Single cell RNA sequencing

Introduction

Jong Kyoung Kim

DGIST

강의교재

Molecules and Cells

Minireview

Dissecting Cellular Heterogeneity Using Single-Cell RNA Sequencing

Yoon Ha Choi and Jong Kyoung Kim*

Department of New Biology, DGIST, Daegu 42988, Korea
*Correspondence: jkkim@dgist.ac.kr
http://dx.doi.org/10.14348/molcells.2018.0446
www.molcells.org

https://scg.dgist.ac.kr/wp-content/uploads/2019/02/MaC2018-12-446R1.pdf

실습 자료

https://github.com/jsleescgl/BIML-2019-SingleCellRNAseq-

Single cell: Limited materials

Uncultivated microbial species

Rare cell types

Early development

Single cell: Heterogeneity

Tumour heterogeneity

Tissue heterogeneity

The growth of scRNA-seq techniques

Development (2017) 144:17

Technologies of scRNA-seq

The first paper: Tang protocol

Table 1 | Single cell mRNA-Seq mapping summary

	Blastomere (50-base reads)	Blastomere (35-base reads)	Wild-type oocyte 1 (50-base reads)	Wild-type oocyte 2 (50-base reads)	*Dicer1*$^{-/-}$ oocyte 1 (50-base reads)	*Dicer1*$^{-/-}$ oocyte 2 (50-base reads)	*Ago2*$^{-/-}$ oocyte (50-base reads)
Reads processed	85,807,979	24,424,339	76,584,432	20,998,366	65,023,554	37,652,933	43,311,094
Known RefSeq transcripts[a]	20,677,262	6,534,175	33,038,025	9,334,408	23,459,955	11,121,519	19,179,784
Reads with 0 mismatches to 21,436[b] RefSeq transcripts	9,166,378	4,471,940	13,960,414	5,071,973	10,059,424	6,588,271	7,873,465
Reads with 1 mismatch to 21,436 RefSeq transcripts	5,733,123	1,741,229	9,898,861	2,382,288	6,917,296	2,658,557	6,300,824
Reads with 2 or more mismatches to 21,436 RefSeq transcripts	5,777,761	321,006	9,178,750	1,880,147	6,483,235	1,874,691	5,005,495
Reads matching mouse genome[c]	49,910,707	14,896,842	50,320,364	14,623,581	37,725,544	21,299,168	31,861,036
Unmatched reads	35,489,449	9,507,906	26,084,368	6,003,943	26,741,547	14,472,985	11,273,117
Reads matching filtered sequences (primers, rRNAs)	407,823	19,591	179,700	370,842	556,463	1,880,780	176,941
Splice junction reads	2,976,501	1,238,737	2,395,418	922,656	1,818,176	1,170,273	1,337,043

Nat Methods (2009) 6:377

Early protocols

Smart-Seq

Nat Biotechnol (2012) 30:777

Quartz-Seq

Genome Biol (2013) 14:R31

Experimental challenges in scRNA-seq

Multi-omics

Capturing single cells

Cell lysis and mRNA enrichment

Reverse transcription

Amplification

NGS

Multiplexing by cellular barcoding

STRT-seq

n=85

"In brief, each sample was prepared by picking single cells into the wells of a 96-well PCR plate preloaded with lysis buffer and then by adding reverse transcription reagents to generate a first-strand cDNA. … A different helper oligo was used in each well, with distinct six-base barcodes and a universal primer sequence. After cDNA synthesis, the 96 reactions were pooled, purified, and amplified by single-primer PCR in a single tube."

Genome Res (2011) 21:1160

Capturing single cells

	Micro-manipulation	LCM	FACS	Micro-fluidics	Micro-droplets
# of cells	10s	10s	100s	100s	10000s
Dissociated?	Any tissue	Any tissue	Dissociated	Dissociated	Dissociated

Kolodziejczyk, Kim et al. (2015) **Mol Cell**

Microfluidics: Fluidigm C1

Nature Biotechnology (2014) 32:1053

Microfluidics: Microdroplet

Cell (2015) 161:1187

Capturing single cells: Microwells

Nat Methods 14:395 (2017)

Cell 172:1091 (2018)

Capturing single cells: SPLiT-seq

Science 360:176 (2018)

Scaling scRNA-seq experiments

Nat Protoc 13:599 (2018)

Experimental challenges in scRNA-seq

Multi-omics

Capturing single cells

Cell lysis and mRNA enrichment

Reverse transcription

Amplification

NGS

Technical noise in scRNA-seq

Brennecke, Anders*, Kim* et al (2013)* ***Nature Methods***

Technical noise: stochastic mRNA loss

Capture efficiency	20%	40%	60%

Technical noise: amplification

Amplification factor	2	3	5

Minimizing amplification bias by molecular barcodes

Molecular barcode

Read count	8	8
Molecular count	2	4

Nat Methods 11:163 (2014)

Further improvements for sensitivity

RT enzymes	Buffer conditions	Primers	Amplification steps	Reaction volume
- Optimization of RT enzymes	- Optimization of the buffer and temperature in the RT reaction	- Shorted RT primers	- Optimization of the cDNA amplification step (# of cycles, reaction conditions, polymerases)	- Implementing nanoliter reactors in a microfluidics device - Adding macromolecular crowding agents
- CEL-Seq2 (Genome Biol 17:77 (2016))	- Quartz-Seq2 (Genome Biol 19:29 (2018))	- CEL-Seq2 (Genome Biol 17:77 (2016))	- mcSCRB-seq (Nat Commun 9:2937 (2018))	- CEL-Seq2 (Genome Biol 17:77 (2016)) - mcSCRB-seq (Nat Commun 9:2937 (2018))

Experimental challenges in scRNA-seq

Multi-omics

Capturing single cells

Cell lysis and mRNA enrichment

Reverse transcription

Amplification

NGS

Single-cell multi-omics

Advantages and limitations

- scDNA-seq and scRNA-seq protocols are easily modified
- Amenable to bisulfite sequencing
- Potential loss of molecules

- scDNA-seq and scRNA-seq protocols are easily modified
- Amenable to bisulfite sequencing
- Loss of cytoplasmic and all nuclear mRNA molecules

- Minimal risk of loss of molecules (as preamplified in the first vessel)
- cDNA contamination in gDNA library
- Not amenable to bisulfite sequencing

