Single-cell RNA sequencing

Introduction

Jong Kyoung Kim

DGIST

강의교재

Molecules and Cells

Minireview

Dissecting Cellular Heterogeneity Using Single-Cell RNA Sequencing

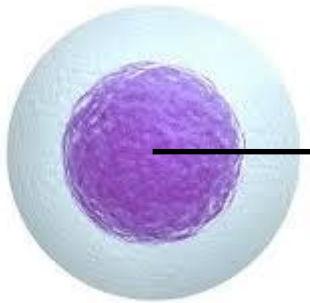
Yoon Ha Choi and Jong Kyoung Kim*

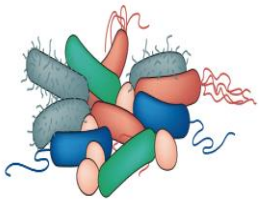
Department of New Biology, DGIST, Daegu 42988, Korea
*Correspondence: jkkim@dgist.ac.kr
http://dx.doi.org/10.14348/molcells.2018.0446
www.molcells.org

실습 자료

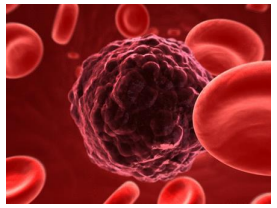
https://github.com/scg-dgist/2021_KOGO_workshop

Single cell: Limited materials

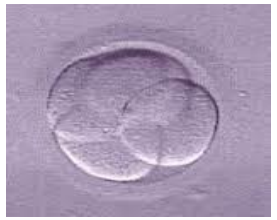




Uncultivated microbial species

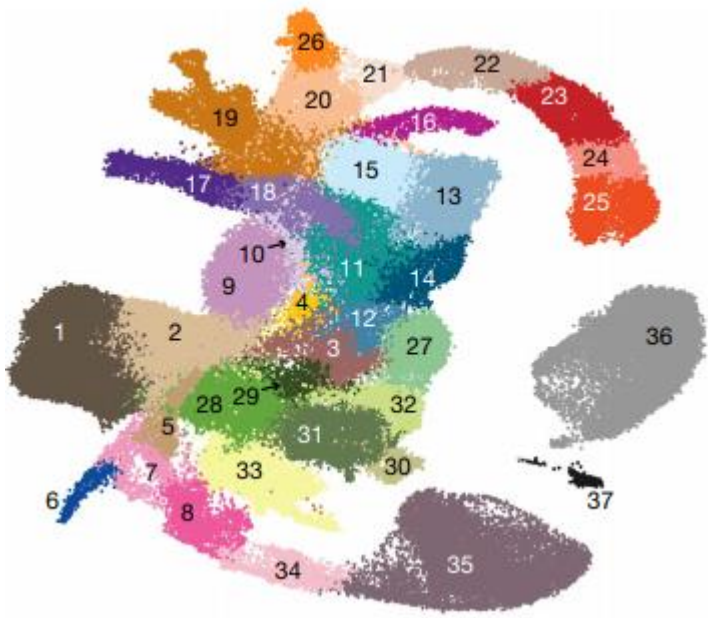


Rare cell types

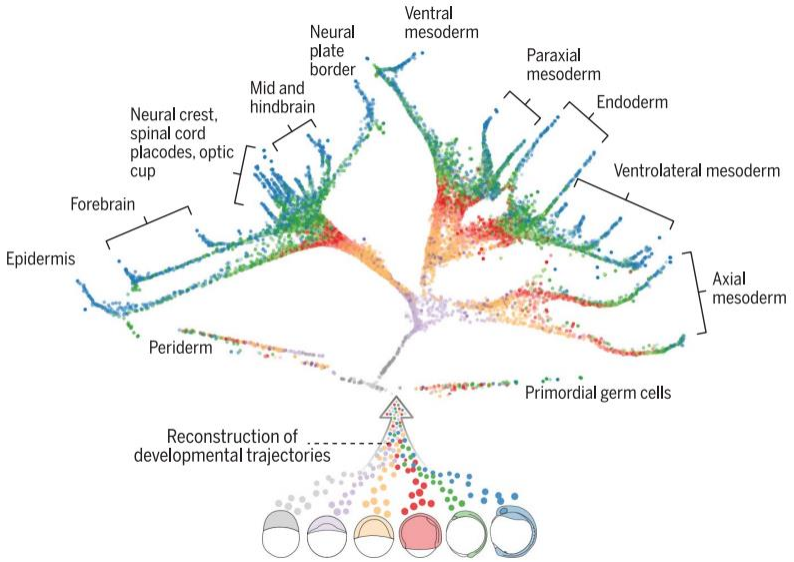


Early development

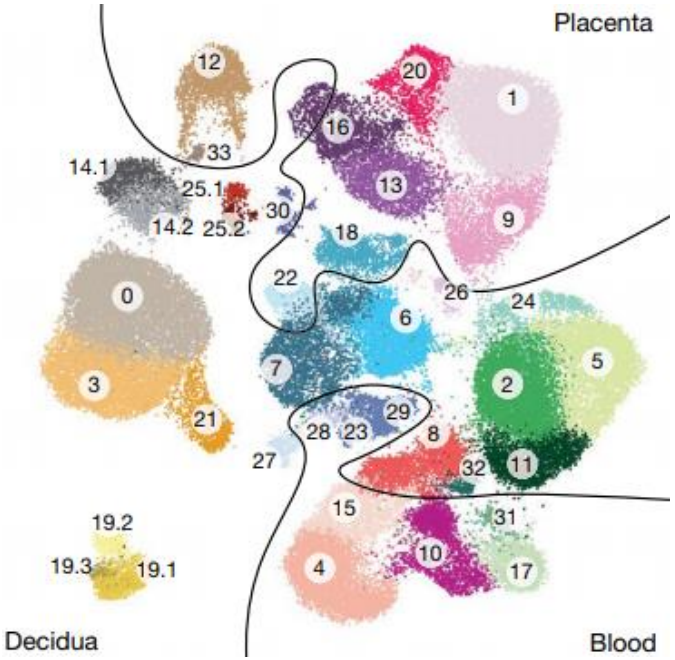
Embryogenesis at single-cell resolution



116,312 cells of mouse early organogenesis, Nature (2019) 566:490

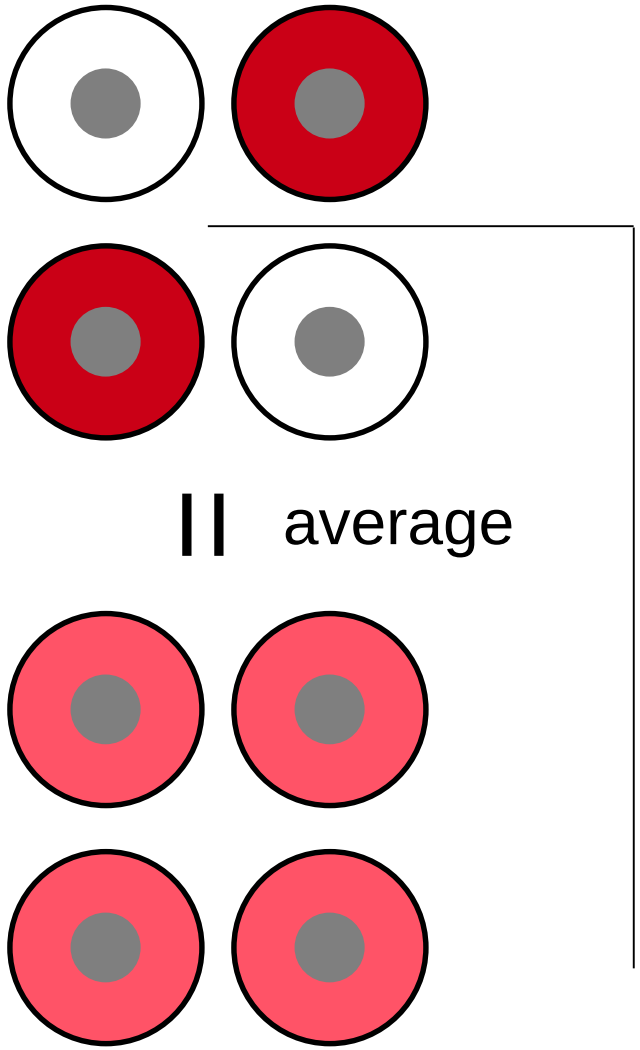


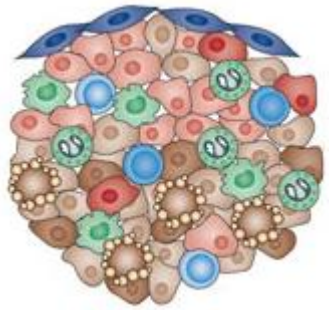
38,731 cells of zebrafish embryogenesis, Science (2018) 360:eaar3131



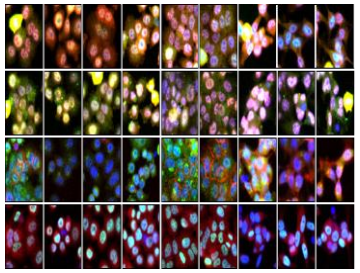
70,000 cells of human placentas, Nature (2018) 563:347

Single cell: Heterogeneity





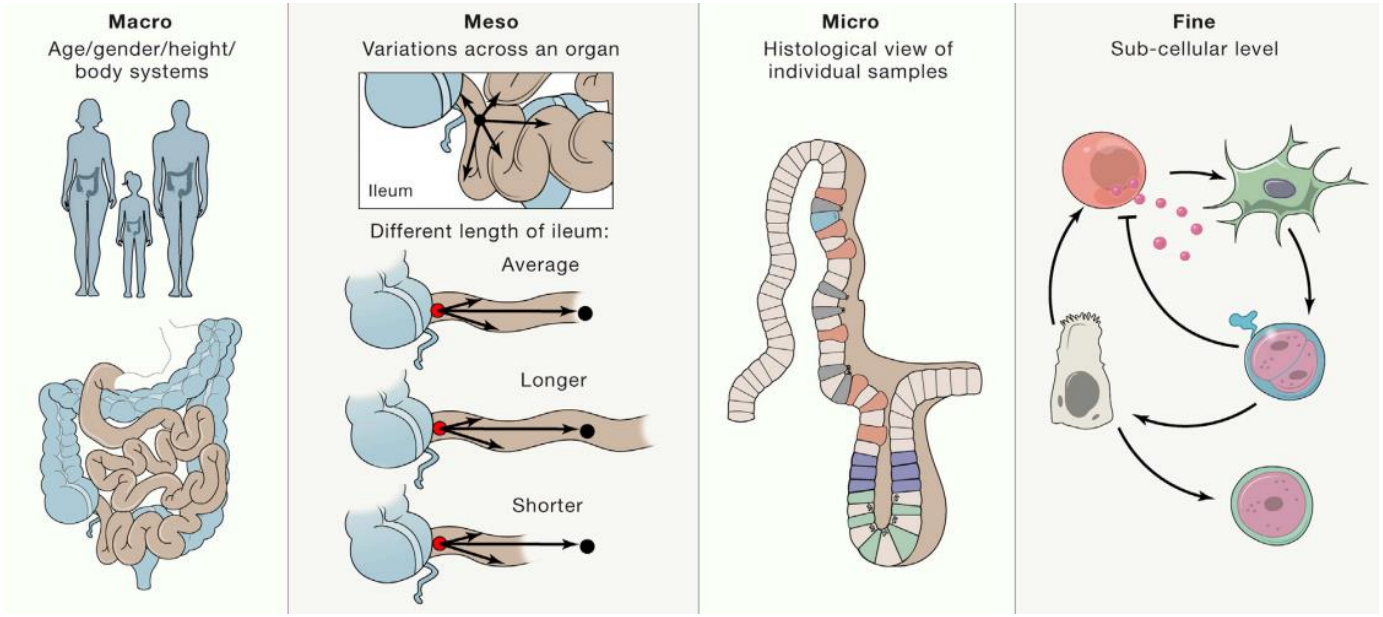
Tumour heterogeneity



Tissue heterogeneity

The Cell Atlas

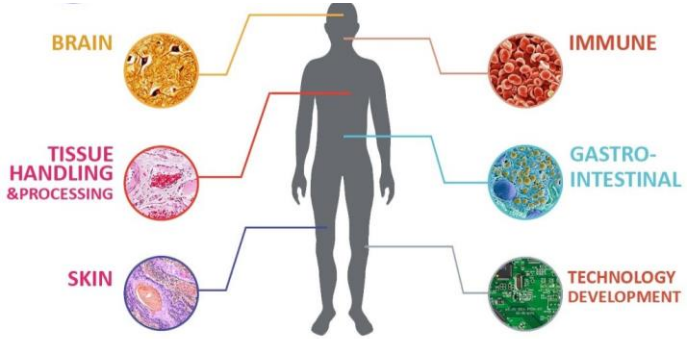
An atlas of cells, tissues, and organs throughout the whole body of a model organism that captures their molecular characteristics and their spatial location and organization during development and in health and disease.



Cell (2019)179:1455

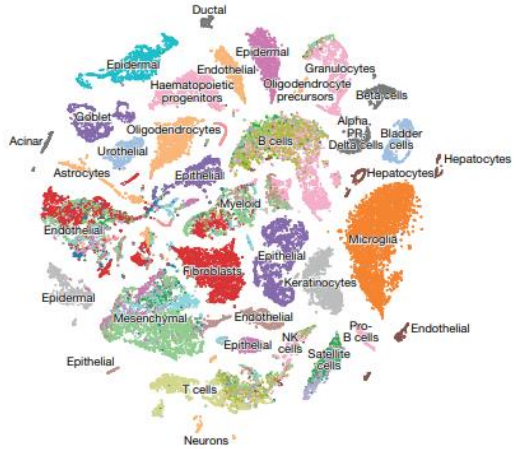
The Cell Atlas: examples

The Human Cell Atlas



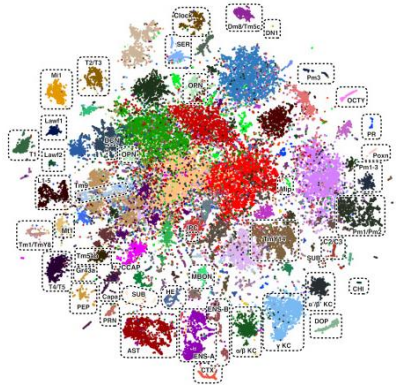
eLife (2017) 6:e27041

The Mouse Cell Atlas



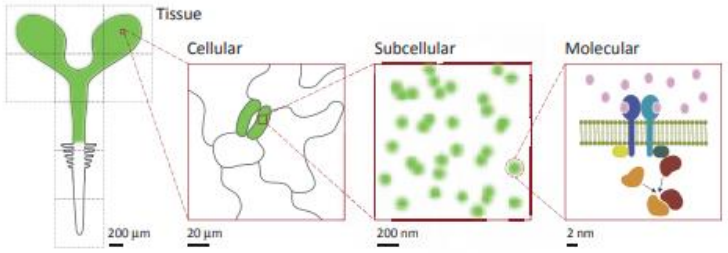
Cell (2018) 172:1091, Nature (2018) 562:367

