_1.3.2           pkgconfig_2.0.3           GenomicAlignments_1.32.0
[130] terra_1.5-34           plotly_4.10.0             scuttle_1.6.2
[133] spatstat.sparse_2.1-1  foreach_1.5.2             vipor_0.4.5
[136] dqrng_0.3.0            XVector_0.36.0            stringr_1.4.0
[139] digest_0.6.29          sctransform_0.3.3         RcppAnnoy_0.0.19
[142] spatstat.data_2.2-0    Biostrings_2.64.0         leiden_0.4.2
[145] uwot_0.1.11            DelayedMatrixStats_1.18.0 restfulr_0.0.15
[148] shiny_1.7.1            Rsamtools_2.12.0          nloptr_2.0.3
```

```
[151] rjson_0.2.21           lifecycle_1.0.1          nlme_3.1-158
[154] jsonlite_1.8.0         SeuratWrappers_0.3.0     BiocNeighbors_1.14.0
[157] viridisLite_0.4.0      fansi_1.0.3              pillar_1.7.0
[160] lattice_0.20-45        GO.db_3.15.0             KEGGREST_1.36.2
[163] fastmap_1.1.0          httr_1.4.3               survival_3.3-1
[166] remotes_2.4.2          glue_1.6.2               iterators_1.0.14
[169] png_0.1-7              bit_4.0.4                bluster_1.6.0
[172] stringi_1.7.6          blob_1.2.3               BiocSingular_1.12.0
[175] memoise_2.0.1          dplyr_1.0.9              irlba_2.3.5
[178] future.apply_1.9.0
```

Data availability

Underlying data

The single-cell RNA-seq datasets used in this study were obtained from the Gene Expression Omnibus (GEO) with accession numbers of GSE103275[17] and GSE164017.[18]

Software availability

Source code available from: https://github.com/jinming-cheng/TimeCoursePaperWorkflow

Archived source code at time of publication: https://doi.org/10.5281/zenodo.7879833[25]

License: GNU General Public License version 3 (GPL-3.0-only)

All the packages used in this workflow are publicly available from the Bioconductor project (version 3.15) and the Comprehensive R Archive Network (CRAN).

References

1. Macosko EZ, Basu A, Satija R, *et al.*: **Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets.** *Cell.* 2015; **161**(5): 1202–1214.
PubMed Abstract | Publisher Full Text | Free Full Text

2. Klein AM, Mazutis L, Akartuna I, *et al.*: **Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells.** *Cell.* 2015; **161**(5): 1187–1201.
PubMed Abstract | Publisher Full Text | Free Full Text

3. Zheng GX, Terry JM, Belgrader P, *et al.*: **Massively parallel digital transcriptional profiling of single cells.** *Nat. Commun.* 2017; **8**: 14049.
PubMed Abstract | Publisher Full Text | Free Full Text

4. Hao Y, Hao S, Andersen-Nissen E 3rd, *et al.*: **Integrated analysis of multimodal single-cell data.** *Cell.* 2021; **184**(13): 3573–3587.e29.
PubMed Abstract | Publisher Full Text | Free Full Text

5. Amezquita RA, Lun ATL, Becht E, *et al.*: **Orchestrating single-cell analysis with Bioconductor.** *Nat. Methods.* 2020; **17**(2): 137–145.
Publisher Full Text

6. Wolf FA, Angerer P, Theis FJ, *et al.*: **SCANPY: large-scale single-cell gene expression data analysis.** *Genome Biol.* 2018 Feb 6; **19**(1): 15.
PubMed Abstract | Free Full Text

7. Korsunsky I, Millard N, Fan J, *et al.*: **Fast, sensitive and accurate integration of single-cell data with Harmony.** *Nat. Methods.* 2019; **16**(12): 1289–1296.
PubMed Abstract | Publisher Full Text | Free Full Text

8. Haghverdi L, Lun ATL, Morgan MD, *et al.*: **Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.** *Nat. Biotechnol.* 2018; **36**(5): 421–427.
PubMed Abstract | Publisher Full Text | Free Full Text

9. Risso D, Perraudeau F, Gribkova S, *et al.*: **A general and flexible method for signal extraction from single-cell RNA-seq data.** *Nat. Commun.* 2018; **9**(1): 284.
Publisher Full Text

10. Lun AT, McCarthy DJ, Marioni JC: **A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor.** *F1000Res.* 2016; **5**: 2122.
PubMed Abstract | Publisher Full Text

11. Crowell HL, Soneson C, Germain PL, *et al.*: **muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data.** *Nat. Commun.* 2020; **11**(1): 6077.
Publisher Full Text

12. Cao J, Spielmann M, Qiu X, *et al.*: **The single-cell transcriptional landscape of mammalian organogenesis.** *Nature.* 2019; **566**(7745): 496–502.
PubMed Abstract | Publisher Full Text | Free Full Text

13. Street K, Risso D, Fletcher RB, *et al.*: **Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics.** *BMC Genomics.* 2018; **19**(1): 477.
PubMed Abstract | Publisher Full Text | Free Full Text

14. Trapnell C, Cacchiarelli D, Grimsby J, *et al.*: **The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells.** *Nat. Biotechnol.* 2014; **32**(4): 381–386.
PubMed Abstract | Publisher Full Text | Free Full Text

15. Saelens W, Cannoodt R, Todorov H, *et al.*: **A comparison of single-cell trajectory inference methods.** *Nat. Biotechnol.* 2019; **37**(5): 547–554.
Publisher Full Text

16. McCarthy DJ, Chen Y, Smyth GK: **Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation.** *Nucleic Acids Res.* 2012; **40**(10): 4288–4297.
PubMed Abstract | Publisher Full Text | Free Full Text

17. Pal B, Chen Y, Vaillant F, *et al.*: **Construction of developmental lineage relationships in the mouse mammary gland by single-cell RNA profiling.** *Nat. Commun.* 2017; **8**(1): 1627.
PubMed Abstract | Publisher Full Text | Free Full Text

18. Pal B, Chen Y, Milevskiy MJG, *et al.*: **Single cell transcriptome atlas of mouse mammary epithelial cells across development.** *Breast Cancer Res.* 2021; **23**(1): 69.
PubMed Abstract | Publisher Full Text | Free Full Text

19. Germain PL, Lun AT, Meixide CG, *et al.*: **Doublet identification in single-cell sequencing data using scDblFinder.** *F1000Res.* 2021; **10**: 979.
Publisher Full Text

20. Shackleton M, Vaillant F, Simpson KJ, *et al.*: **Generation of a functional mammary gland from a single stem cell.** *Nature.* 2006; **439**(7072): 84–88.
Publisher Full Text

21. Robinson MD, Oshlack A: **A scaling normalization method for differential expression analysis of RNA-seq data.** *Genome Biol.* 2010; **11**(3): R25.
PubMed Abstract | Publisher Full Text | Free Full Text

22. Lund SP, Nettleton D, McCarthy DJ, *et al.*: **Detecting differential expression in RNA-sequence data using quasi-likelihood with shrunken dispersion estimates.** *Stat. Appl. Genet. Mol. Biol.* 2012; **11**(5): Article 8.
PubMed Abstract | Publisher Full Text

23. Chen Y, Lun AT, Smyth GK: **From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline.** *F1000Res.* 2016; **5**: 1438.

24. Kanehisa M, Goto S: **KEGG: Kyoto Encyclopedia of Genes and Genomes.** *Nucleic Acids Res.* 2000; **28**(1): 27–30.
PubMed Abstract | Publisher Full Text | Free Full Text

25. Cheng J, Smyth GK, Chen Y: **Source code of a single-cell RNA-seq pseudo-temporal trajectory analysis.** *Zenodo.* May 2023.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ? ?