- Amenable to RNA and protein measurements in parallel
- Significant loss of molecules

Annu Rev Genomics Hum Genet 19:15 (2018)

Computational challenges

scRNA-seq

High technical noise

statistics

High biological noise

mathematical modelling

Big data

machine learning

Workflow for analyzing scRNA-seq data

Data generation

- **Barcode processing:** trimming CB and UMI
- **Read mapping:** mapping reads to the genome or transcriptome
- **Gene counting:** demultiplexing CB and collapsing UMI
- **Cell filtering:** discarding low-quality cells

Data processing

- **Normalization:** estimating cell-specific global scaling factors
- **Imputation:** denoising sparse counts
- **Feature selection:** selecting informative genes

Exploratory Analysis

- **Visualization:** finding a low-dimensional representation preserving relevant structures

Heterogeneity Analysis

- **Clustering:** labeling cells with a discrete cell type or state
- **Trajectory inference:** labeling cells with a continuous pseudotime for differentiation trajectories

Number of tools over time

https://www.scrna-tools.org/analysis

Platforms

https://www.scrna-tools.org/analysis

Categories

https://www.scrna-tools.org/analysis

Unique features of scRNA-seq data analysis

- Sparse count matrix
- Large number of cells (> 10,000)
- Unsupervised analysis
- Single-cell specific biological questions

R/Bioconductor packages

1. DropletUtils
Usage: pre-processing, QC
URL: https://bioconductor.org/packages/release/bioc/html/DropletUtils.html

2. Scater
Usage: pre-processing, QC, normalization, and visualization
URL: https://bioconductor.org/packages/release/bioc/html/scater.html

3. Scran
Usage: normalization, highly variable genes, cell-cycle phase assignment
URL: http://bioconductor.org/packages/release/bioc/html/scran.html

4. Seurat
Usage: dimension reduction, differential expression, visualization
URL: http://satijalab.org/seurat/

5. Destiny
Usage: pseudo-time analysis
URL: https://bioconductor.org/packages/release/bioc/html/destiny.html

6. Monocle2
Usage: Pseudo-time analysis
URL: https://bioconductor.org/packages/release/bioc/html/monocle.html

Single cell RNA-seq

Data generation

Jong Kyoung Kim

Overview

Data generation

- **Barcode processing:** trimming CB and UMI
- **Read mapping:** mapping reads to the genome or transcriptome
- **Gene counting:** demultiplexing CB and collapsing UMI
- **Cell filtering:** discarding low-quality cells

Barcode processing

Cell barcode

UMI

```
@GSE63473_ERCC.223:CELL_AACATAGCGAAT:UMI_GCCAAGGT
ACAAATTCAAGCCGCTCGCGAGATTACCTCGAGAAGCAACTAAGCCCGATCAAGGAAGAT
+
.A))<<)FA)F)F.AF).)FF)FFFF.FF).FF7<<F.)FF.A..F))AF7))A.F.7<F
```

Read mapping

Input sam file

```
GSE63473_ERCC.512:CELL_AGGAGCAGTCGT:UMI_TTGATCCT        0       ERCC-00074      278     37      60M     *       0       0       ACTCATAACCCCAGCAC
TAAGATATTTTAAAGAGGCATCTATCACATAAGGCATCATTAT     )AA.7FFA7F.<..FFFF).F)<F)7..7F.FFF<FFFFF.FFFFF<F<FFFFF7.F..)    MD:Z:18T41      NH:i:1  HI:i:1  N
M:i:1   SM:i:37 XQ:i:40 X2:i:0  XO:Z:UU PG:Z:A
GSE63473_ERCC.516:CELL_CCCGCGGCTATT:UMI_GTGCGTGA        0       ERCC-00096      996     37      60M     *       0       0       GAGGGACTCATCCTGGG
CTACAATCCTATTGCCGAGATAGTATTTCTTAGCTTCCTGAGG     AAAA.AFAFFAFFFF.FFAFFF<FFAFF<FFFFAFF.FF.FFFFFFFFFFFFFFFFAFF    MD:Z:60 NH:i:1  HI:i:1  NM:i:0  S
M:i:37  XQ:i:40 X2:i:0  XO:Z:UU PG:Z:A
GSE63473_ERCC.522:CELL_ATTTGCACGTTA:UMI_GGCGGAAG        0       ERCC-00096      814     37      60M     *       0       0       ATTGGAAGCATCAACCT
GCGCCGTCTTGTTAACTTGTCATATCGCGCACGTAGTAGCCTA     <A7)A.FFFFFFA7FFFFFFFFFFFFF7FFFFFFF<FF<AFFF7F.FFFFF7<F7FFFFF    MD:Z:60 NH:i:1  HI:i:1  NM:i:0  S
M:i:37  XQ:i:40 X2:i:0  XO:Z:UU PG:Z:A
GSE63473_ERCC.379:CELL_CAGTCCACGCGC:UMI_AGGGGGGG        0       ERCC-00096      722     37      57M3S   *       0       0       CGGAAGATACGCTCTAA
GCTCGGCAATCGCTCTTGCCGTGCGAGCTGCTAGCAAAAAAAA     <<<A7A)FFFFFAAF.)7AAAFFFF)<FAFAAFFAFFAFAFA.F)FA.F<<FFFFFFFF7    MD:Z:44A12      NH:i:1  HI:i:1  N
M:i:1   SM:i:37 XQ:i:40 X2:i:0  XO:Z:UU PG:Z:T
```

Raw UMI count table

Sequencing error in UMI

ED = 1
AAAAAAAA
AAACAAAA

Sequencing error in UMI

No merging

Merging UMIs with ED <=1

Tools for barcode processing

Name	Reference	Open source	Quality filter	UMI collapsing	Mapper	BC detection	Intron	Down-sampling	Compatible UMI library protocols
Cell Ranger	[2]	yes	BC+UMI	Hamming distance	STAR	A	no	yes	[2]
CEL-seq	[15]	yes	BC+UMI	Identity only	bowtie2	WL	no	no	[15, 46]
dropEst	[16]	yes	BC	Frequency-based	TopHat2 or Kallisto	WL,top-n,EM	yes	no	[2, 13, 19]
Drop-seq-tools	[13]	no	BC+UMI	Hamming distance	STAR	WL,top-n	no	no	[13, 15, 17]
scPipe	[47]	yes	BC+UMI	Hamming distance	subread	WL,top-n	no	no	[13, 17, 18, 46]
umis	[14]	yes	BC	Frequency-based	Kallisto	WL,top-n,EM	no	no	[2, 13, 17–19, 46, 48]
UMI-tools	[25]	yes	BC+UMI	Network-based	BWA	WL	no	no	[17, 19]
zUMIs	This work	yes	BC+UMI	Hamming distance	STAR	A,WL,top-n	yes	yes	[2, 3, 12, 13, 15, 17, 18, 21, 46, 48]

Gigascience 7:1 (2018)

Cell filtering

of unique cell index: 812,692

Minimum Phred quality score withi
n the cell barcode >= 10

DropletUtils

1. The set *G* of all barcodes with total counts less than or equal to *T (100)* are considered to represent empty droplets.