The Fly Cell Atlas



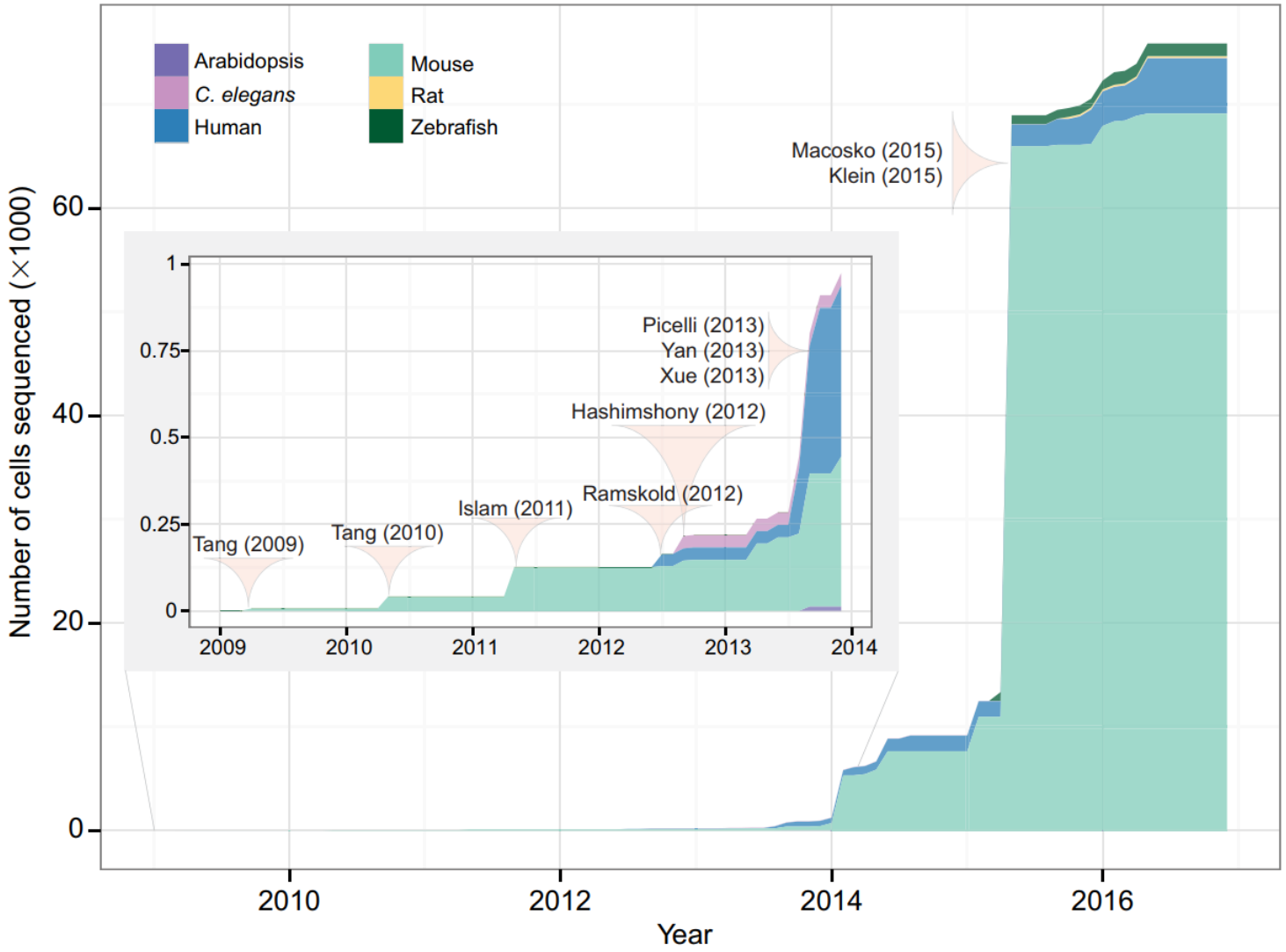
Cell (2018) 174:982

The Plant Cell Atlas



Trends in Plant Science (2019) 24:303

The growth of scRNA-seq techniques



Development (2017) 144:17

The first paper: Tang protocol

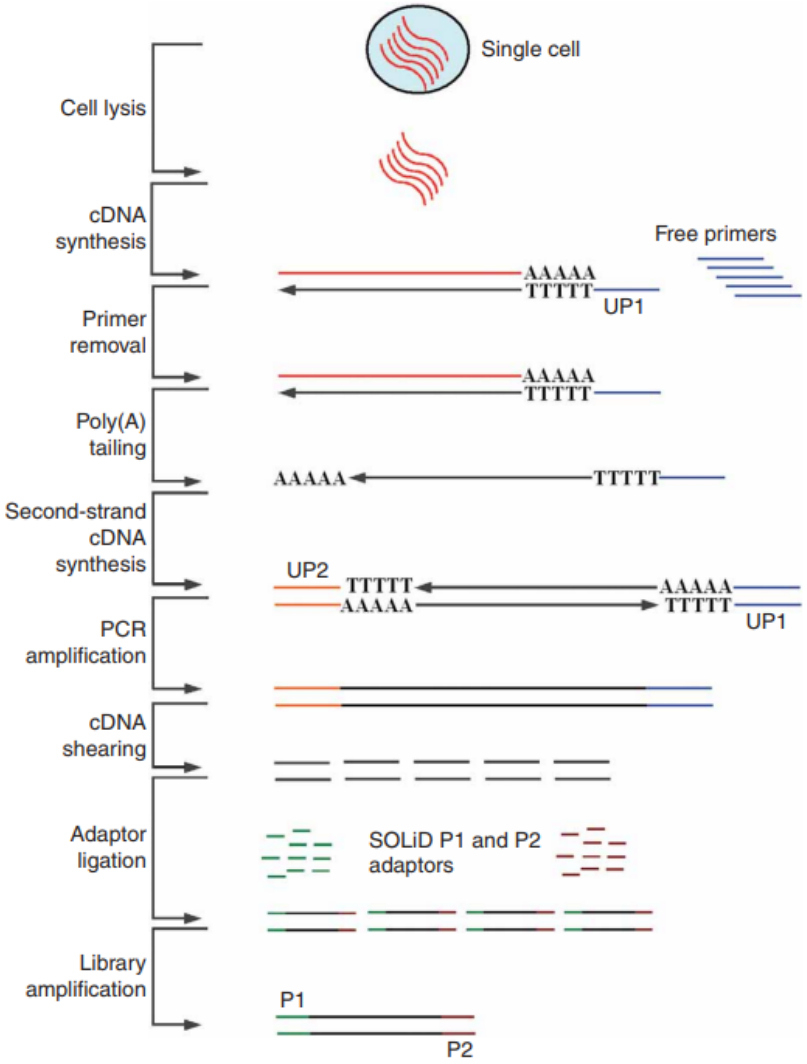
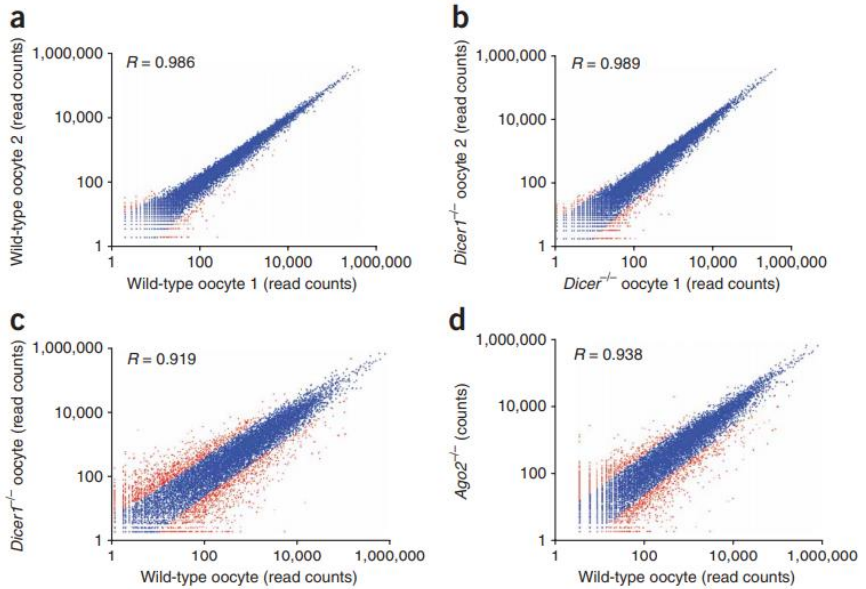


Table 1 | Single cell mRNA-Seq mapping summary

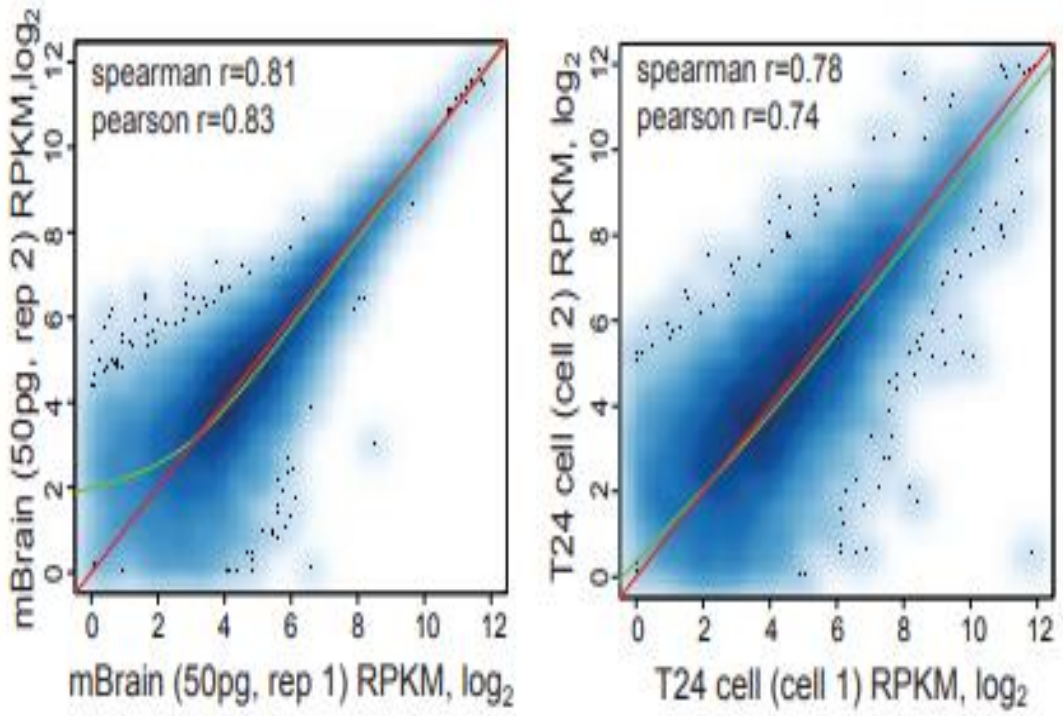
	Blastomere (50-base reads)	Blastomere (35-base reads)	Wild-type oocyte 1 (50-base reads)	Wild-type oocyte 2 (50-base reads)	*Dicer1*$^{-/-}$ oocyte 1 (50-base reads)	*Dicer1*$^{-/-}$ oocyte 2 (50-base reads)	*Ago2*$^{-/-}$ oocyte (50-base reads)
Reads processed	85,807,979	24,424,339	76,584,432	20,998,366	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