Version 2

Reviewer Report 20 November 2023

https://doi.org/10.5256/f1000research.157785.r222366

✔ **Michael D Morgan**
Institute of Medical Sciences, School of Medicine, Medical Sciences and Nutrition, University of Aberdeen, Aberdeen, Scotland, UK

The authors have largely addressed my previous concerns with the manuscript. A few minor issues are outstanding however.

In their response to my comments, and in the revised manuscript, the authors state "In general, the same cellranger reference build is preferred for consistency, although the effect on the downstream analysis is negligible." This trivialises the importance of harmonised genome/transcriptome annotations. While the impact on *this* analysis may be negligible, that may not hold across all data and analyses. The authors should note that it does not impact on *their specific analysis*. The reason for this nit-picking is so as not to instill bad habits in newcomers to the field.

In response to a comment on the linear vs. bifurcating nature of the trajectory. It would be useful for readers to understand that this particular epithelial lineage is known to progress from Basal -> LP -> ML. The current manuscript text can be interpreted as either ML <- Basal -> LP or Basal -> LP -> ML. This is important because the biology drives an analytical choice here - this is purely for clarity.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computational biology, single-cell, genetics, immunology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 31 August 2023

https://doi.org/10.5256/f1000research.147104.r190769

?

Anna Alemany
[1] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, Netherlands Antilles
[2] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, Netherlands Antilles
Xuan Quy Nguyen
[1] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, The Netherlands
[2] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, The Netherlands
Noëlle Dommann
[1] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, The Netherlands
[2] Department of Anatomy and Embryology, Leiden University Medical Center, Leiden, The Netherlands

In this manuscript, Cheng *et al* describe an R pipeline to perform scRNA-seq analysis, filter out specific cell types present in different datasets, integrate them together and perform pseudo-temporal analysis using monocle3. In addition, the authors then introduce a modified RNAseq analysis to identify genes with differential expression patterns along pseudo-temporal trajectories and perform both GO and KEGG analysis.

The pipeline that they present has the potential to become a tutorial for researchers that are starting scRNAseq analysis. However, some parts of the text and the code require extra clarification (e.g. the pseudo-bulk analysis and the fitting of dispersion parameters suffers a lot from lack of explanations). Below, we summarise some comments that we hope the authors can use to make their manuscript stronger.

- When locally rerunning the code, outcomes are not fully reproducible. Most likely this is due to the fact the authors do not set random state for the code. The authors should consider fixing this.

- Some lines of code present in the manuscript are missing on the Github repository. Is there any rational for this?

- In the introduction, python-specific pipelines (e.g. scanpy) should also be referred to.

- When introducing the pseudo-bulk method to identify marker genes (3rd paragraph in the

introduction), the authors mention that it has superior computational efficiency. Could the authors specify what they mean?

- In the last paragraph of the introduction, the authors say that they present a “new” single cell workflow. In what sense this pipeline is novel?

- In “reading the data” section, two for loops of the first gray box run from 1:5. However, to make it as general as possible for future users of the pipeline, 5 should be replace by length(samples) or length(targets).

- In “reading the data” section, the step in which a dge_merged object is created is unnecessary. All the QC analysis presented later is done at the level of individual samples: working with the elements in the list dge_all would simplify several lines of code.

- In “reading the data” section, in the last gray box the authors show number of genes. Maybe they could consider printing out also the number of reads per sample, since this is a common QC.
- In the sub-section “Quality control”, the authors could elaborate a bit more on what each QC parameter is telling us. For example, high mitochondrial genes indicate damaged or dead cells.

- In sub-section “Quality control”, the choice of thresholds (<500 number of genes, >10% mitochondria, thresholds for high number of genes) seem very arbitrary. The plots on Figure 1 should be modified to visualize better the effect of the different thresholds. In addition, a histogram of the log-number of reads per cell should be included as a QC. In addition, the y label in Figure 1 should be replaced: library size is not the same as number of counts per cell. In the caption, one should explicitly mention that each dot is a cell barcode.

- When each dataset is converted into a Seurat object, the parameters min.cells = 3 and min.features = 200 should be discussed. How relevant is the min.features parameter given the previous 500 threshold?
- In the “Standard Seurat analysis of individual sample” sub-sections, the authors should replace “the data of each sample” by the “raw counts of each sample” when describing the normalization step.

- In the “Standard Seurat analysis of individual sample” sub-section, 30 PCs are used by default to perform the umap embedding. The authors should discuss why 30, and whether/why this is a good choice for all the libraries.

- In addition, why do they use different resolution for each dataset for the cell clustering? What criteria is followed to decide on this? Maybe some Silhouette plot or Gap Statistics should be included?
- For the sake of readability, the authors could consider including a sub-section header after the first gray box of the “removing potential doublets and non-epithelial cells” sub-section.

- The choice of clusters expressing Epcam seems arbitrary: why is cluster 0 included in pre-puberty, why are not all the clusters included in puberty, and why in adult cluster 4 is not included? To select Epcam+ clusters a bit more quantitatively, a boxplot (or violin plot or

heatmap) should be made displaying mean Epcam expression per cluster.

- The Expcam expression range use to select epithelial cells should either be the same for the 5 samples or follow some trend as a function of time. Can the authors show this in a plot?

- Did the authors try to re-cluster data after doublet removal? This might be impactful in some datasets (such as the Puberty one).
- In "integrating epithelial cells of five samples", why do we need to integrate the data using the anchor-based method? How does the UMAP of all combined epithelial cells look like without the anchor-based strategy?

- The author listed some batch-correction methods in the introduction, but here they do not explain the choice of this anchor-based method. Could they comment on this?

- Why do the authors use the min.cells = 3 parameter in the first gray box of the "integrating epithelial cells of five samples"?

- Which criteria does the "findIntegrationAnchors"to select features?

- As before, 30 PC are used. Additionally, clustering is done using a resolution of 0.2. How did the authors decide of these values?

- While rerunning the code to generate Fig. 3, a slightly different version of the UMAP was obtained. This leads to define different cell clusters and generate list of markers gene for downstream. Maybe it would be good to set a random seed in the umap or FindNeighbors to make the results from this pipeline robust.
- In "cell type identification", the authors should consider extending the gene marker list. Why were specifically these markers selected? Csn3 and Elf5 are markers for luminal alveolar cells that are particularly active during lactation and not in the progenitor stage. Also Prlr is considered to be involved in milk production. What about performing Gene Ontology on the clusters?

- Maybe cluster 5 should be removed from downstream analysis since it is stromal cells contamination?
- What is the definition of a "key point" along a trajectory?

- How does pseudotime estimation compare to the embryo stage of origin? A scatter plot would be very informative.
- In section "constructing pseudo-bulk profiles", the authors aggregate cells with the same combination of sample and cluster. However, this goes against the anchor-based integration method. Could the authors discuss why they do not use pseudotime intervals to aggregate cells with similar pseudotime values?

- In the "design matrix" sub-section, could the authors discuss the structure of the design parameter? What are Z1, Z2 and Z3?

- What does CPM stand for in Figures 8 and 9?

- It would be beneficial to explain better what is the y-axis of Figure 9. This section is in general very challenging to follow for anyone that does not have experience in bulk differential expression analysis. At least, a proper definition of “dispersion” should be given.

- In Figure 10, it would be useful to color the black dots according to sample of origin to better understand the trends.

- Some of the genes shown in Figure 10 should also be displayed in the UMAP obtained with integration of epithelial cells to appreciate better gene expression patterns along pseudo-time and along cell types.
- In the Gene ontology and KEGG analysis subsections, it is not clear whether the GO is performed on the genes that exhibit a significant linear increase or decrease with pseudotime. Could the authors clarify this?