2. Estimate the posterior expectation of the proportion of counts

$$\tilde{p}_g \longleftarrow A_g = \sum_{b \in \mathcal{G}} y_{gb}$$

3. Null hypothesis: free-floating transcripts in solution are randomly encapsulated into the empty droplets.

https://www.biorxiv.org/content/10.1101/234872v2

DropletUtils

4. The likelihood of obtaining the counts for barcode b (from Dirichlet-multinomial distribution) is

$$L_b = \frac{t_b!\Gamma(\alpha)}{\Gamma(t_b+\alpha)} \prod_{g=1}^{N} \frac{\Gamma(y_{gb}+\alpha_g)}{y_{gb}!\Gamma(\alpha_g)} \qquad \alpha_g = \alpha\tilde{p}_g$$

5. Compute the P-value for b using a Monte Carlo approach.

6. Retain all barcodes with $t_b \geq U$ (knee point) or FDR < 0.1%.

https://www.biorxiv.org/content/10.1101/234872v2

Classification of low-quality cells

Illic, Kim* et al (2016)* ***Genome Biology***

Overview of the pipeline

Feature selection

Feature selection

Feature selection

Hidden bad-quality cells

Hidden bad-quality cells

Hidden bad-quality cells

Dataset 1

Single cell RNA-sequencing

Nat Commun 8:14049 (2017)

Scater

The **scater** package is contains tools to help with the analysis of single-cell transcriptomic data, with the focus on RNA-seq data. The package features:

1. Use of the **SingleCellExperiment** class as a data container for interoperability with a wide range of other Bioconductor packages;
2. Wrappers to kallisto and Salmon for rapid quantification of transcript abundance and tight integration with scater;
3. Simple calculation of many **quality control metrics** from the expression data;
4. Many tools for visualising scRNA-seq data, especially **diagnostic plots for quality control**;
5. Subsetting and many other methods for **filtering out problematic cells and features**;
6. Methods for identifying important experimental variables and normalising data ahead of downstream statistical analysis and modeling.

Bioinformatics 33:1179 (2017)

QC: Dataset1

- Empty Droplet filtering

```
set.seed(100)
e.out <- emptyDrops(my.counts)
is.cell <- e.out$FDR < 0.01
sum(is.cell, na.rm=TRUE)
sce <- sce[,which(e.out$FDR < 0.01)]
```

QC: Dataset1

- Empty Droplet filtering

QC: Dataset1

- Quality control with mitochondrial genes

```
## Quality control on the cells ##
# MT gene QC
library(scater)
library(EnsDb.Hsapiens.v86)

location <- mapIds(EnsDb.Hsapiens.v86, keys=rowData(sce)$ID,
                   column="SEQNAME", keytype="GENEID")

rowData(sce)$CHR <- location
summary(location=="MT")
sce <- calculateQCMetrics(sce, feature_controls=list(Mito=which(location=="MT")))
```

QC: Dataset1

- Quality control with mitochondrial genes

Before 2nd QC filter

QC: Dataset1

- Quality control with total UMI counts

QC: Dataset1

- Quality control with detected features

QC: Dataset1

- Quality control with mitochondrial proportions

QC: Dataset1

```
Filter cutoff
log10_total_counts > 2 &
pct_counts_Mito <= 10 &
log10_total_features_by_counts > 2
```

- Quality control with multiple filters

QC: Dataset1

- Quality control with multiple filters

Further analysis with cells which passed 2nd filter

Single cell RNA-seq

Data processing

Jong Kyoung Kim

Overview

Data processing

- **Normalization:** estimating cell-specific global scaling factors
- **Imputation:** denoising sparse counts
- **Feature selection:** selecting informative genes

Dataset 2

nature immunology

ARTICLES

https://doi.org/10.1038/s41590-018-0294-9

Memory CD4+ T cells are generated in the human fetal intestine

Na Li[1], Vincent van Unen[1], Tamim Abdelaal[2,3], Nannan Guo[1], Sofya A. Kasatskaya[4,5], Kristin Ladell[6], James E. McLaren[6], Evgeny S. Egorov[4], Mark Izraelson[4], Susana M. Chuva de Sousa Lopes[7], Thomas Höllt[2,8], Olga V Britanova[4], Jeroen Eggermont[9], Noel F. C. C. de Miranda[10], Dmitriy M. Chudakov[4,5,11,12,13,14], David A. Price[6,15], Boudewijn P. F. Lelieveldt[3,9] and Frits Koning[1*]

Dataset 2

Legally aborted fetus
(Intestine)

Sorting only CD4+ T cells

Single cell RNA-sequencing

Normalization

- Estimate the true expression level of each gene in each cell by removing cell-specific biases in the count matrix

- Assumption: the expected count of a gene in a cell is proportional to the product of the relative expression level of the gene and the cell-specific global scaling factor.

- Global scaling factor: cell-specific systematic biases affected by cell-to-cell differences in cell size, capture and RT efficiency, amplification factor, dilution factor, and sequencing depth.

Normalization

Nat Methods 14:565 (2017)

Normalization

	Cell-specific effects	Gene-specific effects	Not removed by UMIs
Sequencing depth	✓		✓
Amplification	✓	✓	
Capture and RT efficiency	✓	✓	✓
Gene length		✓	
GC content	✓	✓	✓
mRNA content	✓		✓

Nat Methods 14:565 (2017)

Global scaling factor

Nat Methods 14:565 (2017)

Cell-specific biases

- Cell-specific biases can be removed by normalizing the raw counts within each cell by a single scaling factor, applied to all genes in a cell.