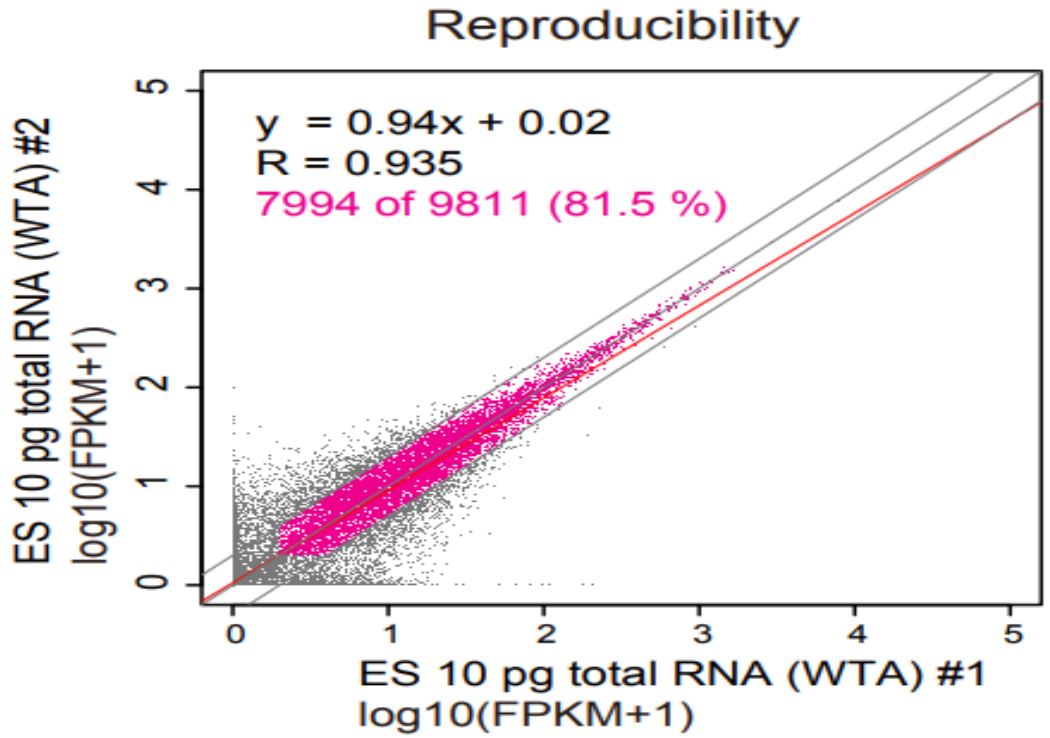


Nat Methods (2009) 6:377

Early protocols

Smart-Seq	Quartz-Seq



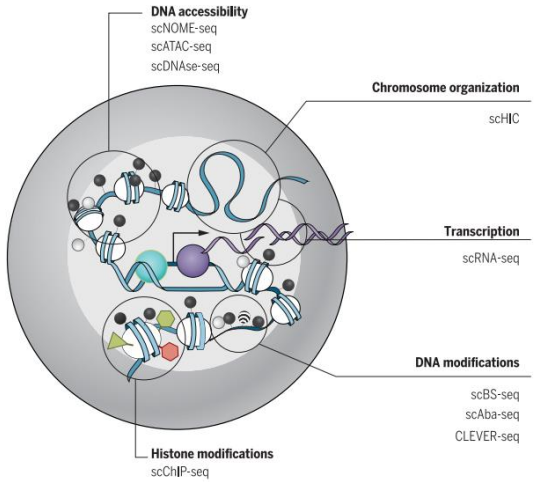


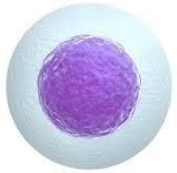
Nat Biotechnol (2012) 30:777

Genome Biol (2013) 14:R31

Experimental challenges in scRNA-seq

Multi-omics





Capturing single cells

Cell lysis and mRNA enrichment

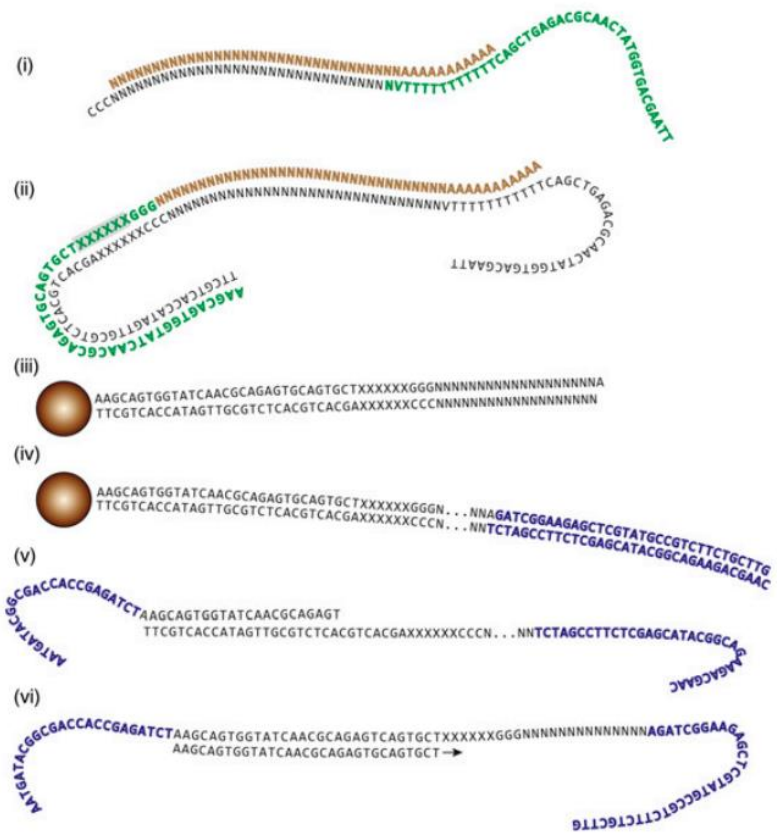
Reverse transcription

Amplification

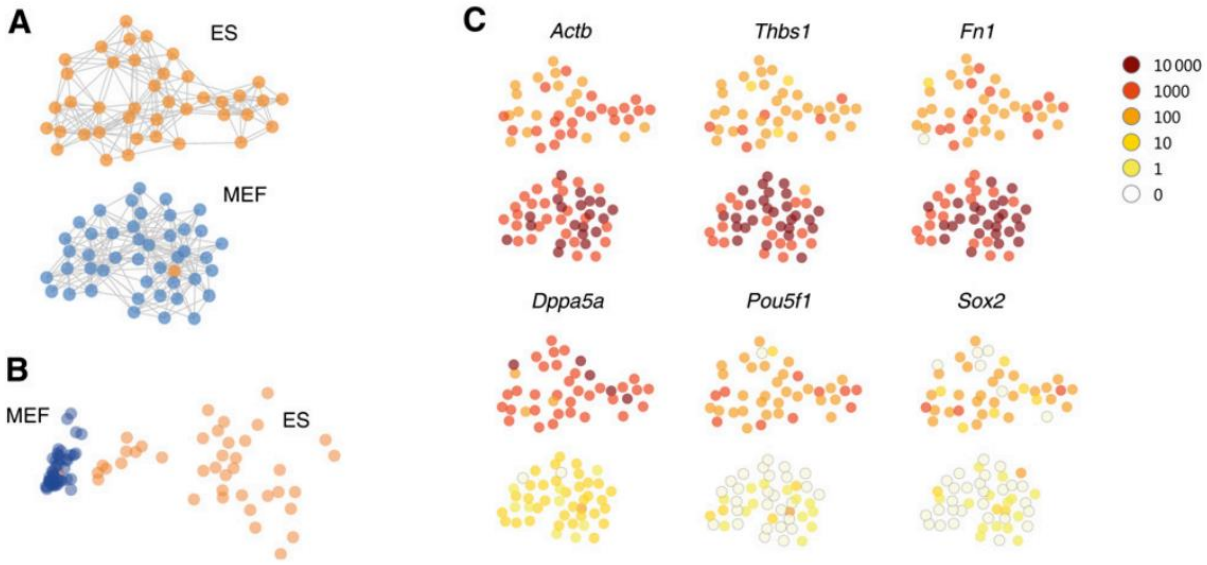
NGS

Multiplexing by cellular barcoding

STRT-seq



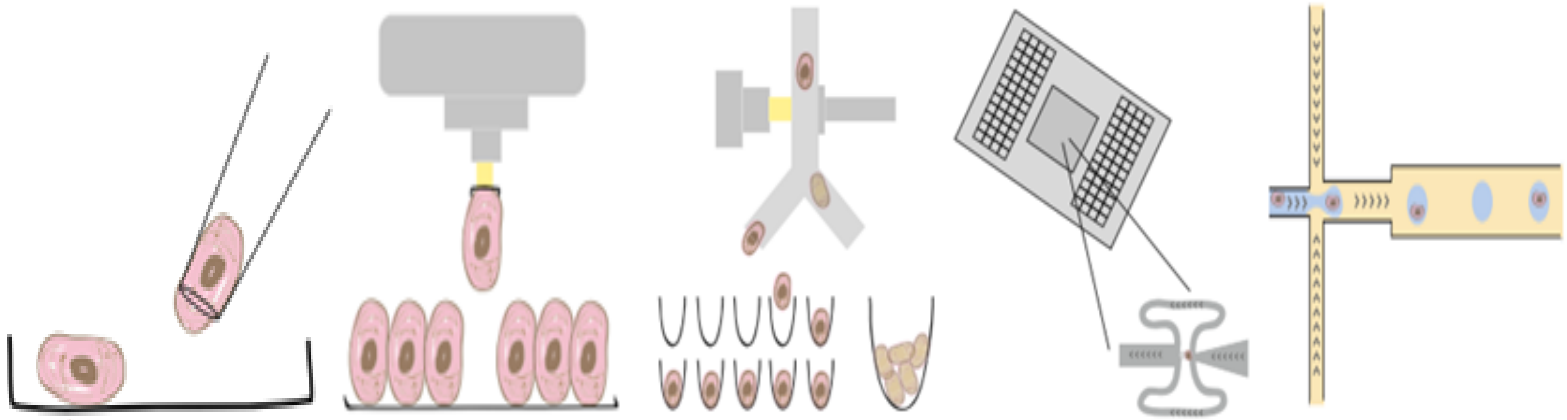
n=85



"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."