- In the discussion, the authors mention that their analysis revealed new insights specific to early developmental stages of the mammary gland. Could the authors elaborate on those?
- We would advise the authors to load all the required libraries at the beginning of the pipeline.

Is the rationale for developing the new method (or application) clearly explained?
No

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Stem cell biology; bioinformatics

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Author Response 23 Oct 2023

Yunshun Chen

We warmly thank the Reviewers for their positive assessment of our work, and for the numerous suggestions that helped us considerably improve the manuscript.

When locally rerunning the code, outcomes are not fully reproducible. Most likely this is due to the fact the authors do not set random state for the code. The authors should consider fixing this.

Random seeds were set for those steps that involve randomness. In particular, we use set.seed(42) before running scDblFinder::scDblFinder() and monocle3::cluster_cells(). All the Suerat functions (e.g., RunPCA, RunUMAP) also use a fixed random seed by default. The outcomes are fully reproducible provided the same versions of all the packages are used. The results may slightly differ (e.g., the selection of starting node of the time-course trajectory) if different versions of one or more packages are used.

Some lines of code present in the manuscript are missing on the Github repository. Is there any rational for this?

We thank the reviews for point this out. We have now added all the code required to run the workflow from start to finish.

In the introduction, python-specific pipelines (e.g. scanpy) should also be referred to.

The scanpy reference has now been added to the introduction.

When introducing the pseudo-bulk method to identify marker genes (3rd paragraph in the introduction), the authors mention that it has superior computational efficiency. Could the authors specify what they mean?

The advantages of the pseudo-bulk methods over single-cell level methods, including the computational efficiency, have been explored and described in greater details in the cited publication [1].

In the last paragraph of the introduction, the authors say that they present a "new" single cell workflow. In what sense this pipeline is novel?

All the existing software tools (e.g., monocle3, slingshot, etc.) perform trajectory analysis at the single-cell level. Meanwhile, most single-cell pseudo-bulk analyses are performed for group-wise comparisons (between samples, clusters, or experimental conditions). The single-cell workflow we present is novel in the way that it utilizes the advanced edgeR GLM framework in modelling pseudo-time effect and combines the single-cell level trajectory analysis with the pseudo-bulking strategy.

In "reading the data" section, two for loops of the first gray box run from 1:5. However, to make it as general as possible for future users of the pipeline, 5 should be replace by length(samples) or length(targets).

We thank the reviewers for the kind suggestion. Since the number of samples is fixed at 5 throughout the manuscript, we think using '5' explicitly would simplify the code a bit. However, we could certainly consider generalize the coding format in the future.

In "reading the data" section, the step in which a dge_merged object is created is unnecessary. All the QC analysis presented later is done at the level of individual samples: working with the elements in the list dge_all would simplify several lines of code.

Since one of the samples (Puberty) was processed using a different version of mouse genome, subsetting by row is required for all five DGEList objects. And this is easier to deal with using the merged DGEList object.

In "reading the data" section, in the last gray box the authors show number of genes. Maybe they could consider printing out also the number of reads per sample, since this is a common QC.

The number of reads per sample reflects the sequencing depth of that entire sample, which is of less interest for QC at single-cell level. The cell-level QC statistics, which we showed in Fig1, are often more informative.

In the sub-section "Quality control", the authors could elaborate a bit more on what each QC parameter is telling us. For example, high mitochondrial genes indicate damaged or dead cells.

We thank the reviewers for the comment. We have added some extra explanation on interpreting the QC metrics.

In sub-section "Quality control", the choice of thresholds (<500 number of genes, >10% mitochondria, thresholds for high number of genes) seem very arbitrary. The plots on Figure 1 should be modified to visualize better the effect of the different thresholds. In addition, a histogram of the log-number of reads per cell should be included as a QC. In addition, the y label in Figure 1 should be replaced: library size is not the same as number of counts per cell. In the caption, one should explicitly mention that each dot is a cell barcode.

These thresholds were chosen in the same way as in the original paper [2]. We did not visualize the thresholds in Fig1 since some of the thresholds were determined afterwards by examining the scatter plots in Fig1. We did not produce a histogram of the log-number of reads per cell as it does not provide extra information in addition to Fig1 for QC. The y-label is now renamed to 'Number of reads', and it is now explicitly mentioned in the caption that each dot is a cell.

When each dataset is converted into a Seurat object, the parameters min.cells = 3 and min.features = 200 should be discussed. How relevant is the min.features parameter given the previous 500 threshold?

When each individual data is read into a Seurat object, the "min.cells = 3" and "min.features = 200" are the initial QC parameters adopted in a Seurat online vignette [3]. The "min.features" would not be relevant to the downstream analysis since a threshold of 500 is applied right after that. We have now revised that part of the manuscript to clarify that.

In the "Standard Seurat analysis of individual sample" sub-sections, the authors should replace "the data of each sample" by the "raw counts of each sample" when describing the normalization step.

We thank the reviewers for the comment. We have revised the sentence accordingly.

In the "Standard Seurat analysis of individual sample" sub-section, 30 PCs are used by default to perform the umap embedding. The authors should discuss why 30, and whether/why this is a good choice for all the libraries.

Here we use the first 30 PCs to be consistent with the analysis in Pal *et al*. 2021 [2]. In general, the downstream results are very robust on the number of PCs chosen provided it is large enough (>10). We have now added some extra comments to the manuscript.

In addition, why do they use different resolution for each dataset for the cell clustering? What criteria is followed to decide on this? Maybe some Silhouette plot or Gap Statistics should be included?

As described in the manuscript, cell clustering resolution is carefully chosen for each sample so that distinct cell types are grouped into separate clusters. This process usually involves some trial and error with different resolution parameters so that the final clustering result agree with its UMAP visualization. We did not use any sophisticated methods or statistics to set resolution as this would make the workflow more complicated than it should be.

For the sake of readability, the authors could consider including a sub-section header after the first gray box of the "removing potential doublets and non-epithelial cells" sub-section.

We thank the reviewer for the suggestion. However, the content after the first grey box is still part of the sub-section "removing potential doublets and non-epithelial cells".

The choice of clusters expressing Epcam seems arbitrary: why is cluster 0 included in pre-puberty

The mammary gland epithelium consists of three major subtypes: basal, luminal progenitor (LP) and mature luminal (ML). Using some other marker genes, we can confirm that the cluster 0 in pre-puberty is the basal population, which is a subpopulation within epithelium.

why are not all the clusters included in puberty

In the puberty sample, cells in cluster 8 are stromal cells and they have low expression of Epcam. Cells in cluster 7 do express Epcam. However, these cells have much higher library

size than cells in other clusters, and hence could be homotypic doublets. Therefore, cluster 7 is also removed.

and why in adult cluster 4 is not included?

In the adult sample, cells in cluster 4 are stromal cells and they have low expression of Epcam.

To select Epcam+ clusters a bit more quantitatively, a boxplot (or violin plot or heatmap) should be made displaying mean Epcam expression per cluster.

As mentioned above, even though Epcam is a typical signature gene of epithelium, we still need to examine some other markers to fully confirm the identities of different cell clusters. We intentionally do not include all the details as this would go way beyond the scope of this workflow. This part of the workflow is simply to demonstrate that users can subset their data and focus on the cell type of their interest. When users apply our workflow to their own single-cell data, the marker genes and subsetting strategies might be completely different to ours.

The main focus of this workflow is to showcase a novel approach that combines the single-cell level pseudotime trajectory analysis with the pseudo bulking strategy followed by an edgeR-style time course analysis. This main part of the workflow would remain the same regardless of what cell type users are interested in and how these subsets are obtained.