- The cell-specific scaling factor can be estimated based on:
  - Library size: RPM, TPM (sensitive to a few highly expressed genes)
  - Upper quantile values of counts
  - Normalization factors: size factor of DESeq, trimmed mean of M-value of edgeR (problematic for sparse scRNA-seq data)

High frequency of zeros in scRNA-seq

DESeq size factor

$$\hat{s}_j = \underset{i}{\text{median}} \frac{k_{ij}}{\left(\prod_{v=1}^{m} k_{iv}\right)^{1/m}}$$

Pooling-base size factor

Single cell

All cells (averaged to make a reference pseudo-cell)

Cell pool A:

$\theta_1 + \theta_2 + \theta_3 + \theta_4 = \theta_A$

Cell pool B:

$\theta_5 + \theta_6 + \theta_7 + \theta_8 = \theta_B$

System of linear equations:

$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\ldots \\ 0\,0\,0\,0\,1\,1\,1\,1\ldots \\ 1\,0\,1\,0\,1\,0\,1\,0\ldots \\ 0\,1\,1\,0\,1\,1\,0\,0\ldots \\ \ldots \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \ldots \end{bmatrix} = \begin{bmatrix} \theta_A \\ \theta_B \\ \theta_C \\ \theta_D \\ \ldots \end{bmatrix}$$

Genome Biology 17:75 (2016)

Scater functions

- Normalization
  - Log2

```
assay(umi, "logcounts_raw") <- log2(counts(umi) + 1)
```

  - Log2-CPM

```
assay(umi, "logcounts_cpm") <- log2(calculateCPM(umi, use.size.factors = FALSE) + 1)
```

  - CPM

```
assay(umi, "cpm") <- calculateCPM(umi, use.size.factors = FALSE)
```

  - TPM

```
assay(umi, "tpm") <- calculateTPM(umi, effective_length = 5e4)
```

scran

- Implements a variety of low-level analyses of **single-cell** RNA-seq data.
- Methods are provided for
  - **normalization of cell-specific biases**
  - assignment of cell cycle phase
  - **detection of highly variable**
  - significantly correlated genes

Normalization by scran

- Compute size factor

```
library(scran)
clusters <- quickCluster(sce, method="igraph")
sce <- computeSumFactors(sce, cluster=clusters)
```

Normalization by scran

- Library size vs Size factor plot

```
plot(sizeFactors(filteredSCEset), (filteredSCEset$total_counts)/1000, log="xy",
     ylab="Library size (kilo)", xlab = "Size factor")
```

Normalization by scran

- Normalize()

```
scesetFiltered <- normalize(scesetFiltered)
```

※ Logcounts changes

- Access to count tables
  - normalized

```
scater::exprs(scesetFiltered)
```

  - Original count table

```
scater::counts(scesetFiltered)
```

Imputation

- A high frequency of zero counts is driven by:
  - Stochastic gene expression
  - Low mRNA capture efficiency
  - Low sequencing depth

- This zero inflation leads to high technical variability in gene expression.

- Because global scaling normalization methods are unable to address this issue, computational approaches that recover the true expression levels of zero counts have been proposed.

Imputation

Nat Commun 9:997 (2018)

Imputation

- These imputation methods take a normalized count matrix (usually log-transformed) as input and replace input data with de-noised values, estimated by borrowing information across similar cells or genes.

- Imputed expression values can be used to
  - Recover regulatory interactions between genes
  - Increase the accuracy of estimates of cell-to-cell variability in gene expression
  - Improve cell clustering and differential gene expression analysis.

Imputation

- It should be noted that all such methods introduce unexpected biases, including spurious gene-to-gene correlations, artificial cell subpopulation structure, and removal of rare cell types and transient cell states.

- Imputation should be applied with caution and is not included in the general workflow of scRNA-seq data analysis.

Estimating technical noise with spike-ins

of transcripts (or concentration) available for sequencing

Observed read count

$$Var[K_{ij}] = \underbrace{E\left[Var[K_{ij}|Z_{ij}]\right]}_{\text{Poisson noise}} + \underbrace{\mathrm{Var}\left[\mathrm{E}[K_{ij}|Z_{ij}]\right]}_{\text{Technical noise generated by } Z_{ij}}$$

Estimating technical variance: making a parametric assumption

$$Var[K_{ij}] = a\mu_i + b\mu_i^2$$

$$CV^2[K_{ij}] = \mathrm{a}\frac{1}{\mu_i} + b$$

Brennecke, Anders*, Kim* et al (2013)* ***Nature Methods***

Estimating technical variance: making a parametric assumption

$$E[Z_{ij}] = X_i, Var[Z_{ij}] = aX_i + bX_i^2$$

$$K_{ij}|Z_{ij} \sim Poisson(s_j Z_{ij})$$

$$\begin{aligned} Var[K_{ij}] &= E\left[Var[K_{ij}|\mathrm{Z_{ij}}]\right] + \mathrm{Var}\left[\mathrm{E}[K_{ij}|\mathrm{Z_{ij}}]\right] \\ &= E[s_j Z_{ij}] + Var[s_j Z_{ij}] \\ &= (s_j + as_j^2)X_i + bs_j^2 X_i^2 \end{aligned}$$

Estimating biological noise with spike-ins

of transcripts (or concentration) of gene *i* in cell *j*

# of transcripts (or concentration) available for sequencing

Observed read count

$$Var[K_{ij}] = \underbrace{E\left[Var[K_{ij}|Z_{ij}, X_{ij}]\right]}_{\text{Poisson noise}} + \underbrace{E\left[Var\left[E[K_{ij}|Z_{ij}, X_{ij}]|X_{ij}\right]\right]}_{\text{Technical noise generated by } Z_{ij}} + \underbrace{\mathrm{Var}\left[\mathrm{E}[K_{ij}|X_{ij}]\right]}_{\text{Biological noise}}$$

Estimating biological variance: making a parametric assumption

$$E[X_{ij}] = X_i, Var[X_{ij}] = c_i X_i^2$$

$$E[Z_{ij}|X_{ij}] = \theta_j X_{ij}, Var[Z_{ij}|X_{ij}] = a\theta_j X_{ij} + b\theta_j^2 X_{ij}^2$$

$$K_{ij}|Z_{ij} \sim Poisson(s_j Z_{ij})$$

Estimating biological variance: making a parametric assumption

Brennecke, Anders*, Kim* et al (2013)* ***Nature Methods***

Detecting highly variable genes

- To plug-in the estimated technical variability of spike-ins into that of endogenous genes, we make an assumption that the technical variance of spike-ins is a nonlinear function of their mean expression levels.