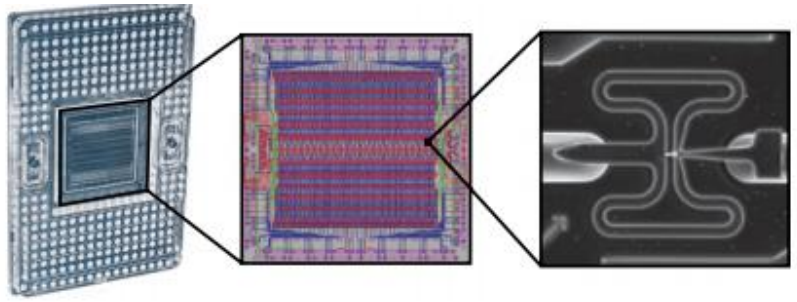
Genome Res (2011) 21:1160

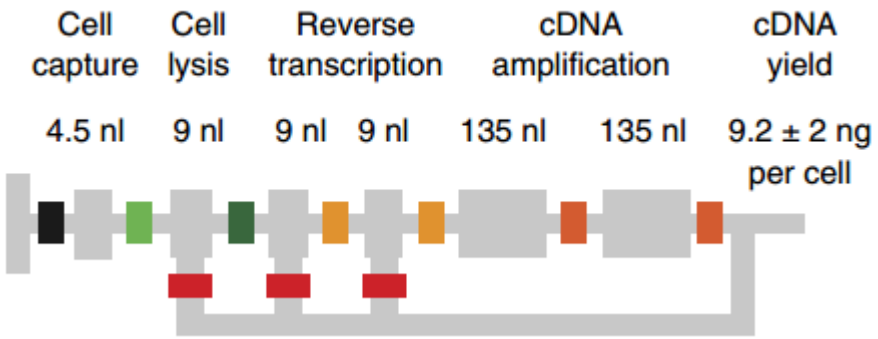
Capturing single cells

	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**

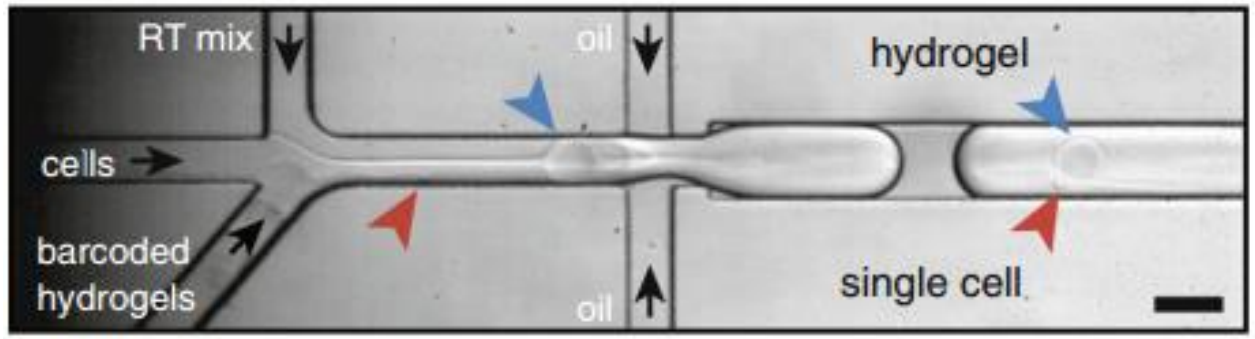
Microfluidics: Fluidigm C1

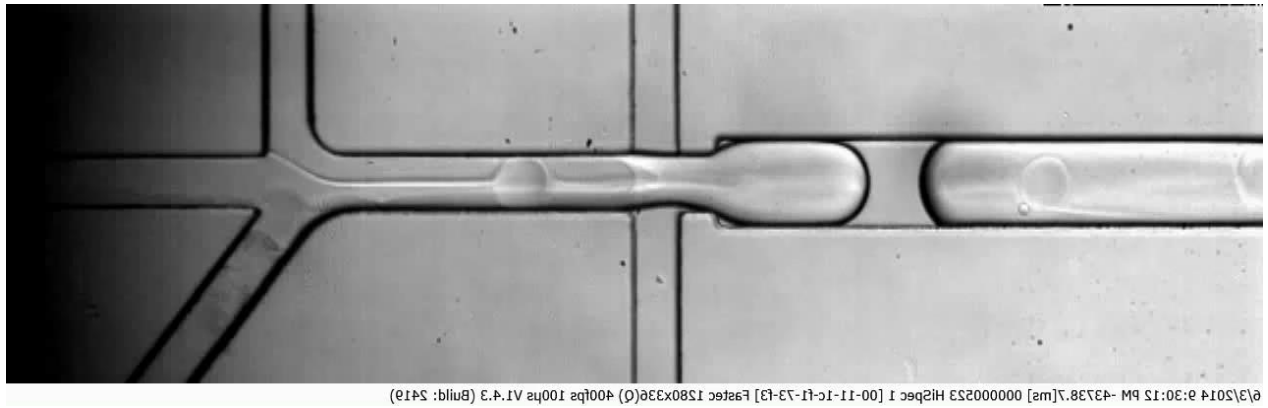




Nature Biotechnology (2014) 32:1053

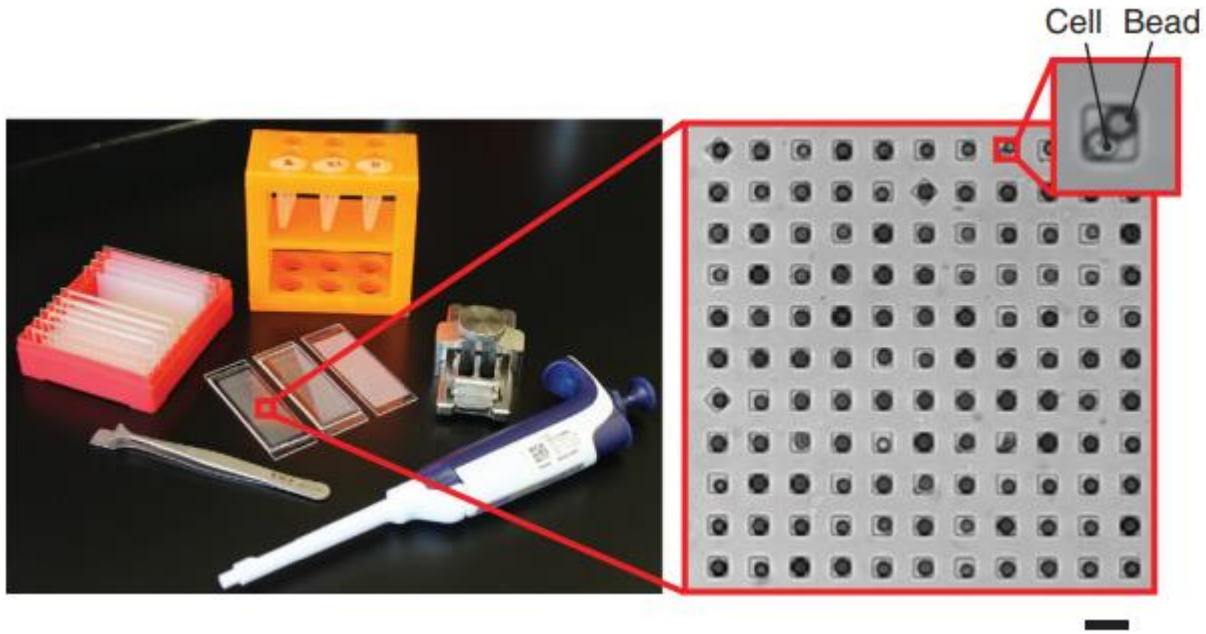
Microfluidics: Microdroplet

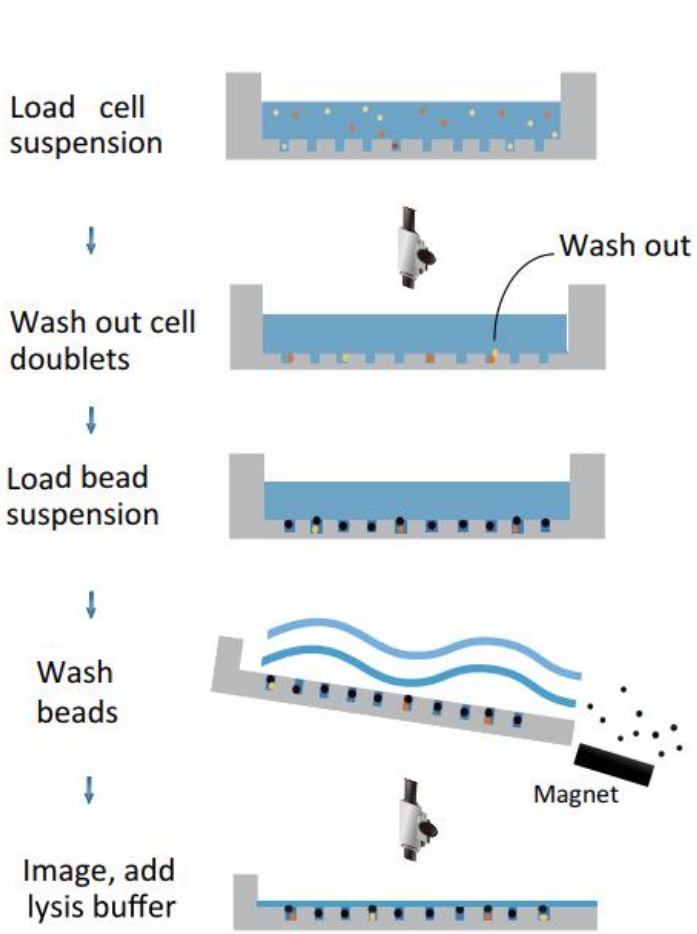


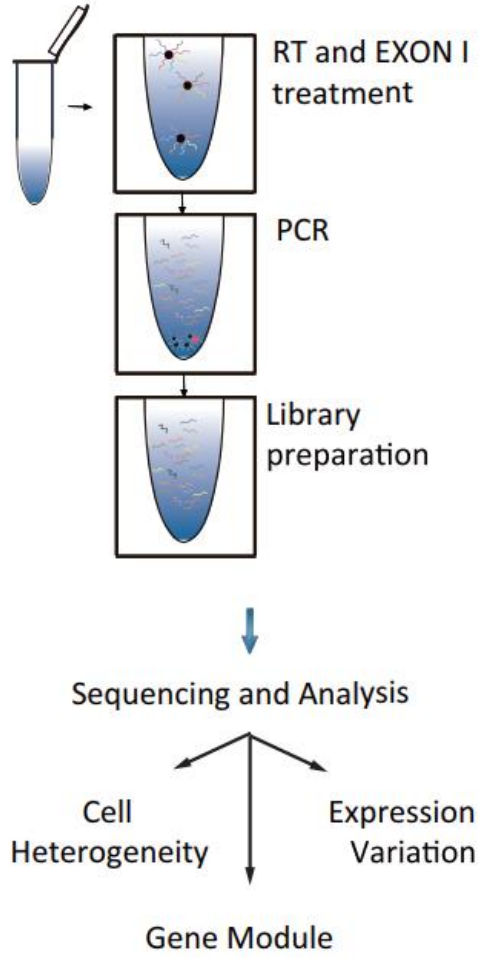


Cell (2015) 161:1187

Capturing single cells: Microwells



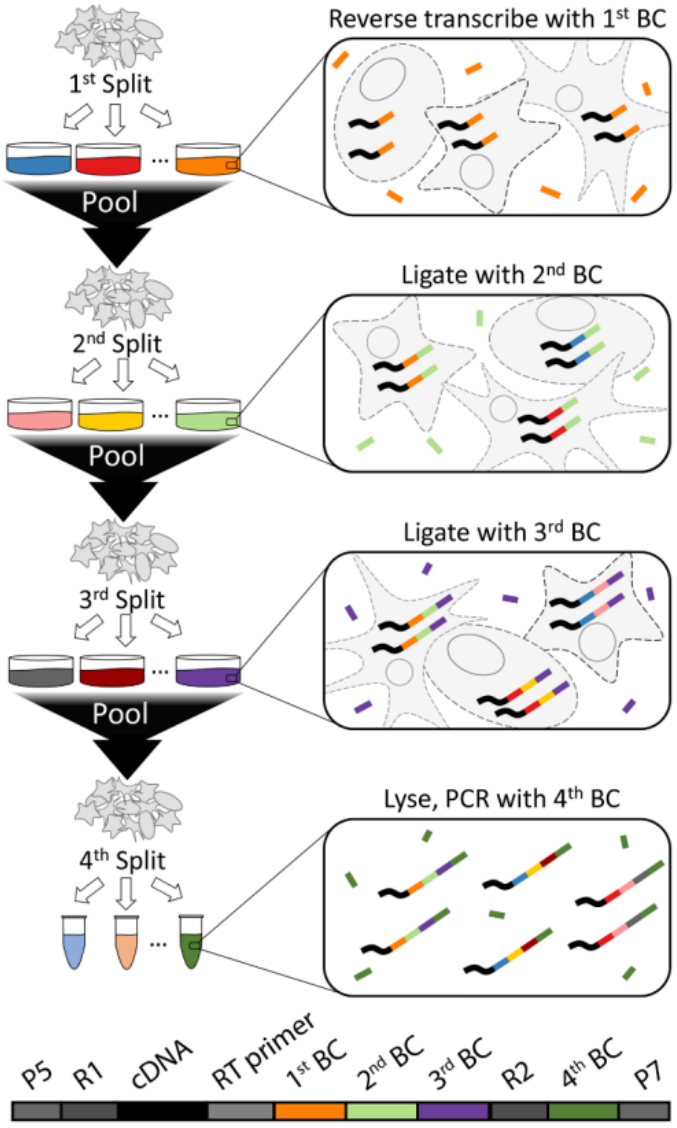




Nat Methods (2017) 14:395

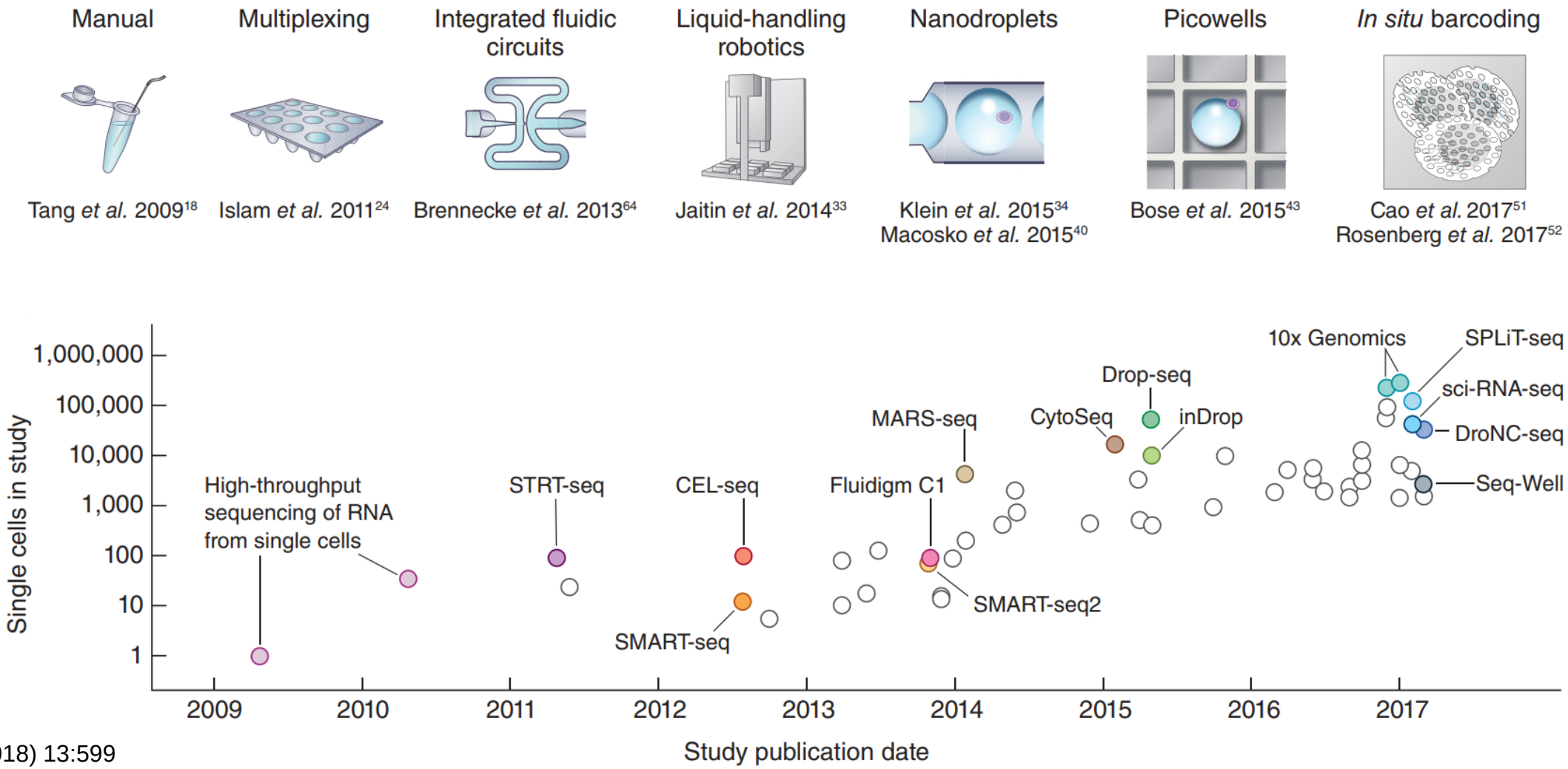
Cell (2018) 172:1091

Capturing single cells: SPLiT-seq



Science (2018) 360:176

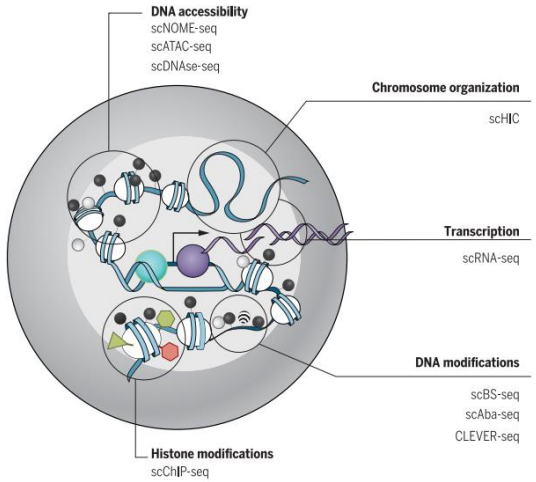
Scaling scRNA-seq experiments

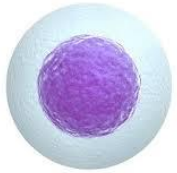


Nat Protoc (2018) 13:599

Experimental challenges in scRNA-seq

Multi-omics





Capturing single cells

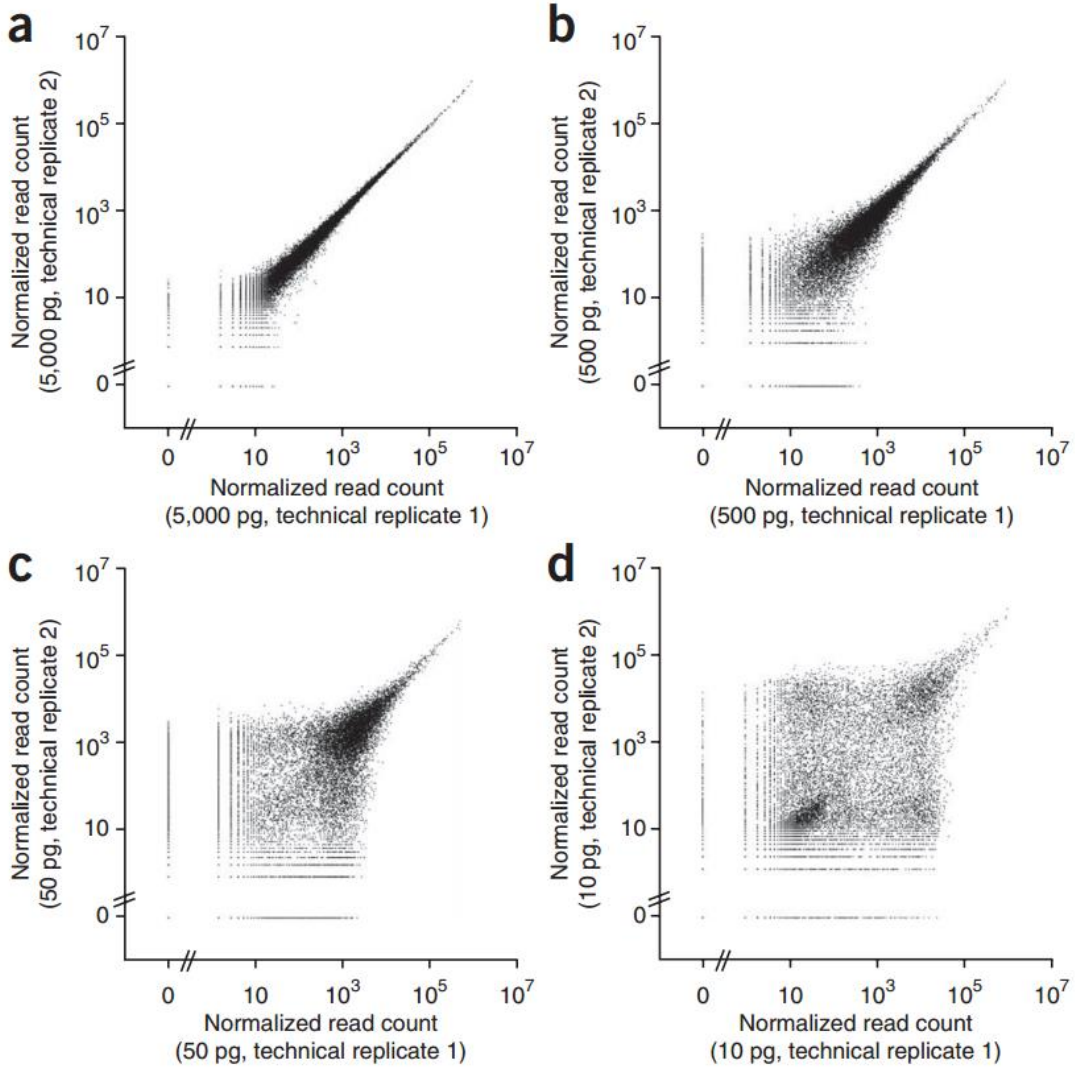
Cell lysis and mRNA enrichment

Reverse transcription

Amplification

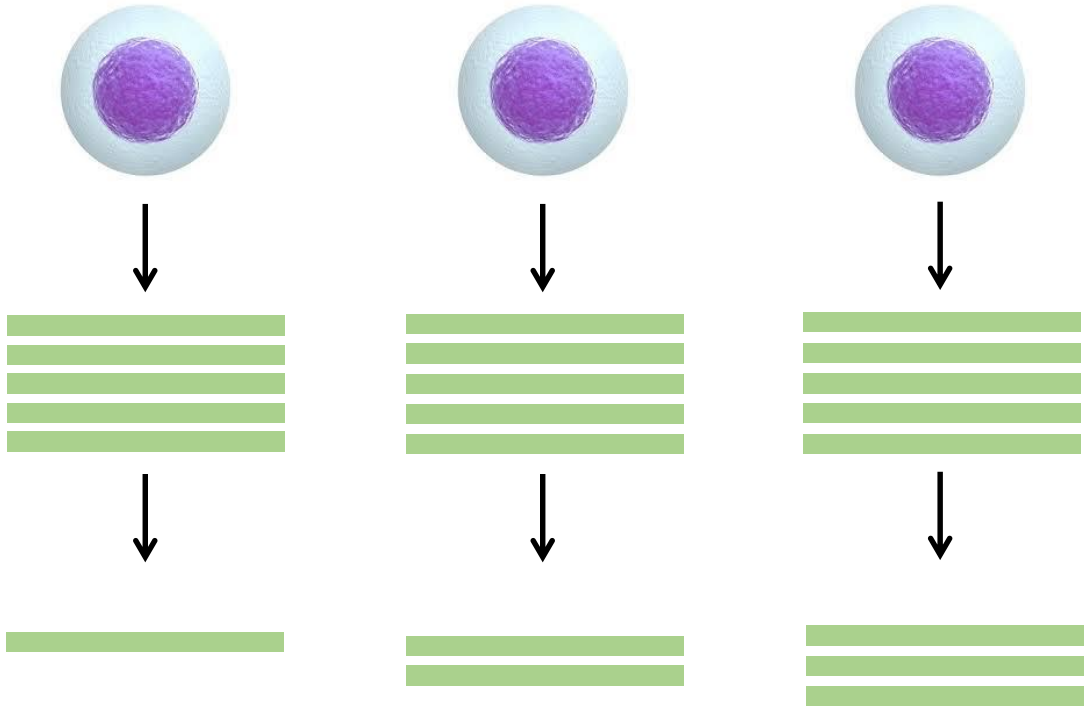
NGS

Technical noise in scRNA-seq



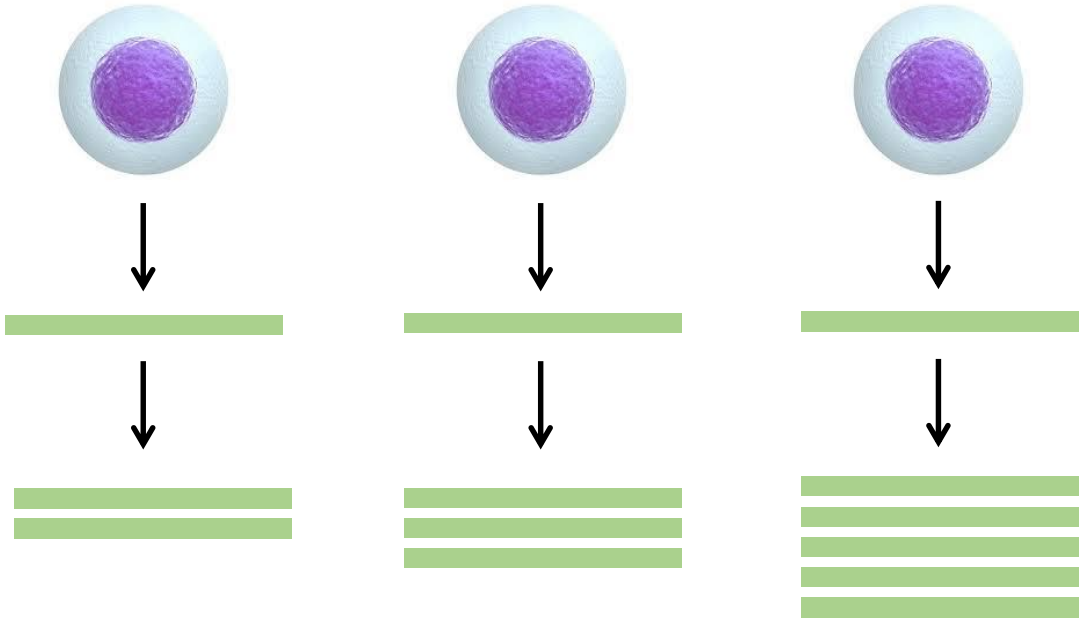
Brennecke, Anders*, Kim* et al (2013)* ***Nature Methods***