The Expcam expression range use to select epithelial cells should either be the same for the 5 samples or follow some trend as a function of time. Can the authors show this in a plot?

In general, the Epcam gene is highly expressed in the epithelial cell population (basal, luminal progenitor, and mature luminal cells) compared to other cell population. Here we identify epithelial cell clusters by examining the expression level of Epcam within each individual sample. However, the Epcam expression levels are not directly comparable between different samples.

Did the authors try to re-cluster data after doublet removal? This might be impactful in some datasets (such as the Puberty one).

We did try re-clustering data after doublet removal. We did not show it in the manuscript as i) the clustering results are very similar as before, and ii) all the samples are integrated right after doublet removal so there is no use of the re-clustering results from individual sample.

In "integrating epithelial cells of five samples", why do we need to integrate the data using the anchor-based method? How does the UMAP of all combined epithelial cells look like without the anchor-based strategy?

If no integration is performed, a clear batch (sample) effect can be observed on UMAP (i.e., cells of the same cell population are separated by sample).

The author listed some batch-correction methods in the introduction, but here they do not explain the choice of this anchor-based method. Could they comment on this?

We use the anchor-based method as we adopt the Seurat workflow for all the single-cell analyses in the manuscript and Seurat uses the anchor-based integration method by default. We now added some comments on other integration methods as well.

Why do the authors use the min.cells = 3 parameter in the first gray box of the "integrating epithelial cells of five samples"?

The 'min.cells = 3' is used to remove lowly expressed genes as these genes are not of any biological interest here. The same threshold is also used in the Seurat online vignette [4] although it is somewhat *ad hoc* and can be adjusted depending on the data. We added some extra comments on that to the manuscript.

Which criteria does the "findIntegrationAnchors" to select features?

The details of the feature selection criteria in "findIntegrationAnchors" is explained in the Seurat paper [5]. We did not include it as it is beyond the scope of the manuscript.

As before, 30 PC are used. Additionally, clustering is done using a resolution of 0.2. How did the authors decide of these values?

As mentioned above, we use the first 30 PCs to be consistent with the analysis in Pal *et al*. 2021 [2]. The choice of the number of PCs (i.e., 30) is also consistent with the Seurat online vignette [4]. As for the cell clustering resolution, we choose 0.2 after experimenting with different resolution parameters. This is because under this resolution the three major epithelial subpopulations, two intermediate cell clusters, and a small group of stroma cells can be clearly separated in distinct cell clusters. We added some extra comments to the manuscript.

While rerunning the code to generate Fig. 3, a slightly different version of the UMAP was obtained. This leads to define different cell clusters and generate list of markers gene for downstream. Maybe it would be good to set a random seed in the umap or FindNeighbors to make the results from this pipeline robust.

We thank the reviewers for the kind suggestions. As mentioned before, we set random seeds for analysis steps that involve randomness. In addition, all the Seurat functions such as RunPCA and RunUMAP use a fixed random seed by default. The outcomes are fully reproducible if the same versions of R and all the R packages, as well as the same operating system, are used. The whole analysis workflow itself is very robust, and the results (e.g., marker genes, UMAP visualization, etc.) would not be significantly different if different versions of R or R packages are used.

In "cell type identification", the authors should consider extending the gene marker list. Why were specifically these markers selected? Csn3 and Elf5 are markers for luminal

alveolar cells that are particularly active during lactation and not in the progenitor stage. Also Prlr is considered to be involved in milk production. What about performing Gene Ontology on the clusters?

The mouse mammary gland epithelium has been well studied in the literature. We have extensive lists of marker genes for each of the three major epithelial subpopulations (basal, LP and ML) from previous bulk RNA-seq experiments [6]. In fact, these hand-picked marker genes are all typical markers of the corresponding epithelial major cell populations, and they all appear in the lists of DE genes of the published study. There is no need to perform GO analysis as we can already identify the three major epithelial subpopulations with great confidence.

Maybe cluster 5 should be removed from downstream analysis since it is stromal cells contamination?

We thank the reviewers for the suggestion. The cluster 5 is stroma contamination, but it only contains a total of 18 cells compared to ~22,000 cells from all five samples. The effect of having this small cell cluster in the downstream analysis is negligible. Therefore, we did not remove it for simplicity.

What is the definition of a "key point" along a trajectory?

We thank the reviewers for pointing this out. A more precise word for this should be "principal nodes" as used by the monocle3 authors. The principal nodes (or points) include roots, leaves and branch points in the graph (the trajectory is a graph), they are identified by learn_graph() function in monocle3.

How does pseudotime estimation compare to the embryo stage of origin? A scatter plot would be very informative.

The information of the pseudo-time estimation under each embryo stage of origin is summarized in the MDS plots (Fig 7).

In section "constructing pseudo-bulk profiles", the authors aggregate cells with the same combination of sample and cluster. However, this goes against the anchor-based integration method. Could the authors discuss why they do not use pseudotime intervals to aggregate cells with similar pseudotime values?

Aggregating cells with the same combination of sample and cluster is the traditional way of constructing pseudo-bulk profiles [1], regardless of the integration method. We don't see why this is considered "against the anchor-based integration method". Using pseudotime intervals could be an alternative way for constructing pseudo-bulk samples. We do not use it here because the current approach (sample-cluster combination) is more straightforward, and it provides enough information for assessing the biological variation between different samples.

In the "design matrix" sub-section, could the authors discuss the structure of the design

parameter? What are Z1, Z2 and Z3?

Response: Z1, Z2, and Z3 are the three columns of the matrix that represents the family of piecewise-cubic splines generated by splines::ns(). Their values do not have any particular meaning in general. More detailed description has been added to the manuscript regarding the interpretation of the three spline coefficients.

What does CPM stand for in Figures 8 and 9?

CPM stands for 'count-per-million'. The figure captions are now revised to explain what CPM means.

It would be beneficial to explain better what is the y-axis of Figure 9. This section is in general very challenging to follow for anyone that does not have experience in bulk differential expression analysis. At least, a proper definition of "dispersion" should be given.

As stated in caption of Fig 9 (now Fig 10), the y-axis represents the quarter-root of the QL dispersion. We now added an extra sentence describing what QL dispersion is. References were also given for readers who want to know more about the details. Note that reads are expected to have some prior knowledge in both single-cell RNA-seq and bulk RNA-seq analysis to effectively follow the entire workflow.

In Figure 10, it would be useful to color the black dots according to sample of origin to better understand the trends.

We thank the reviewers for this comment. However, this particular analysis and its results focus on pseudotime rather than sample of origin. Adding irrelevant information such as sample of origin to Fig 10 (now Fig11) would make the plots less straightforward.

Some of the genes shown in Figure 10 should also be displayed in the UMAP obtained with integration of epithelial cells to appreciate better gene expression patterns along pseudo-time and along cell types.

We thank the reviewers for this suggestion. The plots in Fig 10 (now Fig 11) have already illustrated the strong association between gene expression and pseudo-time. We don't see the necessity of displaying them again in the UMAP. In addition, this workflow was structured in the way that every step after "Pseudo-bulk time course analysis with edgeR" is performed in edgeR at the pseudo-bulk level.

In the Gene ontology and KEGG analysis subsections, it is not clear whether the GO is performed on the genes that exhibit a significant linear increase or decrease with pseudotime. Could the authors clarify this?

We did mention in the manuscript that: "To perform a GO analysis, we apply the goana function to the above test results". To avoid confusion, we also used a different object 'res_2' for storing the testing results.

In the discussion, the authors mention that their analysis revealed new insights specific to early developmental stages of the mammary gland. Could the authors elaborate on those?

The new insights refer to the discoveries of genes and pathways that exhibit continuous changes along the epithelial lineage, which is now specifically mentioned in the discussion now.