- By fitting a curve to the mean-variance data of spike-ins using a nonlinear regression function, we can estimate the average technical variance of each endogenous gene at the given mean expression level.

- The biological variance of endogenous genes can be estimated by subtracting the average technical variance from the total observed variance.

Detecting highly variable genes

- If spike-ins have poor quality or are not available, we directly estimate the mean-variance trend of technical noise from endogenous genes by assuming that most genes are dominated by technical variability.

- To test whether the biological variance is equal to a specified value (usually set to 0), the ratio of the sample variance of the normalized counts to the expected variance under the null hypothesis is used as a test statistic.

- The test statistic under the null hypothesis follows a chi-squared distribution if we assume that the normalized counts follow a normal distribution.

- To make this normality assumption more reasonable, some method (e.g. scran) take log-transformed normalized counts as their input expression values.

Quality of spik-ins

Mean-variance plot

Visualization with HVGs

Detecting highly variable genes

- Highly variable genes

```
### Select Highly variable genes (feature selection)
var.fit <- trendVar(filteredSCEset, parametric=TRUE, use.spikes=FALSE)#, design = batchDesign)
var.out <- decomposeVar(filteredSCEset, var.fit)
hvg <- var.out[which(var.out$FDR < 0.05 & var.out$bio > .01),]
dim(hvg)
```

Detecting highly variable genes

- Highly variable genes

```
### Select Highly variable genes (feature selection)
var.fit <- trendVar(filteredSCEset, parametric=TRUE, use.spikes=FALSE)#, design = batchDesign)
var.out <- decomposeVar(filteredSCEset, var.fit)
hvg <- var.out[which(var.out$FDR < 0.05 & var.out$bio > .01),]
dim(hvg)
```

Reference

Chapter 3

Analysis of Technical and Biological Variability in Single-Cell RNA Sequencing

Beomseok Kim, Eunmin Lee, and Jong Kyoung Kim

https://dx.doi.org/10.1007/978-1-4939-9057-3_3

Single cell RNA-seq

Exploratory analysis

Jong Kyoung Kim

Overview

Exploratory Analysis

- **Visualization:** **finding a low-dimensional representation preserving relevant structures**

Dimensionality reduction

cell

gene

$X \in \mathbb{R}^{D \times N}$

$x_i \in \mathbb{R}^D$

$z_i \in \mathbb{R}^2$

Dimensionality reduction

- Dimensionality reduction is performed to find a low-dimensional representation that preserves the relevant structure of the original high-dimensional data.

- Two different relevant structures are considered
  - A local structure that preserves cell-to-cell distance within a local neighborhood of cells
  - A global structure that preserves cell-to-cell distance on the low-dimensional manifold associated with the underlying biological process

Local and global structure

- Capturing local structure in a low-dimensional representation is important for clustering cells of the same type or state close together.

- Capturing global structure is useful for preserving distance between clusters and revealing underlying biological processes for cell-to-cell variability in gene expression.

Algorithms

PCA
diffusion map
UMAP
t-SNE
Global structure
Local structure

Principal Component Analysis (PCA)

Cell	Radial	Longitudinal
QC1	AGL 42	1
QC2	AGL 42	1
QC3	AGL 42	1
QC4	AGL 42	1
QC5	AGL 42	1
QC6	AGL 42	1
GL2-1	GL2	3
GL2-2	GL2	4
GL2-3	GL2	4
GL2-4	COBL9	4
GL2-5	GL2	4
GL2-6	GL2	4
GL2-7	COBL9	4

Brennecke, Anders*, Kim* et al (2013)* ***Nature Methods***

Kolodziejczyk, Kim* et al (2013)* ***Cell Stem Cell***

Principal Component Analysis (PCA)

$$w_j^T w_j = 1$$

Principal Component Analysis (PCA)

Suppose we want to find an orthogonal set of L linear basis vectors $w_j \in \mathbb{R}^D$, and the corresponding scores $z_i \in \mathbb{R}^L$, such that we minimize the reconstruction error

$$J(W, Z) = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2$$

where $\hat{x}_i = W z_i$ and $W \in \mathbb{R}^{D \times L}$ is a $D \times L$ matrix with the w_j in its columns and $W^T W = I$.

Principal Component Analysis (PCA)

The optimal solution is given by

$$W = V_L$$

where V_L contains the L eigenvectors with largest eigenvalues of the empirical covariance matrix

$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$$

if we assume that $\frac{1}{N}\sum_{i=1}^{N} x_i = 0$.

Principal Component Analysis (PCA)

Furthermore, the optimal low-dimensional encoding of the data is given by

$$z_i = W^T x_i$$

which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

PCA: mean centering and unit variance

$$mean = 0 \; and \; sd = 1$$

- Mean centering is necessary for ensuring that the first PC describes the direction of maximum variance.

- Standardizing is necessary for ensuring that PCA is not misled by directions in which the variance is high merely because of the measurement scale.

PCA: terms

- z_i: principal component scores
- w_j: principal components, principal directions or loadings

PCA: Principal component loadings

gene

cell $\tilde{X}^T \in \mathbb{R}^{N \times D}$

PC

cell $Z \in \mathbb{R}^{N \times L}$

$$Corr\left(\tilde{X}^T, Z\right) = Cov(\tilde{X}^T, ZD^{-\frac{1}{2}})$$

where $D = diag(\lambda_1, \lambda_2, \ldots, \lambda_L)$.

PCA: Principal component loadings

$$Cov\left(\tilde{X}^T, ZD^{-\frac{1}{2}}\right) = \frac{1}{N}\tilde{X}ZD^{-\frac{1}{2}} = \frac{1}{N}\tilde{X}\tilde{X}^T WD^{-\frac{1}{2}} = \hat{\Sigma}WD^{-\frac{1}{2}}$$

$$= WDW^T WD^{-\frac{1}{2}} = WD^{\frac{1}{2}}$$

Correlations between the principal component scores and the original variables

PCA: Principal component loadings

Let $\mathrm{S} = WD^{\frac{1}{2}}$ and $L = D$.

$$\mathrm{S}S^{\mathrm{T}} = WDW^{T} = \hat{\Sigma}$$

We can see that the general expression for the variance accounted for in the kth gene by is

$\sum_{j=1}^{C} s_{kj}^{2}$ the first C PCs

PCA: Choosing the number of PCs

The next question is how many components should be retained.