Technical noise: stochastic mRNA loss



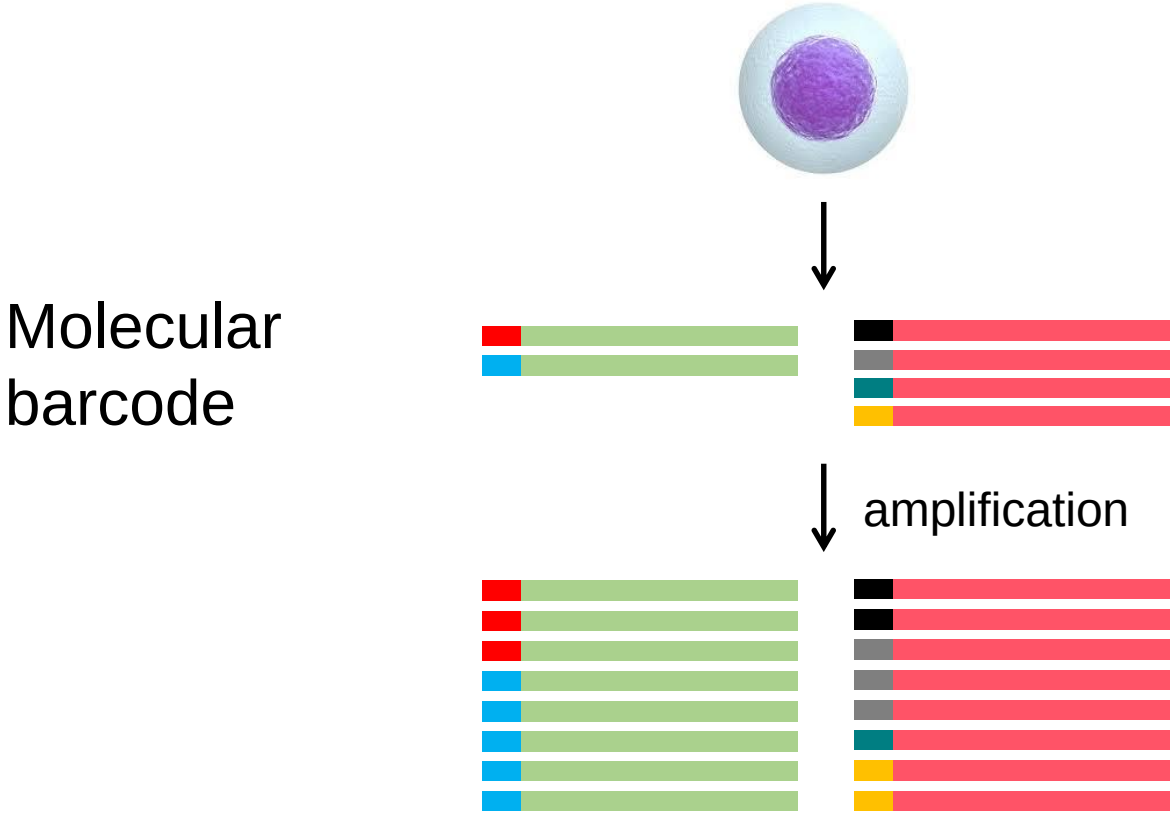
Capture efficiency	20%	40%	60%

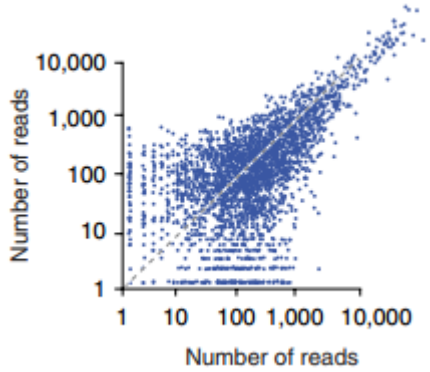
Technical noise: amplification

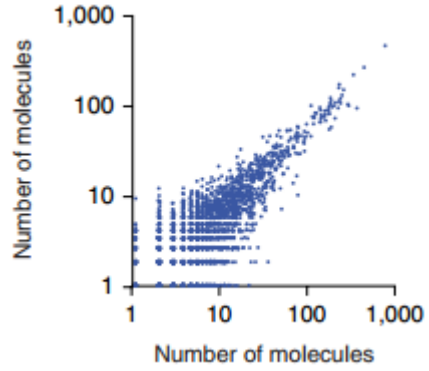


Amplification factor	2	3	5

Minimizing amplification bias by molecular barcodes





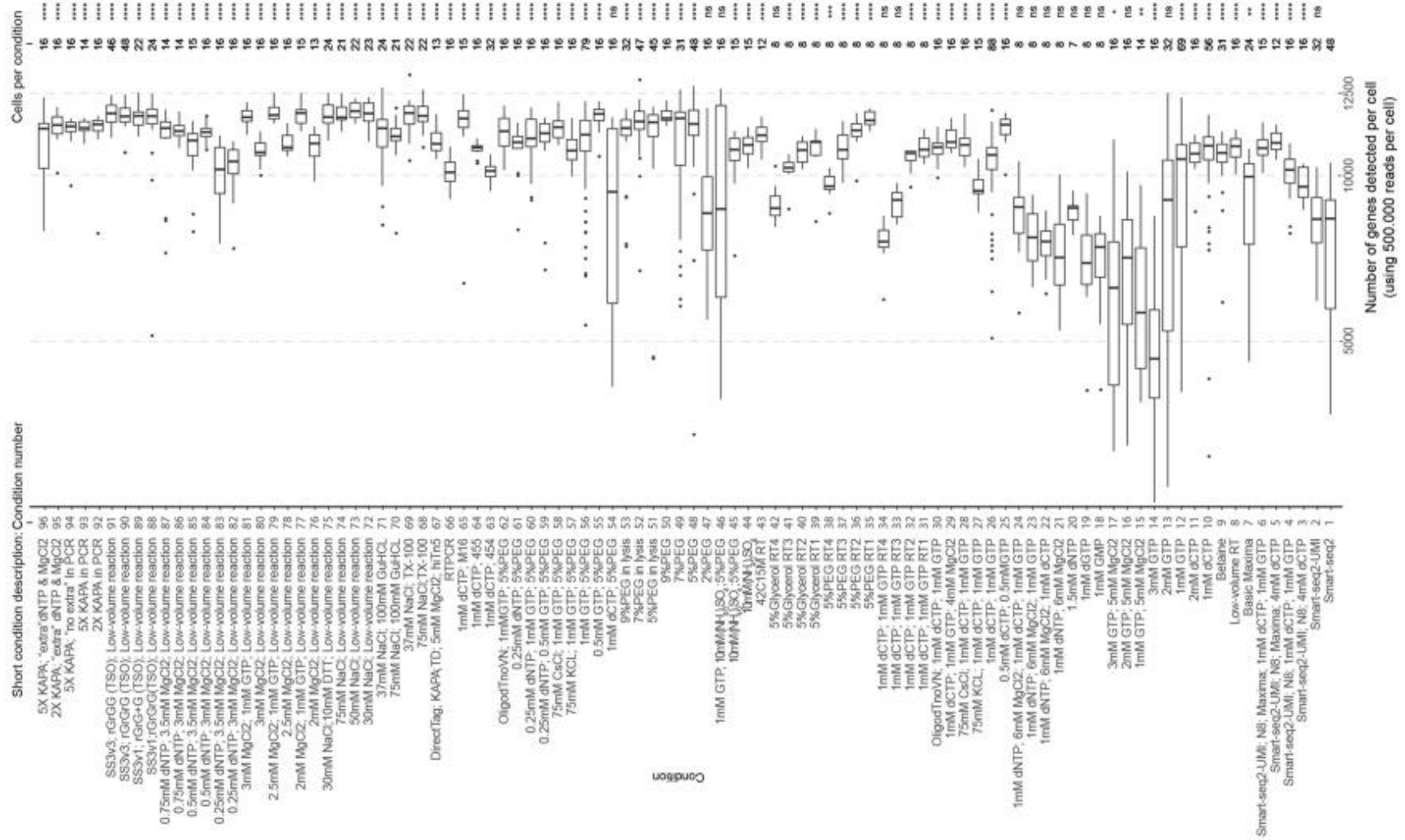


Nat Methods (2014) 11:163

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 (2016) 17:77) - Smart-seq3 (Nat Biotechnol (2020) 38:708)	- Quartz-Seq2 (Genome Biol (2018) 19:29) - Smart-seq3 (Nat Biotechnol (2020) 38:708)	- CEL-Seq2 (Genome Biol (2016) 17:77) - Smart-seq3 (Nat Biotechnol (2020) 38:708)	- mcSCRB-seq (Nat Commun (2018) 9:2937) - Smart-seq3 (Nat Biotechnol (2020) 38:708)	- CEL-Seq2 (Genome Biol (2016) 17:77) - mcSCRB-seq (Nat Commun (2018) 9:2937) - Smart-seq3 (Nat Biotechnol (2020) 38:708)