We would advise the authors to load all the required libraries at the beginning of the pipeline.

We thank the reviewers for this suggestion. We decided to load each package when it is first used. This would give users a better understanding of which part of the workflow is conducted using which specific R package(s). We also added a section of "Package installation" at the start to tell users what packages are required for running through the workflow. Once all the required packages are installed, there would be no trouble loading them later on.

References:

1. H. L. Crowell, *et al*. muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nat. Commun*., 11 (1):6077, 2020.
2. B. Pal, *et al*. Single cell transcriptome atlas of mouse mammary epithelial cells across development. Breast Cancer Res., 23(1):69, 2021.
3. https://satijalab.org/seurat/articles/pbmc3k_tutorial.html
4. https://satijalab.org/seurat/articles/integration_introduction
5. Y. Hao, *et al*. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587 e29, 2021.
6. J. M. Sheridan, *et al*. A pooled shRNA screen for regulators of primary mammary stem and progenitor cells identifies roles for Asap1 and Prox1. *BMC Cancer* 15 (1): 221. 2015

Competing Interests: No competing interests were disclosed.

Reviewer Report 25 August 2023

https://doi.org/10.5256/f1000research.147104.r190774

?

Koen Van den berge
[1] Statistics and Decision Sciences, Janssen R&D, Beerse, Belgium

[2] Statistics and Decision Sciences, Janssen R&D, Beerse, Belgium

This article develops a method to discover genes whose gene expression is associated with a dynamic process, represented as a trajectory; a timely and critical contribution that is useful for the community. While several methods exist for this, the authors develop a procedure that is able to deal with multi-sample single-cell RNA-sequencing data. This extension is important, however some steps in the workflow may be improved upon.

Major comments:

- The workflow R script on GitHub does not contain several chunks of code of the workflow. It seems like many of the chunks that require a larger amount of time are not included, and need to be copied from the paper to the R script if one would like to reproduce the analysis. Please ensure the script is complete to increase ease of reproducibility.

- The authors use a 'standard' Seurat workflow to process the dataset before showcasing their methodology. While I understand that the general workflow is considered standard and expanding on it too much would deviate from the focus of the article, the description of some of the steps is inaccurate or incomplete. Across the manuscript, it would be good to add a bit more context. For example, for the first steps of the workflow, the 'NormalizeData' function does not simply log-normalize but divides the count by the total count for that cell, multiplies by a scale factor and then log-normalizes. Also, please mention the number of PCs being calculated, the clustering method being used and why sample-specific clustering resolutions being chosen for different samples.

- When following the workflow, I was surprised to see the authors pick a starting point for the trajectory in a region containing mainly cells from the later stages of development (pre-puberty/puberty/adult). The trajectory constructed in this way therefore goes from late stage (cluster 1) to early stage (cluster 4), back to late stage (cluster 2), back to early stage (cluster 3) and finally back to late stage cells (cluster 0). I therefore am suspicious of the biological relevance of this trajectory. Would a branching trajectory starting in the early stage not make more sense? Would the authors' method be able to handle such a setting? If not, an alternative dataset may be more useful, and this limitation should be clearly stated.

- I like the authors' push towards thinking about dealing with replication in trajectory-based differential expression analysis, but some steps seem rather crude:

  (a) the pseudo-bulking happens in the traditional way: for each combination of sample and cluster/cell type. These clusters are obtained in an unsupervised way, with no knowledge of the underlying trajectory and may therefore contain cells with very different pseudotimes. For example, cluster 3 has a group of cells at the left hand side of the UMAP (around -5 of first UMAP dimension), a region with a relatively low pseudotime, but most cells reside in a region with a high pseudotime (around +4 of first UMAP dimension). Would a trajectory-informed grouping of cells make more sense, e.g., making groups of cells based on binning pseudotime?

  (b) The pseudotime corresponding to each pseudobulk sample is obtained by averaging all cell-level pseudotimes. This seems simplistic, and I wonder if alternatives would be useful. First, the average may not be the best summary metric, given the distribution of

pseudotimes within each of the groups (see figure I posted here https://github.com/jinming-cheng/TimeCoursePaperWorkflow/issues/1), and a median may be more appropriate. Alternatively, if possible, one idea would be to project the averaged/pseudobulked expression profile to the UMAP, and calculate the pseudotime of the projected point. In essence, instead of taking the average of pseudotime, one would project the average expression profile onto the trajectory.

- The authors construct a natural cubic spline and use its basis functions as covariates in edgeR. This is an efficient way of estimating smooth functions of pseudotime. Note, however, that the smoothness is fixed and is assumed to be identical for all genes. In smoothing, the smoothness is often controlled using a penalty parameter that is estimated using techniques like cross-validation. This does not happen here. The authors should acknowledge this limitation.

Minor comments:

- In general, the introduction seems very brief and it would be useful to add more context. A few examples:

  - Second paragraph of introduction: it would be helpful for unfamiliar readers to expand what is meant here with 'replicate samples'. The authors are likely thinking about different samples as obtained from different subjects. The pseudobulking approach they use may not be the best approach if the replicate samples may have been derived from the same subject. In addition, integration across samples may not always be necessary and whether or not to perform this should be carefully evaluated. Minimal sample effects may occur in e.g. studies using multiplexing. It would be good to add this nuance.

  - Third paragraph of introduction: It would be relevant to specify what is meant with 'the pseudo-bulk method'.

  - Fourth paragraph: When would trajectory inference be preferred? Expand how pseudotime is derived from a trajectory and why this is useful.

- The last paragraph of the introduction should be partly rephrased:

  - "The single-cell level analysis is performed in Seurat, and the trajectory analysis is conducted using monocle3": both of these analyses are 'single-cell level analysis'.

  - "The analysis pipeline presented in this article can be applied to any scRNA-seq study with replicate samples." while this is true in theory, many datasets are not suitable for trajectory inference, as the biological context of a dataset may not constitute a 'dynamic system'.

- The paper mentions "The calculation of pseudotime, which indicates the distance between a cell and the starting cell in a trajectory, is conducted during the trajectory learning process", but this is inaccurate. The pseudotime is the distance between a cells' projection on the trajectory and the starting point of the trajectory, as measured along that trajectory. Current phrasing could be misunderstood as Euclidean distance between two points in e.g. the UMAP space.

- It may be useful to visualize the spline basis functions to allow the reader to gain intuition

on what is happening.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Statistical omics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 23 Oct 2023

Yunshun Chen

We warmly thank the Reviewer for his positive assessment of our manuscript and fruitful comments that helped us to improve the study in the revised version.

Major comments:

The workflow R script on GitHub does not contain several chunks of code of the workflow. It seems like many of the chunks that require a larger amount of time are not included, and need to be copied from the paper to the R script if one would like to reproduce the analysis. Please ensure the script is complete to increase ease of reproducibility.

We thank the reviewer for pointing this out. We agree that all the code required for the workflow shall be included in the R script on GitHub. We have now updated the R script on GitHub accordingly (https://github.com/jinming-cheng/TimeCoursePaperWorkflow). It now includes all chunks of R code that allows users to reproduce the analysis from start to finish.

The authors use a 'standard' Seurat workflow to process the dataset before showcasing their methodology. While I understand that the general workflow is considered standard and expanding on it too much would deviate from the focus of the article, the description of some of the steps is inaccurate or incomplete. Across the manuscript, it would be good to add a bit more context. For example, for the first steps of the workflow, the 'NormalizeData' function does not simply log-normalize but divides the count by the total count for that cell, multiplies by a scale factor and then log-normalizes. Also, please mention the number of PCs being calculated, the clustering method being used and why sample-specific clustering resolutions being chosen for different samples.