One criterion is the reconstruction error:

$$E(X, L) = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|^2 = \sum_{j=L+1}^{D}\lambda_j$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$.

PCA: Choosing the number of PCs

Scree plot: plotting the retained eigenvalues in decreasing order.

We look for an “elbow” in the curve and retain only those components that are above the elbow.

Unfortunately, in practice the identification of the elbow is not so clean.

PCA: Choosing the number of PCs

Explained variance (total): A related quantity is the fraction of variance explained:

$$F(X, L) = \frac{\sum_{j=1}^{L} \lambda_j}{\sum_{j'=1}^{D} \lambda_{j'}}$$

PCA: Choosing the number of PCs

Explained variance (gene): the fraction of variance explained in the kth gene by the first L PCs :

$$F(X_k, L) = \frac{\sum_{j=1}^{L} s_{kj}^2}{\sum_{j'=1}^{D} s_{kj'}^2}$$

We might impose the following type of constraint on our analysis. “We must retain a sufficient number of components so that we can account for at least 50% of the variance in each gene.”

Limitations of PCA

PCA focuses on keeping the low-dimensional representations of dissimilar points far apart.

For high-dimensional data that lies on or near a low-dimensional, non-linear manifold it is usually more important to keep the low-dimensional representations of very similar points close together.

This is not possible with PCA.

Stochastic Neighborhood Embedding (SNE)

$$\|x_i - x_j\|_2 \longrightarrow p_{j|i}$$

$$\Big\updownarrow \quad \min_Z \sum_{i=1}^{N} KL(P_i|Q_i)$$

$$\|z_i - z_j\|_2 \longrightarrow q_{j|i}$$

https://cs.nyu.edu/~roweis/papers/sne_final.pdf

SNE: $p_{j|i}$

conditional probability that x_i would pick x_j as its neighbor if neighbors were picked in proportional to their probability density under a Gaussian centered at x_i.

$$p_{j|i} = \frac{e^{-\frac{\|x_j - x_i\|_2^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|x_k - x_i\|_2^2}{2\sigma_i^2}}} \qquad p_{i|i} = 0$$

SNE: $q_{j|i}$

conditional probability that z_i would pick z_j as its neighbor if neighbors were picked in proportional to their probability density under a Gaussian centered at z_i.

$$q_{j|i} = \frac{e^{-\|z_j - z_i\|_2^2}}{\sum_{k \neq i} e^{-\|z_k - z_i\|_2^2}} \qquad q_{i|i} = 0$$

SNE: $\mathrm{J}(\mathrm{Z}) = \sum_{i=1}^{N} KL(P_i|Q_i)$

We aim to find Z that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$.

SNE minimizes the sum of Kullback-Leibler divergences over all x_i using a gradient descent method.

$$\mathrm{J}(\mathrm{Z}) = \sum_{i=1}^{N} KL(P_i|Q_i) = \sum_{i=1}^{N}\sum_{j=1}^{N} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

SNE: $\mathrm{J}(\mathrm{Z}) = \sum_{i=1}^{N} KL(P_i|Q_i)$

$$\mathrm{J}(\mathrm{Z}) = \sum_{i=1}^{N}\sum_{j=1}^{N} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Large $\mathrm{J}(\mathrm{Z})$ if a small $q_{j|i}$ to model a large $p_{j|i}$

Small $\mathrm{J}(\mathrm{Z})$ if a large $q_{j|i}$ to model a small $p_{j|i}$

↓

SNE focuses on keeping the low-dimensional representations of very similar points close together.

SNE: Perplexity

We should select σ_i of the Gaussian that is centered over each x_i.

In dense regions, a small value of σ_i is usually more appropriate than in sparser regions.

SNE performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity that is specified by the user.

SNE: Perplexity

The perplexity is defined as

$$Perp(P_i) = 2^{H(P_i)}$$

where $H(P_i)$ is the entropy of P_i

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

The perplexity increases monotonically with the variance σ_i and can be interpreted as a smooth measure of the effective number of neighbors.

t-distributed stochastic Neighborhood Embedding (t-SNE)

The cost function used by t-SNE differs from the one used by SNE in two ways:

(1) it uses a symmetrized version of the SNE cost function with simpler gradients

(2) it uses a Student-t distribution rather than a Gaussian to compute the similarity between two points in the low-dimensional space.

JMLR 9:2579 (2008)

t-SNE

$$\|x_i - x_j\|_2 \longrightarrow p_{ij}$$

$$\Big\updownarrow \quad \min_Z \sum_{i=1}^{N} KL(P|Q)$$

$$\|z_i - z_j\|_2 \longrightarrow q_{ij}$$

t-SNE: q_{ij}

$$q_{ij} = \frac{\left(1 + \left\|z_i - z_j\right\|_2^2\right)^{-1}}{\sum_{k \neq l}(1 + \|z_k - z_l\|_2^2)^{-1}} \qquad q_{ii} = 0$$

$$\downarrow$$

$$q_{ij} = q_{ji}$$

t-SNE: p_{ij}

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$$

$$p_{ii} = 0$$

t-SNE: crowding problem

t-SNE: gradient

$$\frac{\partial J(Z)}{\partial z_i} = 4 \sum_j \left(p_{ij} - q_{ij}\right)\left(z_i - z_j\right)\left(1 + \left\|z_i - z_j\right\|_2^2\right)^{-1}$$

t-SNE: algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=1 **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$
  end
end

t-SNE for single-cell data

Nat Biotechnol 31:545 (2013)

UMAP

Nat Biotechnol 37:38 (2019)

UMAP

Nat Biotechnol 37:38 (2019)

Single cell RNA-seq

Heterogeneity analysis

Jong Kyoung Kim

Overview

Heterogeneity Analysis

- **Clustering:** **labeling cells with a discrete cell type or state**
- **Trajectory inference:** **labeling cells with a continuous pseudotime for differentiation trajectories**

Dissecting cellular heterogeneity

- Two computational approaches:
  - Discrete latent variable approach that labels each cell with a discrete cluster indicator for cell type or state
  - Continuous latent variable approach that labels each cell with a continuous pseudotime for biological processes (e.g. differentiation)

Discrete latent variable approach

- Formulated as an unsupervised clustering problem

- Diverse clustering algorithms have been applied to identify cell clusters in scRNA-seq data: k-means, hierarchical, density-based, and graph-based clustering.