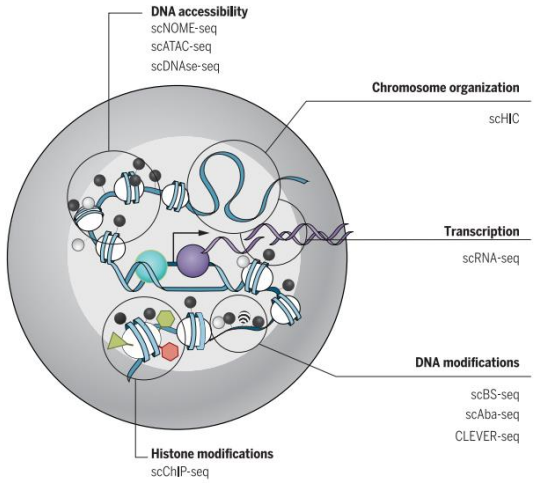
Further improvements for sensitivity

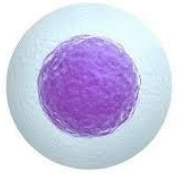


Nat Biotechnol (2020) 38:708

Experimental challenges in scRNA-seq

Multi-omics





Capturing single cells

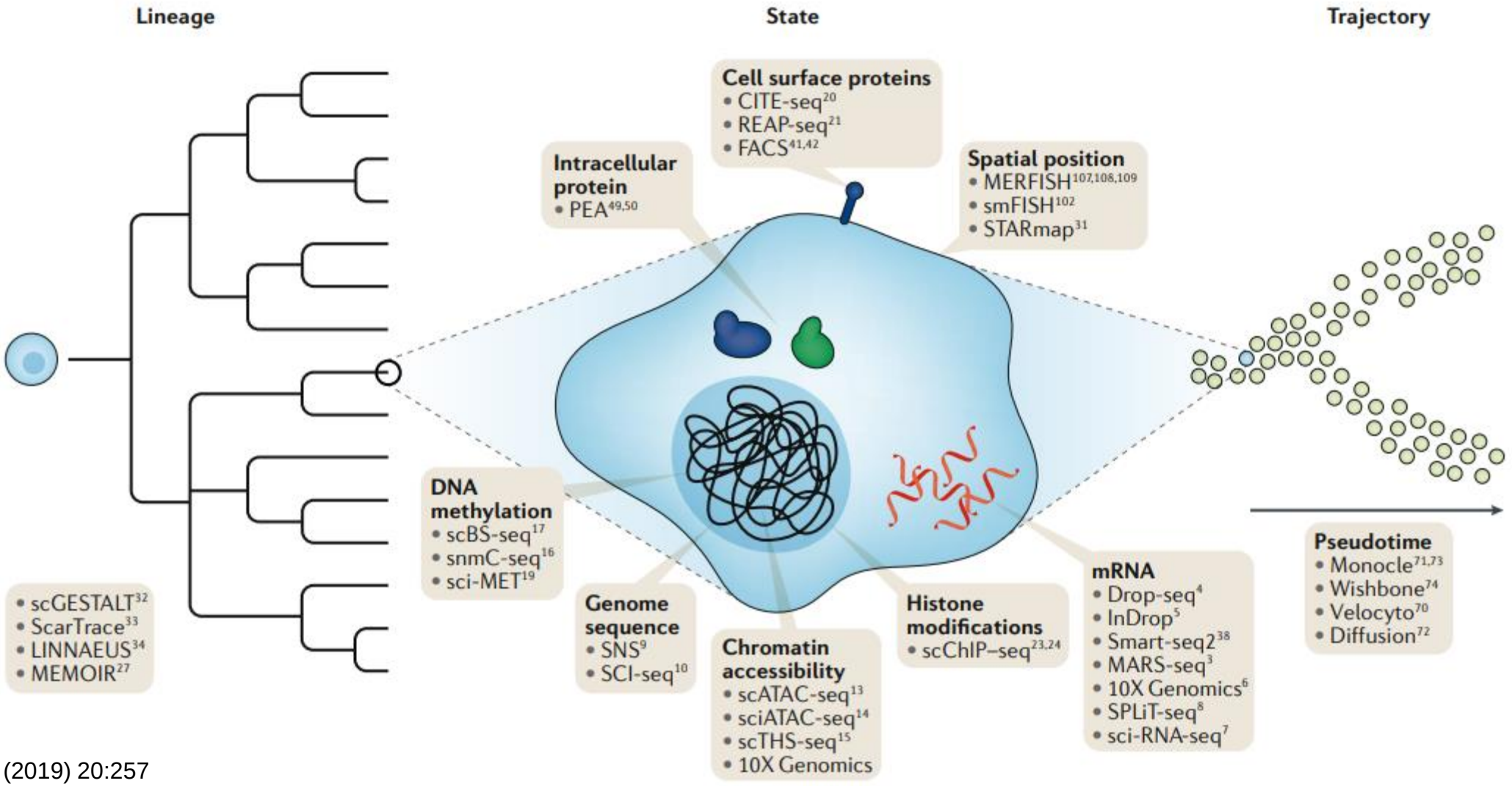
Cell lysis and mRNA enrichment

Reverse transcription

Amplification

NGS

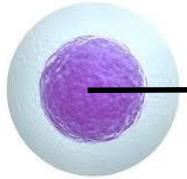
Integrative single-cell analysis

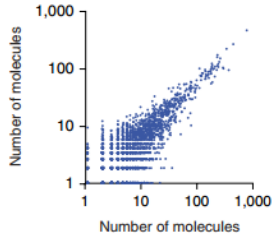


Nat Rev Genet (2019) 20:257

Computational challenges

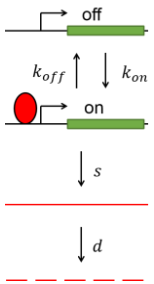
scRNA-seq





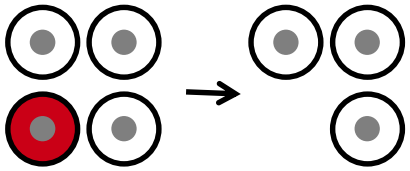
High technical noise

statistics



High biological noise

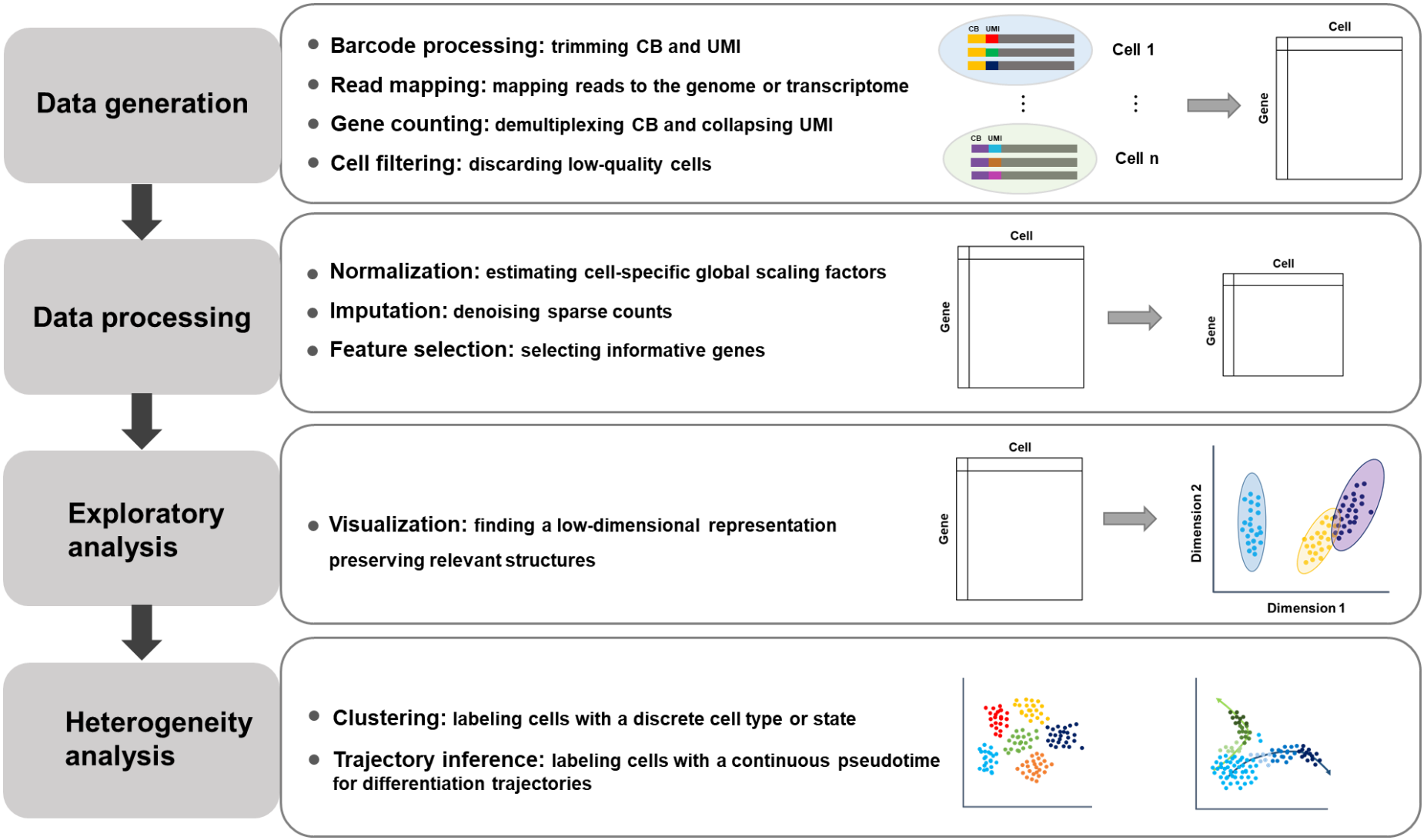
mathematical modelling



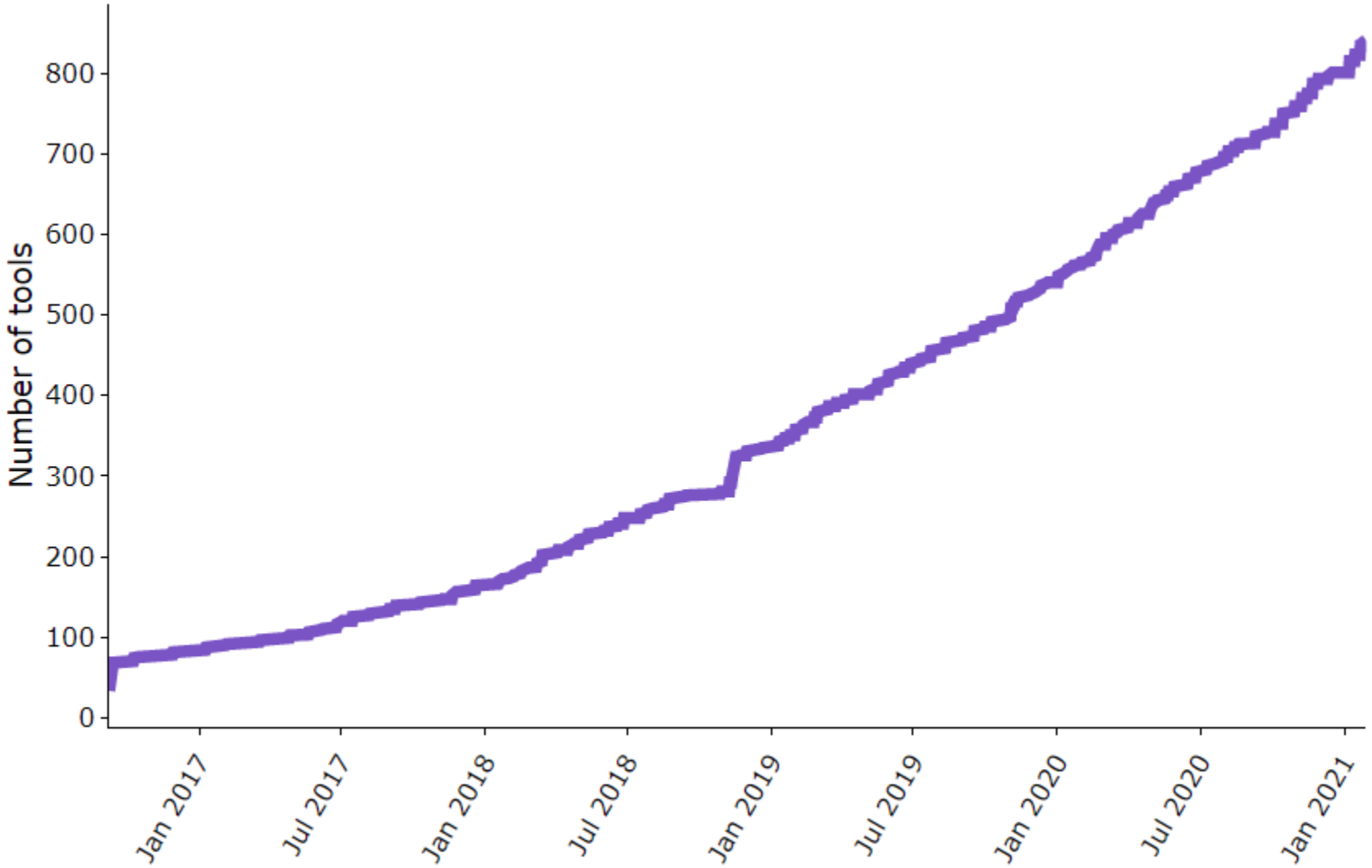
Big data

machine learning

Workflow for analyzing scRNA-seq data



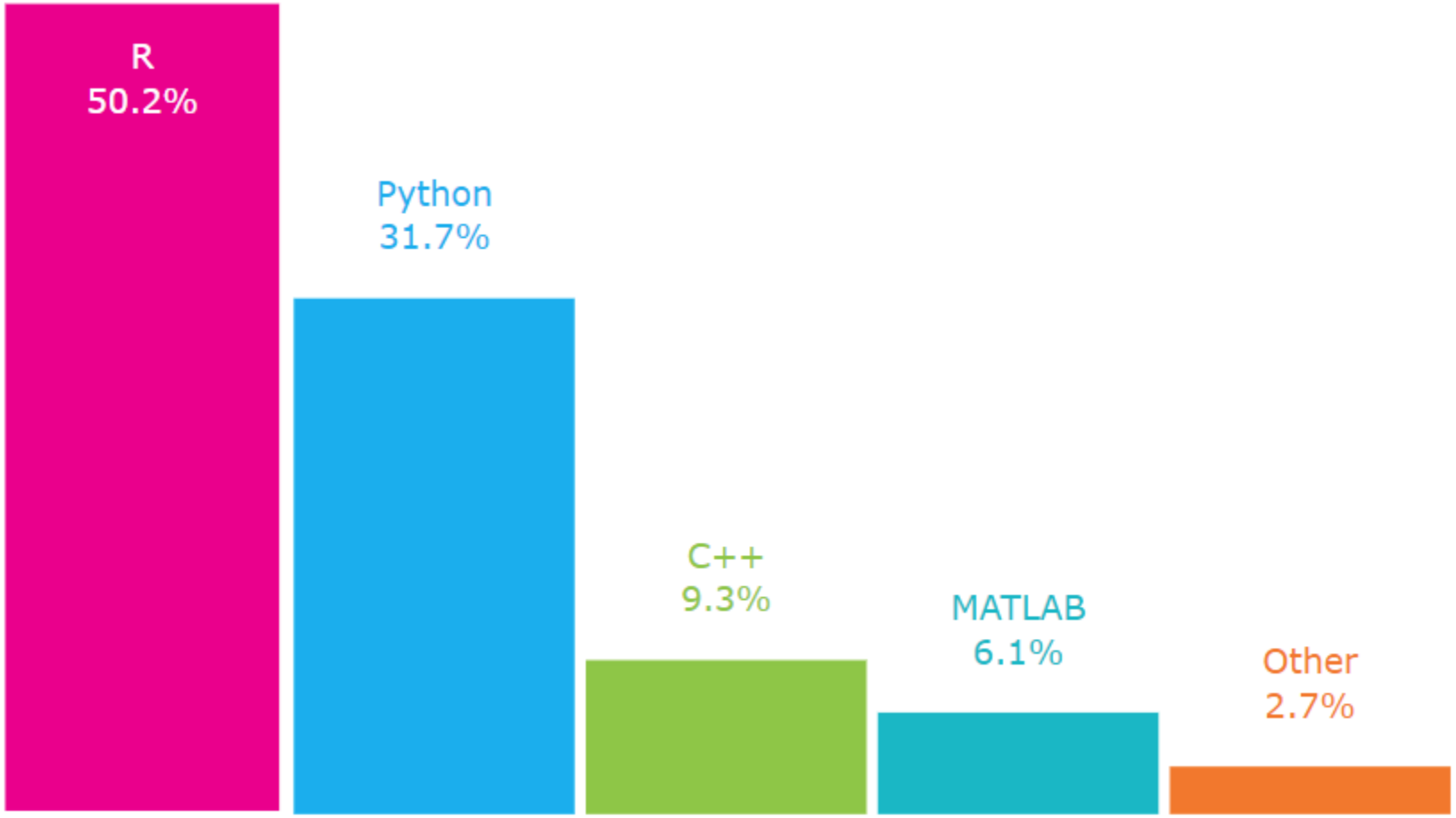
Number of tools over time



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

Jan 31 2021

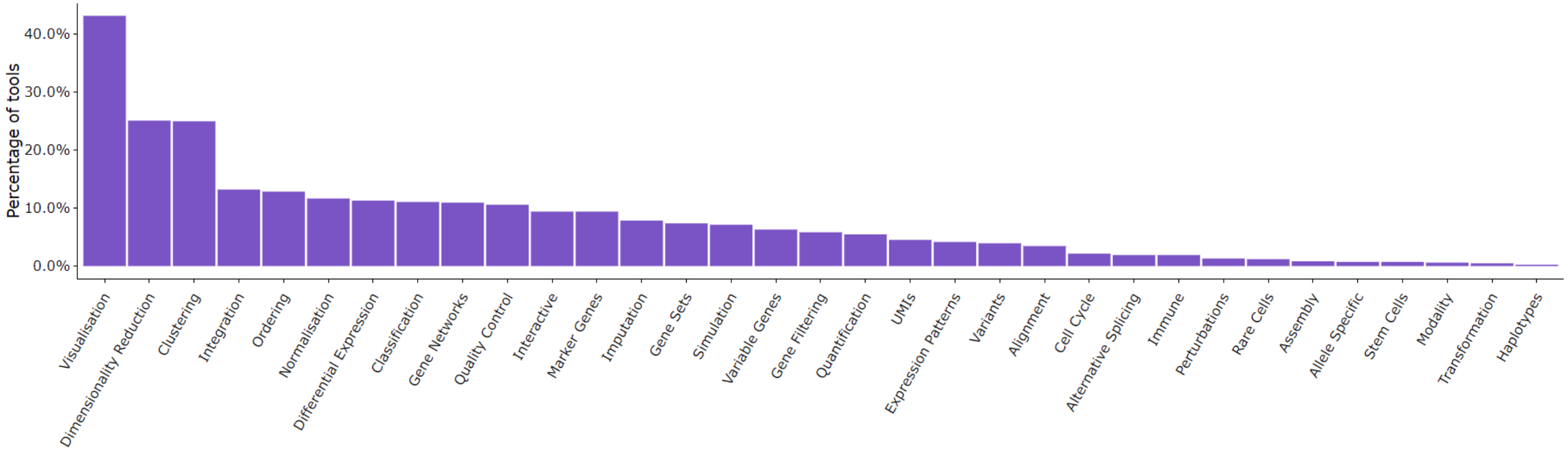
Platforms



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

Jan 31 2021

Categories



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

Jan 31 2021

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

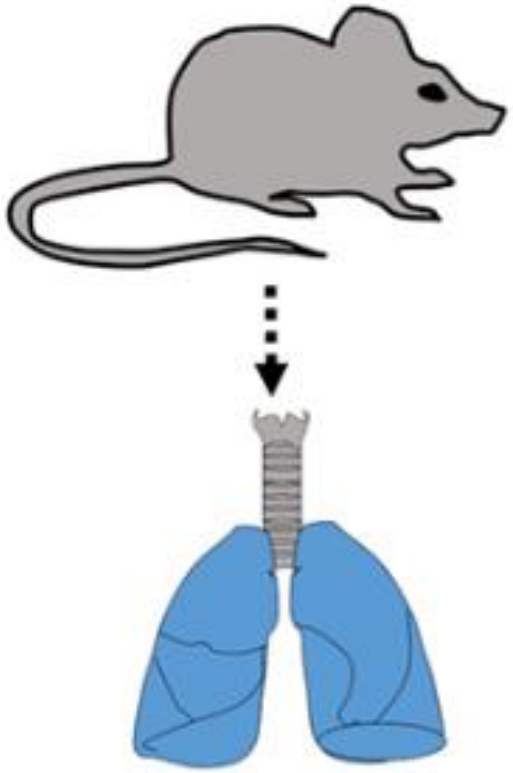
3. SCTransorm
Usage: normalization, highly variable genes, cell-cycle phase assignment
URL: https://github.com/ChristophH/sctransform