We thank the reviewer for this comment. We have now revised the 'standard' Seurat workflow part of the manuscript and added more detailed descriptions to some of the steps. The number of PCs and the clustering method are now mentioned in the manuscript. We also added some explanation on how cell clustering resolutions were chosen for different samples.

When following the workflow, I was surprised to see the authors pick a starting point for the trajectory in a region containing mainly cells from the later stages of development (pre-puberty/puberty/adult). The trajectory constructed in this way therefore goes from late stage (cluster 1) to early stage (cluster 4), back to late stage (cluster 2), back to early stage (cluster 3) and finally back to late stage cells (cluster 0). I therefore am suspicious of the biological relevance of this trajectory. Would a branching trajectory starting in the early stage not make more sense? Would the authors' method be able to handle such a setting? If not, an alternative dataset may be more useful, and this limitation should be clearly stated.

We thank the reviewer for this comment. Our trajectory analysis workflow is based on pseudotime rather than real time. The selection of the starting point depends on the biology or questions of interest, and it doesn't need to agree with real time. As mentioned in the manuscript, we choose a starting point for the trajectory in the basal cluster since mammary stem cells are known to be enriched in the basal population and give rise to LP and ML cells in the epithelial lineage. We wish to study how gene expression profiles change along this epithelial lineage by using the concept of trajectory and pseudotime and then performing a time-course analysis. The workflow we present is very flexible in the way that users can choose their own starting point depending on their research question. Of course, one can subset the data, focus on one particular branch, and pick a starting point at an early stage.

I like the authors' push towards thinking about dealing with replication in trajectory-based differential expression analysis, but some steps seem rather crude:

- ***the pseudo-bulking happens in the traditional way: for each combination of sample and cluster/cell type. These clusters are obtained in an unsupervised way, with no knowledge of the underlying trajectory and may therefore contain cells with very different pseudotimes. For example, cluster 3 has a group of cells at the left hand side of the UMAP (around -5 of first UMAP dimension), a region with a relatively low pseudotime, but most cells reside in a region with a high pseudotime (around +4 of***

first UMAP dimension). Would a trajectory-informed grouping of cells make more sense, e.g., making groups of cells based on binning pseudotime?

We thank the reviewer for the thoughtful comment. Yes, pseudo-bulking cells based on binning pseudotime could be an alternative method. For this workflow, we adopted the traditional pseudo-bulking approach because i) it is more straightforward, and ii) it allows us to assess the biological variation between different samples. If cells are grouped based on, say binning pseudotime, then each formed pseudo-bulk sample would contain cells from different biological samples of origin, making it harder to account for the variation between those samples in the analysis.

- ***The pseudotime corresponding to each pseudobulk sample is obtained by averaging all cell-level pseudotimes. This seems simplistic, and I wonder if alternatives would be useful. First, the average may not be the best summary metric, given the distribution of pseudotimes within each of the groups (see figure I posted here https://github.com/jinming-cheng/TimeCoursePaperWorkflow/issues/1), and a median may be more appropriate. Alternatively, if possible, one idea would be to project the averaged/pseudobulked expression profile to the UMAP, and calculate the pseudotime of the projected point. In essence, instead of taking the average of pseudotime, one would project the average expression profile onto the trajectory.***

We thank the reviewer for sharing the ideas and thoughts on this. Yes, we use the average of cellwise pseudotime as the pseudotime of that pseudo bulk sample for simplicity. We also tried using the median instead of the mean, and the results are very similar. We now added a comment to the manuscript discussing different ways of defining the pseudotime for the pseudo bulked samples. Projecting the averaged/pseudobulked expression profile back to the UMAP/trajectory sounds like a very interesting idea. However, there isn't an easy way to do so without reconstructing the UMAP and the trajectory with the projected points included. This would make the workflow way more complicated than necessary.

The authors construct a natural cubic spline and use its basis functions as covariates in edgeR. This is an efficient way of estimating smooth functions of pseudotime. Note, however, that the smoothness is fixed and is assumed to be identical for all genes. In smoothing, the smoothness is often controlled using a penalty parameter that is estimated using techniques like cross-validation. This does not happen here. The authors should acknowledge this limitation.

We appreciate the reviewer's comment. Nevertheless, it's worth noting that the edgeR pipeline requires a single design matrix for all genes. Controlling the smoothness using a penalty parameter requires constructing gene-specific design matrices for different genes in the data, a task that is currently impractical.

Minor comments:

In general, the introduction seems very brief and it would be useful to add more context. A few examples:
- Second paragraph of introduction: it would be helpful for unfamiliar readers to expand what is meant here with 'replicate samples'. The authors are likely thinking about different samples as obtained from different subjects. The pseudobulking approach they use may

not be the best approach if the replicate samples may have been derived from the same subject. In addition, integration across samples may not always be necessary and whether or not to perform this should be carefully evaluated. Minimal sample effects may occur in e.g. studies using multiplexing. It would be good to add this nuance.

We thank the reviewer for the comment. Yes, the 'replicate samples' means different biological replicate samples. We have revised that sentence to avoid confusion. In this workflow, the five samples we used were from five different mice (i.e., not from the same subject).

- Third paragraph of introduction: It would be relevant to specify what is meant with 'the pseudo-bulk method'.

A description of the 'pseudo-bulk method' have been added to the manuscript.

- Fourth paragraph: When would trajectory inference be preferred? Expand how pseudotime is derived from a trajectory and why this is useful.

We mentioned that the trajectory inference is useful for studies that focus on cell differentiation or cell type development. The details of how pseudotime is derived from a trajectory and why it is useful are covered in the section "Trajectory analysis with monocle3" later on.

The last paragraph of the introduction should be partly rephrased:
- "The single-cell level analysis is performed in Seurat, and the trajectory analysis is conducted using monocle3": both of these analyses are 'single-cell level analysis'.
- "The analysis pipeline presented in this article can be applied to any scRNA-seq study with replicate samples." while this is true in theory, many datasets are not suitable for trajectory inference, as the biological context of a dataset may not constitute a 'dynamic system'.

We thank the reviewer for the above two comments. We have now revised the last paragraph of the introduction accordingly.

The paper mentions "The calculation of pseudotime, which indicates the distance between a cell and the starting cell in a trajectory, is conducted during the trajectory learning process", but this is inaccurate. The pseudotime is the distance between a cells' projection on the trajectory and the starting point of the trajectory, as measured along that trajectory. Current phrasing could be misunderstood as Euclidean distance between two points in e.g. the UMAP space.

In the manuscript, we did specifically mention that this is "the distance between a cell and the starting cell in a trajectory".

It may be useful to visualize the spline basis functions to allow the reader to gain intuition on what is happening.

We thank the reviewer for this thoughtful comment. We have provided the visualization of fitted spline curves in the format of line graphs in Fig 11, which gives reader some intuition on what is happening. The visualization of the spline basis functions, on the other hand, would be less intuitive compared to this. This is mainly because the spline coefficients do not have any particular meaning.

Competing Interests: No competing interests were disclosed.