- To ensure that each cluster is associated with a distinct cell type or state:

Clustering: feature selection

- Selecting genes showing differential expression across multiple cell types is essential for improving the quality of clustering results.

- Such relevant genes can be identified by selecting genes that are highly variable across cells.

- Both feature selection and dimensionality reduction can be sequentially applied to extract informative features that are taken as input to clustering algorithms.

Clustering: number of clusters

- Because the optimal number of clusters is dependent on the definition of cell types or states and subjective clustering resolution, it cannot be generally estimated from data.

- It is generally recommended that the number of clusters should be chosen by a user with domain-specific knowledge.

Clustering: rare cell types

- Identifying rare cell types, such as stem cells and short-lived progenitors, in a heterogeneous population requires
  - careful examination of outliers within a large cluster
  - selection of genes that are specifically expressed in a minor population of cells as features

Clustering: Batch correction

- If samples are processed in multiple batches and technical batch effects largely account for the observed variability, batch effects should be adjusted while preserving global structure.

- If the biological condition is not confounded by batch information, regression-based batch correction methods originally designed for bulk RNA-seq can be applied.

Clustering: Batch correction

- However, in a confounded design, which is common in the droplet-based protocols, the batch correction methods regress out both biological and technical variability.

- One solution is to project the expression profile of each cell to a feature space by calculating the correlation coefficient between the expression vector of single cells and the expression vector of the reference bulk panel of diverse cell types.

- A more general strategy is to merge multiple scRNA-seq data with shared subpopulations using canonical correlation analysis or by identifying mutual nearest neighbors.

Clustering: Batch correction

Nat Rev Genet (2019)

Clustering: Cell type annotation

- The identified clusters are annotated as cell types or states using the expression of known marker genes.

- To automate this annotation, researchers have developed correlation-based scoring methods or machine learning classifiers with the aid of reference bulk transcriptomes or reference single-cell transcriptomes.

- The identity of cell clusters can also be inferred by examining differentially expressed genes across cell clusters and their enriched functional categories of genes. Although statistical methods designed for differential expression analysis in scRNA-seq have been developed, their performance is comparable or sometimes inferior to methods designed for bulk RNA-seq or general purpose two-sample tests.

Continuous latent variable approach

- Pioneered by Monocle.
- Trajectory inference or pseudo-temporal ordering.
- The main assumption underlying this approach is that there exists a dynamic cellular process that shapes the transcriptional landscape and each individual cell can be placed along the process.
- Many dynamic cellular processes, including differentiation, reprogramming, and cell cycling, continuously progress along single or multiple trajectories, passing through transient cell states.

Continuous latent variable approach

- The temporal progression of each cell along these trajectories, termed pseudotime, is the continuous latent variable that is inferred from data.

- If a large number of cells covering transient states are sampled from a mixed population of cells whose cell-to-cell variability is largely driven by a given cellular process, trajectories can be accurately reconstructed.

Trajectory inference

Nat Biotechnol 34:1145 (2016)

Trajectory inference

- Most of these tools operate based on the assumption that cells showing similar expression profiles should be placed close together on the same trajectories.

- They use a recurring framework that consists of two steps: 1) constructing a low-dimensional representation of cells, and 2) modeling trajectories with graphs or curves in the low-dimensional representation.

Trajectory inference

Cell Stem Cell 23:166 (2018)

Constructing a low-dimensional representation of cells

- Two different classes of representation are used:
  - A two- or three-dimensional feature space generated using dimensionality reduction algorithms
  - A k-nearest neighbor graph in which each cell is repr esented as a node and each node is linked its k-near est neighbors

Constructing a low-dimensional representation of cells

- The low-dimensional feature space can be constructed by applying diverse dimensionality reduction algorithms after selecting genes relevant to the cellular process of interest.

- In principle, algorithms that preserve the global structure in the low-dimensional feature space, such as diffusion map and UMAP, should be used.

Constructing a low-dimensional representation of cells

- The k-NNG is usually constructed after projecting cells to the low-dimensional feature space using dimensionality reduction methods.

- For better visualization, k-NNGs can be arranged in a two-dimensional space using the force-directed layout embedding

Constructing a low-dimensional representation of cells

- For feature selection, there is no consensus on the best practice for selecting genes that are informative with respect to constructing the low-dimensional representation.

- Widely used criteria for this process include highly expressed genes, highly variable genes across cells, differentially expressed genes across cell clusters, genes that show gradual changes within a local neighborhood, and a set of known genes related to the cellular process.

Modeling trajectories

- A backbone of trajectories is constructed with graphs or curves in the low-dimensional representation, and then the pseudotime of cells is evaluated by projecting cells onto the backbone.

- Constructing the backbone, which usually requires prior information, such as the structure of trajectories and a root cell with a pseudotime of 0, is the key step for determining the accuracy of inferred trajectories.

Modeling trajectories

- Early methods fixed the structure of trajectories as linear or bifurcating.

- A more complex structure of trajectories is difficult to correctly reconstruct from data, since it becomes more sensitive to outlier cells, requires more prior information, and needs sampling of a sufficient number of cells.

Modeling trajectories

- The most widely used strategy for addressing this issue is to group cells into clusters that represent distinct cell types or states. The backbone is constructed by linking clusters, and the trajectories are inferred by specifying the start clusters, both start and end clusters, or all clusters on a given trajectory.

- Several methods for identifying the least differentiated cells (or stem cells) have been proposed for facilitating construction of the backbone.

Modeling trajectories

- In addition, the direction and the speed of differentiation can be inferred from RNA velocity, but this is sensitive to the set of input genes.

- After reconstructing trajectories, the dynamics of gene regulation along the inferred trajectories can be analyzed (e.g Scenic).