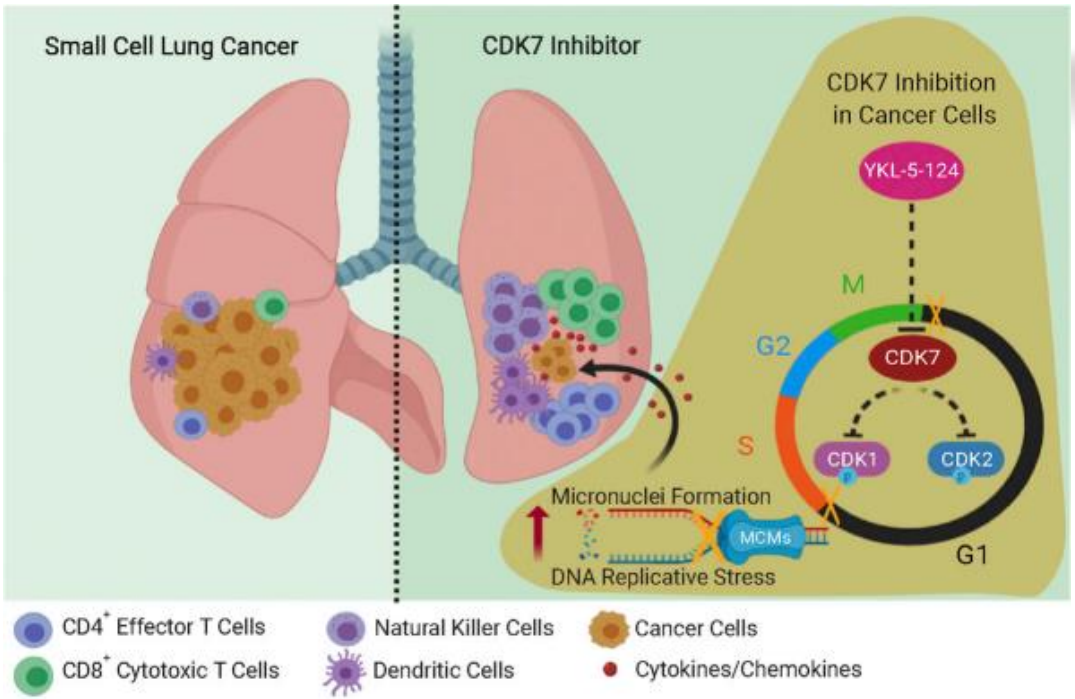
4. Seurat
Usage: dimension reduction, differential expression, visualization
URL: http://satijalab.org/seurat/

5. Monocle3
Usage: Pseudo-time analysis
URL: https://bioconductor.org/packages/release/bioc/html/monocle.html

Dataset 1: Small cell lung cancer

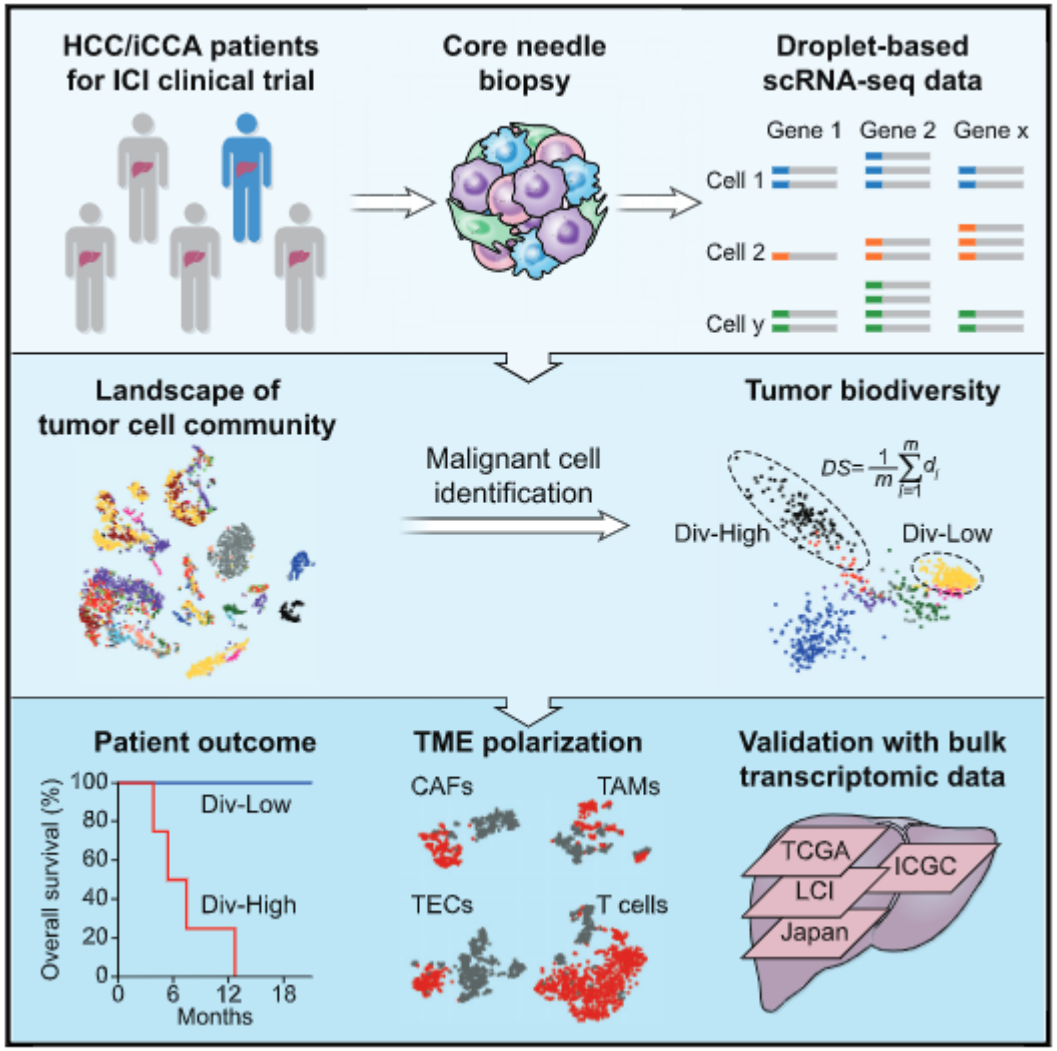
10x Genomics





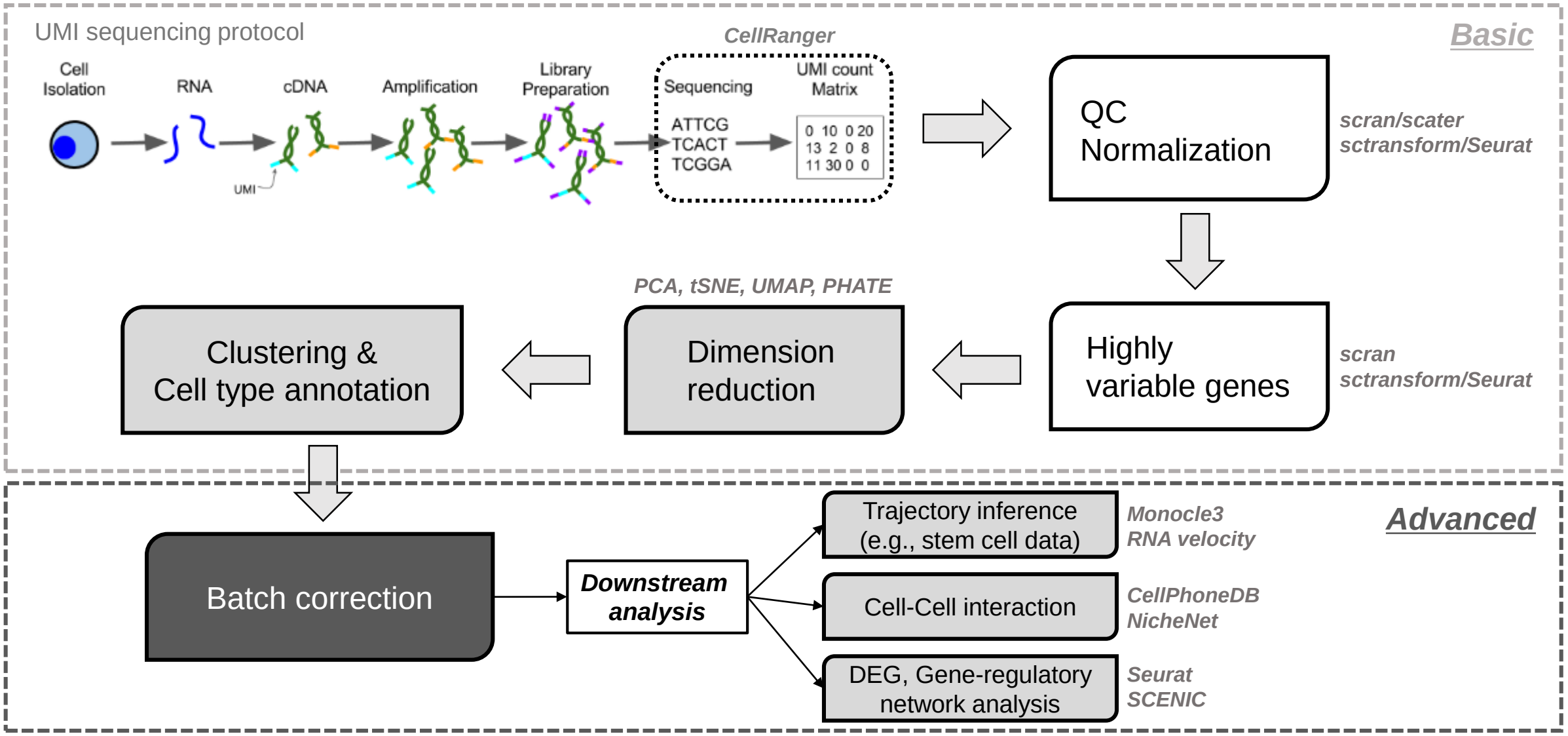
Cancer Cell (2020) 37:37-54.e9

Dataset 2: Liver cancer



Cancer Cell (2019) 36:418-430.e6

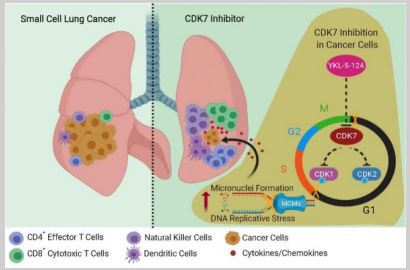
Pipeline for 10X scRNA-seq data



Pipeline for 10X scRNA-seq data

Dataset 1

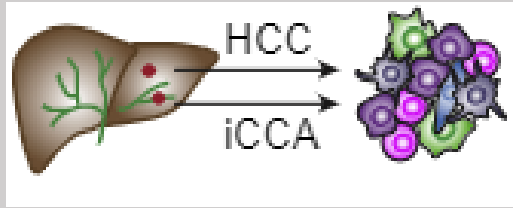
Mouse lung tumor microenvironment



- Cell QC (Empty droplets)

Dataset 2

Human liver cancer



- Cell QC
- Normalization / HVGs selection
- Clustering & visualization
- Cell type annotation
- Cell-Cell interaction

Other pipelines

- Seurat: https://satijalab.org/seurat/
- Scanpy: https://scanpy.readthedocs.io/en/stable/
- Cumulus: https://cumulus.readthedocs.io/en/stable/
- Optimus (HCA): https://data.humancellatlas.org/pipelines/optimus-workflow

Pipeline comparison

- “A systematic evaluation of single cell RNA-seq analysis pipelines” Nature Commun 10:4667 (2019)

- “pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools” Genome Biol 21:227 (2020)