Author Response 20 Nov 2023

Gordon Smyth

I would like to add to Yunshun Chen's comments regarding regression splines vs smoothing splines estimated by cross-validation (CV-splines). In our view, regression splines are more appropriate than CV-splines for this type of workflow. The first consideration is inferential purpose. Methods such as cross-validation or AIC are designed for prediction rather than for interpretation and tend to overfit data from an inferential of view by including terms that do not achieve statistical significance. Regression splines allow us to conduct rigorous and powerful likelihood ratio tests in edgeR, which would be impossible using penalized smoothing splines. A second consideration is practicality. CV-splines are best suited to larger datasets and using CV-splines for datasets with less than 20 residual df would not be reliable. The third consideration is generality. We chose a 3-dimensional basis for our regression splines. The preset dimension (or df) does limit the maximum complexity of curves that can be fitted. In essence, we are assuming that expression trends do not have multiple local maxima, something that would require 4 or more df to accommodate. This limitation was a deliberate decision because we think that more complex trends would rarely be of biological interest. Nevertheless, the workflow could easily accommodate regression splines with df=4 or df=5, which we think would be large enough to accommodate pretty much any smooth trend of biological interest. There is no limitation that the fitted curves for different genes must follow the same shape or have the same smoothness. Our 3-df regression splines can accommodate any trend shape from constant to monotonic to quadratic (up then down) to cubic (up, down, then up again). Classic smoothing splines define smoothness in terms of integrated squared second derivative and regression splines can take on any value for that measure.

Competing Interests: No competing interests were disclosed.

Reviewer Report 03 August 2023

https://doi.org/10.5256/f1000research.147104.r190775

Michael D Morgan
[1] Institute of Medical Sciences, School of Medicine, Medical Sciences and Nutrition, University of Aberdeen, Aberdeen, Scotland, UK
[2] Institute of Medical Sciences, School of Medicine, Medical Sciences and Nutrition, University of Aberdeen, Aberdeen, Scotland, UK

The workflow presented by Cheng *et al.,* seeks to provide a framework for differential gene expression analysis along an inferred cell trajectory from single-cell droplet RNA-sequencing data. Several code snippets are especially useful, for example the summarising and plotting expression of genes in a pathway, and downloading data directly from GEO.

Numerous R packages are used in the workflow, and this presents the first barrier any potential user will face to running the workflow. To address this the authors should include clear instructions at the beginning of the workflow to install all of these different package dependencies, noting which are available through community resources, i.e. CRAN and Bioconductor, vs. those that require installation from work-in-progress repositories. Inclusion of sessionInfo(), while useful, isn't sufficient for newcomers to this type of analysis.

The justification for the workflow is a little weak. This could be significantly strengthened by motivating the work by explaining why Monocle3 was selected over alternative packages that seek to reconstruct a pseudotermporal ordering, especially as the OSCA book describes pseutotime DGE analysis already. The motivation, and manuscript, could be further strengthened by providing concrete examples of the utility of the workflow, e.g. as a teaching tool, or introduction to DGE analysis along a pseudotime ordering for newcomers to the field.

Several statements are made in the current manuscript that aren't strictly correct or are outdated: "an integration method is required to investigate all cells across all samples simultaneously" <- this ignores the common-place use of sample multiplexing to overcome batch effects. "As the cost of scRNA-seq continues to drop" <- what is the real world evidence that the cost of scRNA-seq is falling? Reagent costs rise with inflation (at best). If in reference to the cost of sequencing, this comes at the cost of requiring higher sequencing depths/throughput to achieve lower costs. This should be clarified.

The authors use a dataset with 5 samples, each of which represents a different timepoint. They then proceed to batch integrate these time points, but do not highlight the confounding between development stage and batch which can lead to over-correction of batch effects and remove biological variation. Moreover, I would question whether this truly represents a replicated experiment when the cells from the same sample are not independent. This is important because these cells are pseudobulked by sample and cluster for the DGE analysis and hence are not independent replicates. The authors should discuss this and make it clear that these data are used for illustrative purposes only, and highlight these limitations.

A comment on why the data were selected, and the specific samples would aid the clarity of the manuscript.

The authors note that the samples used different feature annotation versions. The authors should note/describe what barriers this presents to downstream analyses, and if possible, provide a

recommendation on how to resolve the issue. e.g. work from the sequence data and re-process using a harmonised genome build.

The authors use a series of thresholds for quality control of single cells. In reality QC thresholds are highly dependent on the study data, e.g. quiescent or small cells may normally have low RNA expression levels and metabolically active cells may have a high mitochondrial content. The data-dependency of QC thresholds should be noted clearly here. Likewise, the choice to remove cells with large numbers of genes has the potential to remove genuine cells. No justification is chose for these thresholds, and given that doublet detection is performed later it is not clear what this achieves.

The authors state a "standard Seurat analysis" was performed - it is not clear what this means, and is vague particularly for newcomers. This should be clarified with a concrete series of steps described.

In the first Seurat-based analysis the knn-graph and clustering steps are run on each sample separately. There needs to be some justification for these analyses steps otherwise the workflow is not a useful learning tool for newcomers to single-cell analysis.

The authors subset the data using specific clusters. However, the selection of these clusters is not well justified. For instance, in the E18.5-epi sample selecting cluster 1 seems logical when cluster 3 is selected in sample P5. There needs to be a justification for why these clusters were selected otherwise it seems somewhat arbitrary and dependent on how hard one stares at the UMAPs.

It should be noted that several steps in the workflow use algorithms with a random component. Consequently, I get different numbers of cells in each cluster. I suspect 2 possible sources: (1) differences in clustering perhaps due to a random element to the graph building, integration or clustering step, (2) random elements to the doublet detection. These should be stated clearly, and where a random element exists in the relevant algorithms, a seed is set to retain reproducibility between different runs of the same code and data. The downstream consequences are non-trivial. For instance, running the workflow on my local Mac, I detect 268 DEGs vs. the 3268 reported in the manuscript.

Newcomers to the field and this workflow may be unfamiliar with which steps are computationally burdensome - these could be noted in the manuscript and accompanying code to alert users that patience is required at these steps, e.g. FindIntergrationAnchors().

The authors construct a single linear trajectory through the selection EpCam+ cells. Does it even make sense to have a single linear trajectory for a differentiation process that includes a bifurcation point? This analysis doesn't seem appropriate for the composition of the data - separating the two lineages would make more sense especially for the subsequent DGE analysis.

The steps to build the spline should be explained in more detail. e.g. what does the QR decomposition do, what is it's purpose/necessity in terms of finding the smooth linear trends over samples w.r.t. pseudotemporal ordering.

Is the rationale for developing the new method (or application) clearly explained?

Partly

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computational biology, single-cell, genetics, immunology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 23 Oct 2023

Yunshun Chen

We warmly thank the Reviewer for his positive assessment of our work and for the constructive comments that helped us enhance the strength of the manuscript.

Numerous R packages are used in the workflow, and this presents the first barrier any potential user will face to running the workflow. To address this the authors should include clear instructions at the beginning of the workflow to install all of these different package dependencies, noting which are available through community resources, i.e. CRAN and Bioconductor, vs. those that require installation from work-in-progress repositories. Inclusion of sessionInfo(), while useful, isn't sufficient for newcomers to this type of analysis.

We thank the reviewer for pointing this out. We totally agree with the reviewer that clear instructions of package installation should be given at the start of the workflow. We have now added a new section of “Package installation” at the start of the workflow.

The justification for the workflow is a little weak. This could be significantly strengthened by motivating the work by explaining why Monocle3 was selected over alternative packages that seek to reconstruct a pseudotermporal ordering, especially as the OSCA

book describes pseutotime DGE analysis already. The motivation, and manuscript, could be further strengthened by providing concrete examples of the utility of the workflow, e.g. as a teaching tool, or introduction to DGE analysis along a pseudotime ordering for newcomers to the field.

We thank the reviewer for the comment. We did try slingshot and noticed the results from slingshot are not as stable as those from monocle3. Note that we present this workflow not to popularize the use of a particular package, such as monocle3, for trajectory analysis. In fact, we mentioned in the manuscript that "*alternative methods and packages can be used interchangeably with the ones implemented in this study, as long as they perform equivalent functions*." The motivation of this workflow is to showcase a novel approach that utilizes the advanced edgeR GLM framework for a time-course analysis and combines the single-cell level trajectory analysis with the pseudo-bulking strategy.