Example: Monocle2

$$\min_{\mathcal{G}\in\mathcal{G}_b}\ \min_{f_{\mathcal{G}}\in\mathcal{F}}\ \min_{Z}\ \sum_{i=1}^{N} \| \boldsymbol{x}_i - f_{\mathcal{G}}(\boldsymbol{z}_i) \|^2 + \frac{\lambda}{2} \sum_{(V_i,V_j)\in\varepsilon} b_{i,j} \| f_{\mathcal{G}}(\boldsymbol{z}_i) - f_{\mathcal{G}}(\boldsymbol{z}_j) \|^2$$

Nat Methods 14:979 (2017)

Integrative single-cell analysis

Nat Rev Genet (2019)

From SingleCellExperiment to Seurat

- SingleCellExperiment -> Seurat

```
library(Seurat)
Seuratset <- CreateSeuratObject(sce@assays$data$counts)
Seuratset@var.genes <- rownames(hvg)
Seuratset@data <- sce@assays$data$logcounts
Seuratset <- ScaleData(object = Seuratset)
```

PCA by Seurat

- Seurat PCA

```
PCA = 50
Seuratset <- RunPCA(Seuratset, pcs.compute = PCA, weight.by.var = FALSE, do.print = FALSE)
plot(Seuratset@dr$pca@sdev,
     xlab = "PC",
     ylab = "Eigenvalue")
```

t-SNE by Seurat

- Seurat t-SNE

```
Seuratset <- RunTSNE(Seuratset, dims.use = 1:15,
                     do.fast = T, seed.use = 42, perplexity=30)
```

Clustering by Seurat

- Seurat Clustering

```
Seuratset <- FindClusters(Seuratset, reduction.type="pca",
                          dims.use = 1:15, save.SNN = TRUE,
                          force.recalc = TRUE)
```

Visualization by Seurat

- Seurat PCA/t-SNE/UMAP visualization

DEGs by Seurat

- Seurat Markers(differentially expressed genes)

```
markerGenes <- FindAllMarkers(Seuratset)
```

Result

```
> head(markerGenes)
                       p_val avg_logFC pct.1 pct.2     p_val_adj cluster            gene
ENSG00000105640 4.185909e-170 0.7876883 1.000 0.999 5.638419e-166       0 ENSG00000105640
ENSG00000231500 3.758906e-162 0.7943735 0.998 1.000 5.063247e-158       0 ENSG00000231500
ENSG00000145425 2.988751e-161 0.8001494 1.000 0.999 4.025848e-157       0 ENSG00000145425
ENSG00000137154 3.260534e-158 0.7637400 1.000 1.000 4.391939e-154       0 ENSG00000137154
ENSG00000144713 1.348842e-152 0.6985423 1.000 1.000 1.816890e-148       0 ENSG00000144713
ENSG00000147604 3.889303e-141 0.7500199 0.998 0.999 5.238891e-137       0 ENSG00000147604
```

DEGs by Seurat

- Heatmap

```
heatmap_features <- markerGenes_reordered %>%
  group_by(cluster) %>%
  top_n(n = 3, wt = avg_logFC)

DoHeatmap(Seuratset, genes.use = heatmap_features$gene)
```

Cell type annotation

- Seurat Violin plot

Cell type annotation

- Seurat RidgePlot

Cell type annotation

- Seurat DotPlot

Cell type annotation

Cell type annotation

Cell type annotation

Cell type annotation

Cell type annotation

Cluster 0: CCR7+ T_N cells
Cluster 1,2: KLRB1+ CCR6– SELL– T_M cells
Cluster 3: KLRB1+ CCR6+ SELL– T_M cells
Cluster 4: Undefined T_M cells
Cluster 5: FOXP3 + Treg cells
Cluster 6: Proliferating cells

Trajectory inference

- Excluding Treg cells and proliferating cells

Trajectory analysis with cluster (0,1,2,3)
- Excluding cluster 4,5,6

Destiny

```
ordering_genes <- rownames(trajectory_hvg)
destinySCEset <- trajectory_sceset[ordering_genes, ]
dm <- DiffusionMap(t(as.matrix(logcounts(destinySCEset))))

tmp <- data.frame(DC1 = eigenvectors(dm)[,1],
                  DC2 = eigenvectors(dm)[,2],
                  clusters = Seuratset@ident)

ggplot(tmp, aes(x = DC1, y = DC2, colour = clusters)) +
  geom_point(size=0.1) +
  xlab("Diffusion component 1") +
  ylab("Diffusion component 2") +
  theme_classic()
```

Destiny

Diffusion component 2
0.04
0.00
-0.04
-0.025
0.000
0.025
0.050
0.075
Diffusion component 1
clusters
0
1
2
3

Cells ordered by diffusion map pseudotime
clusters
3
2
1
0
0
500
1000
1500
Diffusion map pseudotime (first diffusion map component)
clusters
0
1
2
3

Monocle2

```
library(monocle)
set.seed(12345)

pd <- new("AnnotatedDataFrame", data = trajectory_seurat@meta.data)
fd <- new("AnnotatedDataFrame", data = gene_annotation)
cds <- newCellDataSet(trajectory_seurat@raw.data, phenoData = pd, featureData = fd,
                      expressionFamily = negbinomial.size())
cds <- estimateSizeFactors(cds)
cds <- estimateDispersions(cds)

cds <- detectGenes(cds, min_expr = 0.1)
print(head(fData(cds)))
expressed_genes <- row.names(subset(fData(cds), num_cells_expressed >= 5))

set.seed(123)
cds <- reduceDimension(cds, max_components = 2, num_dim = 50, norm_method = 'log',
                       reduction_method = 'tSNE', verbose = T)

ordering_genes <- rownames(trajectory_hvg)

cds <- setOrderingFilter(cds, ordering_genes)
cds <- reduceDimension(cds, method = 'DDRTree')
cds <- orderCells(cds)

plot_cell_trajectory(cds, color_by = "clusters")
```

Monocle2

clusters 0 1 2 3
KLRB1+ cells
CCR7+ T_N cells
KLRB1+ cells
1
Component 2
Component 1
2
1
0
-1
-2
-2.5
0.0
2.5
5.0

CCR7
Component 2
Component 1
2
1
0
-1
-2
-2.5
0.0
2.5
5.0
0

KLRB1
Component 2
Component 1
2
1
0
-1
-2
-2.5
0.0
2.5
5.0
5.02
0

Monocle2

clusters 0 1 2 3
KLRB1+ CCR6– SELL– T_M cells
KLRB1+ CCR6+ SELL– T_M cells
Component 2
Component 1
CCR6
KLRB1
5.02
0

Monocle2

- Differential expression analysis by pseudotime