Several statements are made in the current manuscript that aren't strictly correct or are outdated: "an integration method is required to investigate all cells across all samples simultaneously" <- this ignores the common-place use of sample multiplexing to overcome batch effects.

The integration analysis not only overcomes batch effects, but also accounts for sample effects when assessing different biological samples simultaneously. The sample multiplexing strategy only adjusts the former but not the latter.

"As the cost of scRNA-seq continues to drop" <- what is the real world evidence that the cost of scRNA-seq is falling? Reagent costs rise with inflation (at best). If in reference to the cost of sequencing, this comes at the cost of requiring higher sequencing depths/throughput to achieve lower costs. This should be clarified.

We thank the reviewer for the comment. The overall cost of kits has remained similar but the actual cost per sample has significantly decreased due to the advent of multiplexing technologies such as CellPlex and the Flex assay. We have now clarified that in the manuscript.

The authors use a dataset with 5 samples, each of which represents a different timepoint. They then proceed to batch integrate these time points, but do not highlight the confounding between development stage and batch which can lead to over-correction of batch effects and remove biological variation. Moreover, I would question whether this truly represents a replicated experiment when the cells from the same sample are not independent. This is important because these cells are pseudobulked by sample and cluster for the DGE analysis and hence are not independent replicates. The authors should discuss this and make it clear that these data are used for illustrative purposes only, and highlight these limitations.

We appreciate the comment from the reviewer. Yes, the 5 samples are from different timepoints, and hence the pseudo-bulk samples are not independent replicates. The MDS plot of all the pseudo-bulk samples further confirms the existence of the sample effects. To address this, we revised our downstream DE analysis and now incorporate the sample

effects into the design matrix. Accounting for the sample effects significantly increases the statistical power of the time-course analysis. As expected, we now detect more genes significantly associated with pseudotime.

A comment on why the data were selected, and the specific samples would aid the clarity of the manuscript.

We select this data as we wanted to study the epithelial lineage by constructing a trajectory that models dynamic cellular changes along the lineage. It is also because this is a study we are very familiar with. In general, one could choose any single-cell experiments aimed at studying dynamic changes along a specific path, whether it involves cell differentiation or the development of cell types.

The authors note that the samples used different feature annotation versions. The authors should note/describe what barriers this presents to downstream analyses, and if possible, provide a recommendation on how to resolve the issue. e.g. work from the sequence data and re-process using a harmonised genome build.

We thank the reviewer for the comment. In general, the same cellranger reference build is preferred for consistency, although the effect on the downstream analysis is negligible. We have added a note to the manuscript in this regard.

The authors use a series of thresholds for quality control of single cells. In reality QC thresholds are highly dependent on the study data, e.g. quiescent or small cells may normally have low RNA expression levels and metabolically active cells may have a high mitochondrial content. The data-dependency of QC thresholds should be noted clearly here.

We agree with the reviewer that QC thresholds shall be considered carefully depending on the study data. We choose these thresholds in the workflow because our main focus is the epithelial cell population which contains decent amount of RNA and also with low mitochondrial content if healthy. We have added a note to the manuscript to clarify that the QC thresholds are data dependent.

Likewise, the choice to remove cells with large numbers of genes has the potential to remove genuine cells. No justification is chose for these thresholds, and given that doublet detection is performed later it is not clear what this achieves.

Even though a separate doublet detection analysis is performed using scDblFinder, we notice from our own practise that the combination of both doublet detection and the removal of cells with large counts works the best. This approach is also adopted in Seurat single-cell analysis vignettes [1].

The authors state a "standard Seurat analysis" was performed - it is not clear what this means, and is vague particularly for newcomers. This should be clarified with a concrete series of steps described.

A standard Seurat analysis refers to the standard way of analysing a scRNA-seq data using the Seurat package. We have now revised the 'standard' Seurat workflow part of the manuscript and added more detailed descriptions to some of the steps. We also refer the readers to Seurat online vignettes for more details.

In the first Seurat-based analysis the knn-graph and clustering steps are run on each sample separately. There needs to be some justification for these analyses steps otherwise the workflow is not a useful learning tool for newcomers to single-cell analysis.

We performed standard Seurat analysis on each individual sample separately to get some general idea of each one of them. This is also needed for the downstream analysis such as subsetting the epithelial cell population.

Note that this workflow is not a learning tool for someone who is completely new to single-cell analysis. Even the OSCA book [2] put doublet detection and trajectory analysis in the "Advanced" section (and also the pseudobulk DE analysis in the "Multi-sample" section after "Advanced").

The authors subset the data using specific clusters. However, the selection of these clusters is not well justified. For instance, in the E18.5-epi sample selecting cluster 1 seems logical when cluster 3 is selected in sample P5. There needs to be a justification for why these clusters were selected otherwise it seems somewhat arbitrary and dependent on how hard one stares at the UMAPs.

As mentioned in the manuscript, we are mostly interested in the epithelial cell population which is typically marked by the Epcam gene. However, some other markers of basal, LP and ML cell populations may also be examined and considered. We have now revised that section accordingly.

It should be noted that several steps in the workflow use algorithms with a random component. Consequently, I get different numbers of cells in each cluster. I suspect 2 possible sources: (1) differences in clustering perhaps due to a random element to the graph building, integration or clustering step, (2) random elements to the doublet detection. These should be stated clearly, and where a random element exists in the relevant algorithms, a seed is set to retain reproducibility between different runs of the same code and data. The downstream consequences are non-trivial. For instance, running the workflow on my local Mac, I detect 268 DEGs vs. the 3268 reported in the manuscript.

We appreciate the reviewer's comment. We are fully aware of the random component in the workflow. Therefore, random seeds were set for the analysis steps that involve randomness. The outcomes are fully reproducible if the same versions of R and all the required R packages, as well as the same operating system, are used. The reason that the reviewer got different results are mostly due to the use of different versions of R or R packages. In addition, there is one step that may require a manual inspection if different versions of software are used, that is the selection of the starting node. This is because the node numbers may vary depending on the version of monocle3 used, and using the same node number as the starting node may lead to unexpected results. We have commented on

this in the manuscript.

Newcomers to the field and this workflow may be unfamiliar with which steps are computationally burdensome - these could be noted in the manuscript and accompanying code to alert users that patience is required at these steps, e.g. FindIntergrationAnchors().

We thank the reviewer for the comment. The running time depends on the computational environment and resources as well as the size of the data when running the workflow. We have now included comments to notify users about the expected time needed to complete these steps.

The authors construct a single linear trajectory through the selection EpCam+ cells. Does it even make sense to have a single linear trajectory for a differentiation process that includes a bifurcation point? This analysis doesn't seem appropriate for the composition of the data - separating the two lineages would make more sense especially for the subsequent DGE analysis.

We constructed this single linear trajectory based on the mammary gland epithelial lineage, where mammary stem cells (enriched basal cell population) give rise to luminal progenitor and then become mature luminal. It is not a bifurcating process.

The steps to build the spline should be explained in more detail. e.g. what does the QR decomposition do, what is it's purpose/necessity in terms of finding the smooth linear trends over samples w.r.t. pseudotemporal ordering.

We thank the reviewer for this comment. The QR decomposition was used to re-parametrize the design matrix so that the first coefficient Z1 represents the linear trend in pseudotime. This would allow us to identify genes of which the expression levels increase or decrease along pseudotime in general. Making the DE results 'directional' is essential for GO and KEGG pathway analysis performed later on. We have now added more detailed explanation accordingly.

References:

1. https://satijalab.org/seurat/articles/pbmc3k_tutorial.html
2. https://bioconductor.org/books/release/OSCA/

Competing Interests: No competing interests were disclosed.