Single cell RNA-seq

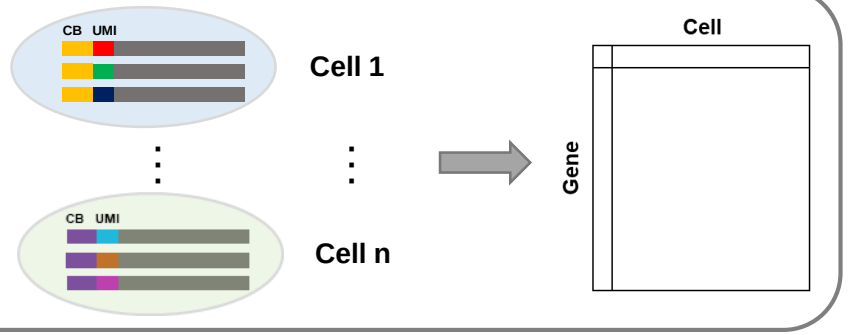
Data generation

Jong Kyoung Kim

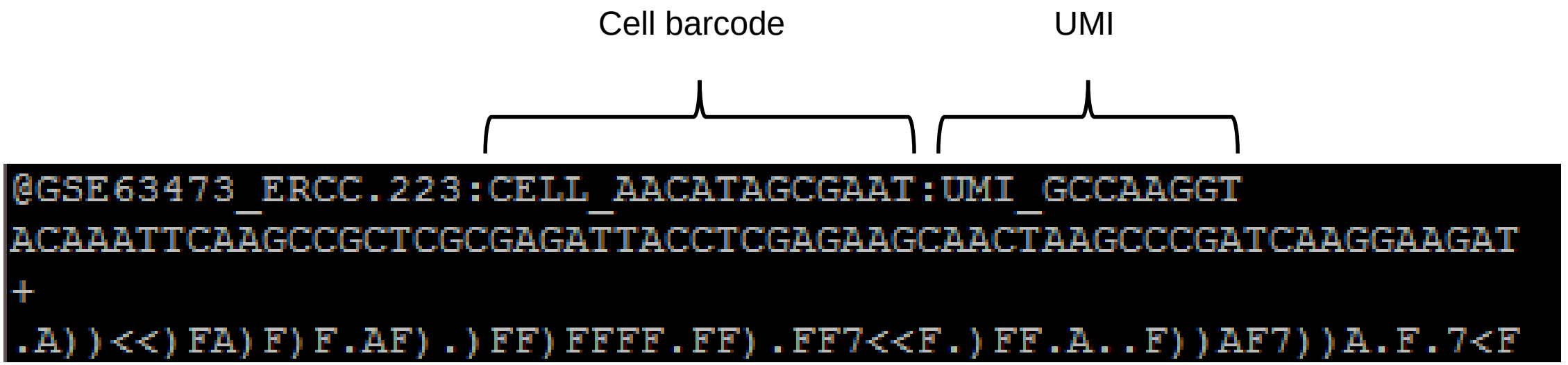
Overview

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



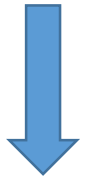
Barcode processing



Read mapping

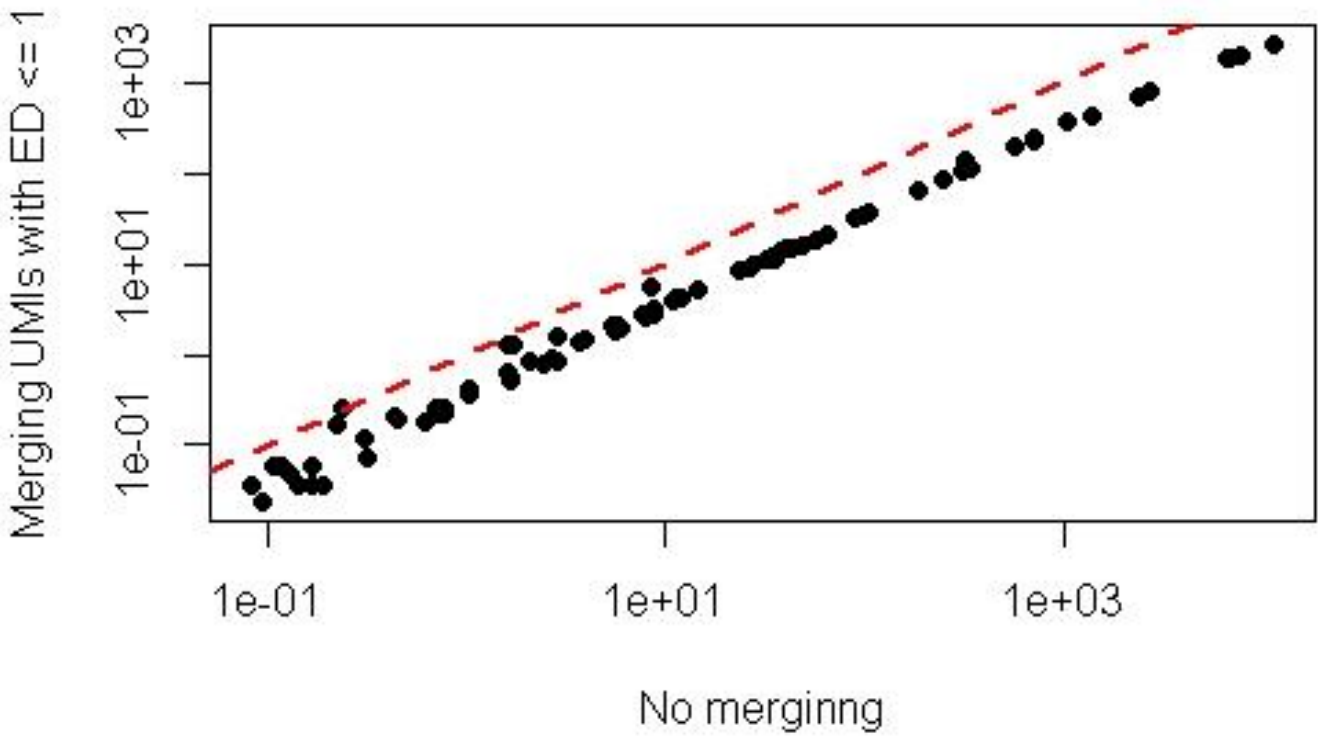
Input sam file

```
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.FFFFFFFFFFFFFFFFFAFF    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.FFFFFFA7FFFFFFFFFFFFFF7FFFFFFF<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

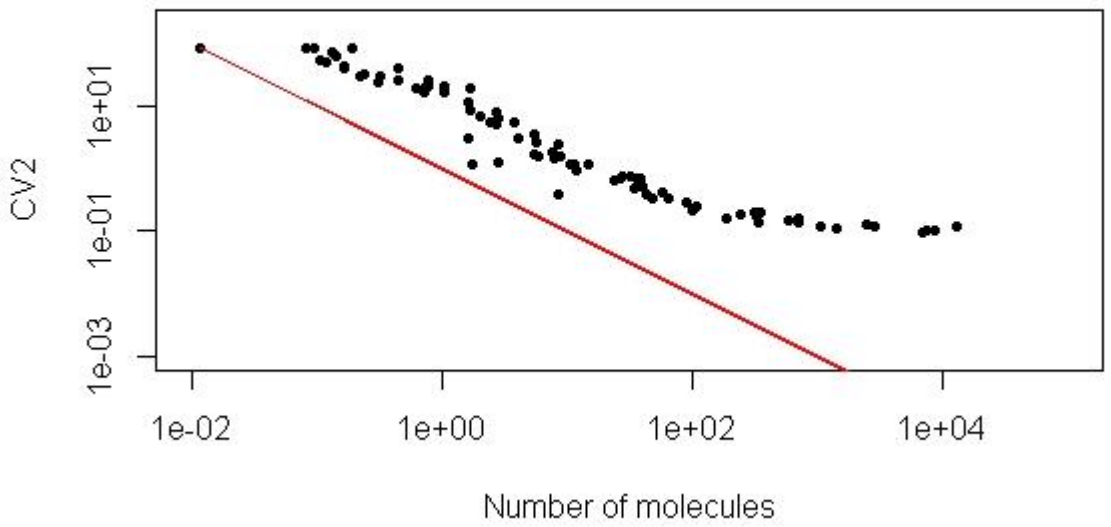
Sequencing error in UMI



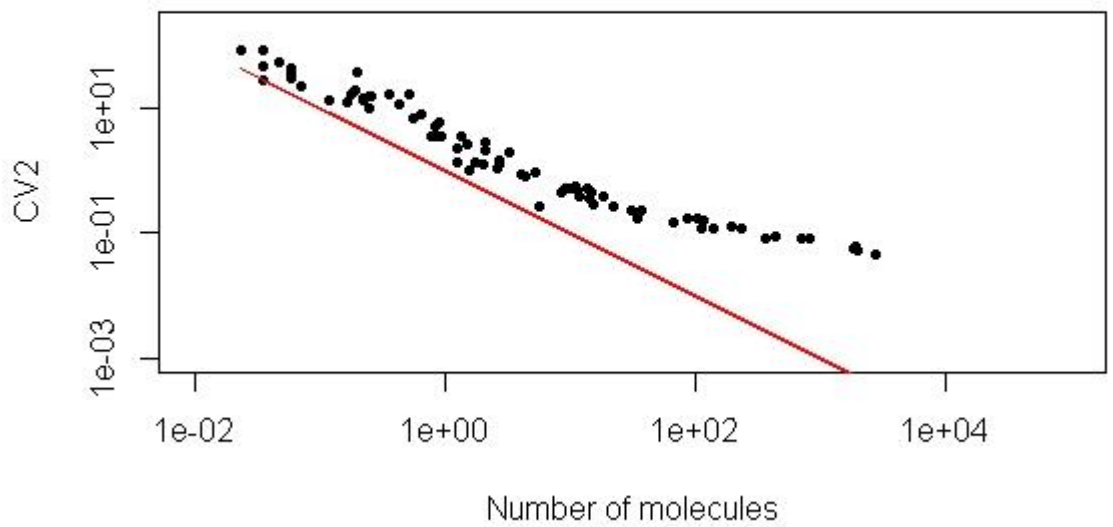
ED = 1
AAAAAAAA
AAACAAAA

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]

zUMIs

Welcome to zUMIs 🔧🚗👊🔧

Quickstart

Nobody likes reading long README files so here is how you get going: Fetch zUMIs by cloning the repository:

```
git clone https://github.com/sdparekh/zUMIs.git
```

Start anaylysing your data:

```
zUMIs/zUMIs.sh -c -y my_config.yaml
```

zUMIs now comes with its own miniconda environment, so you do not need to deal with dependencies or installations (use the -c flag to use conda). If you wish to use your own dependencies, head to the zUMIs wiki to see what is required.

Intro

zUMIs is a fast and flexible pipeline to process RNA-seq data with (or without) UMIs.

The input to this pipeline is simply fastq files. In the most common cases, you will have a read containing the cDNA sequence and other read(s) containing UMI and Cell Barcode information. Furthermore, you will need a STAR index for your genome and GTF annotation file.

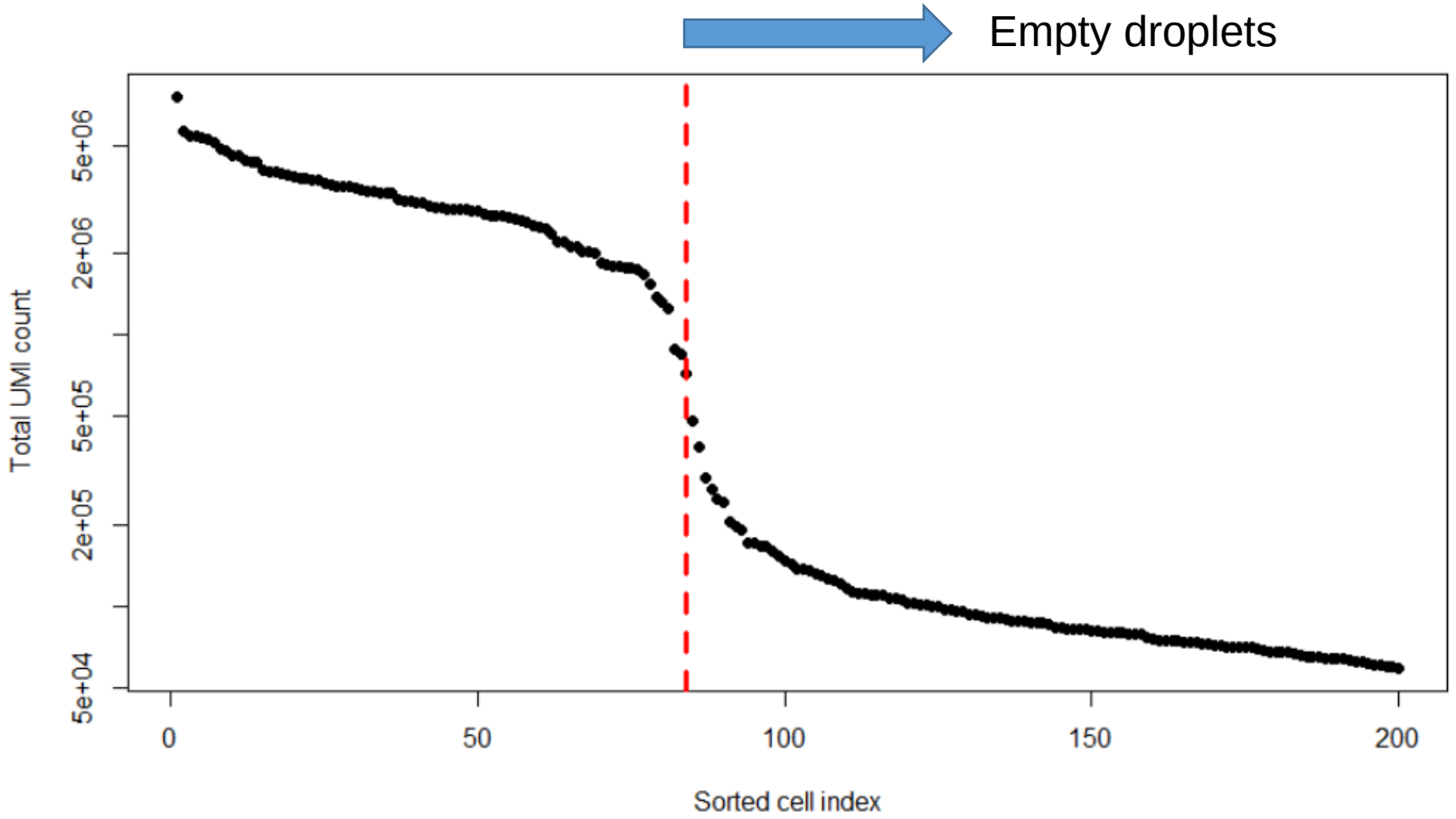
You can read more about zUMIs in our paper!

YAML config Rshiny application

Note: zUMIs is compatible with R release 4.0!

https://github.com/sdparekh/zUMIs

Cell filtering



# of unique cell index: 812,692

Minimum Phred quality score within 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.

DropletUtils

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

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

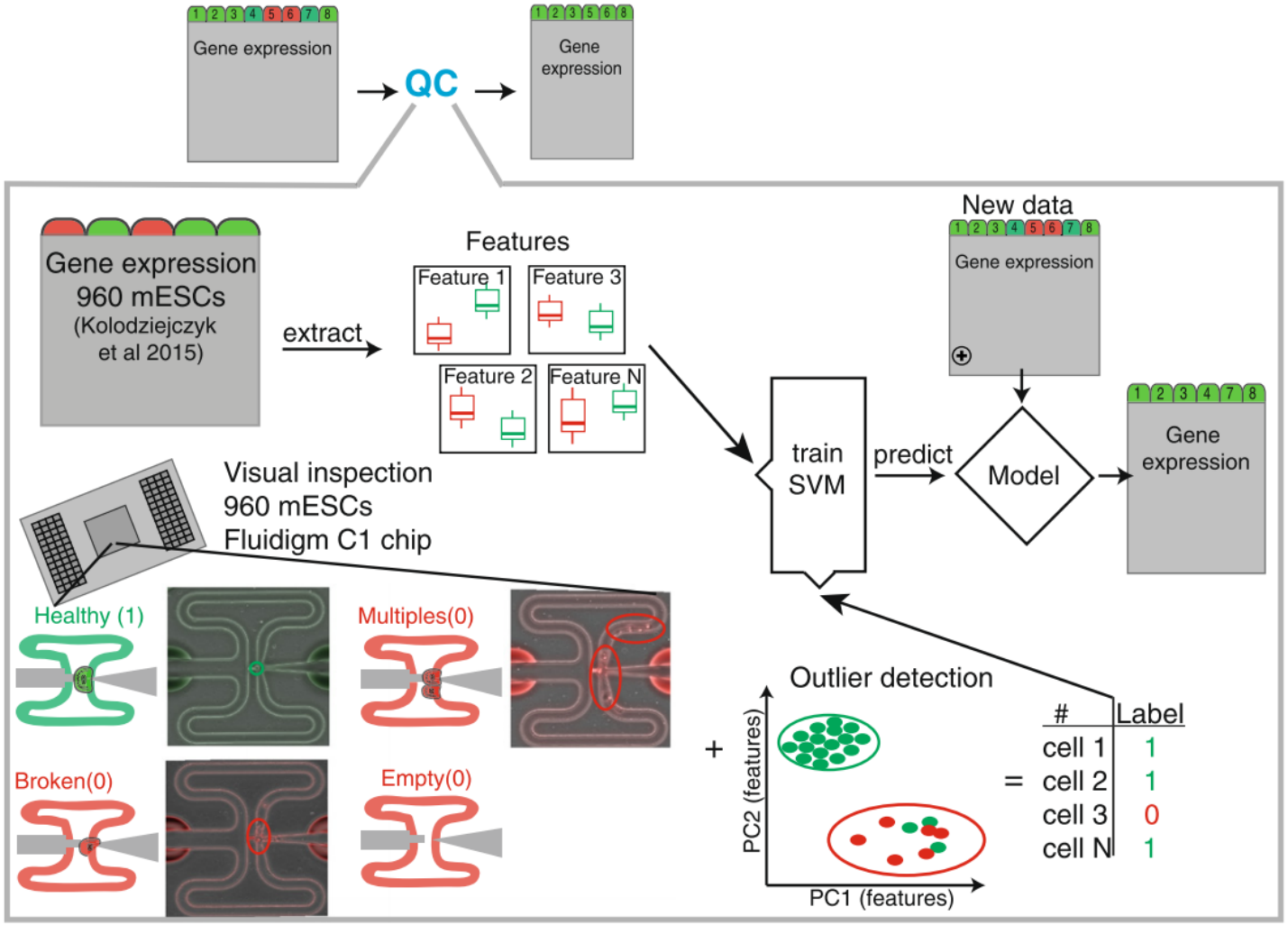
5. Compute the P-value for b using a Monte Carlo approach.

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

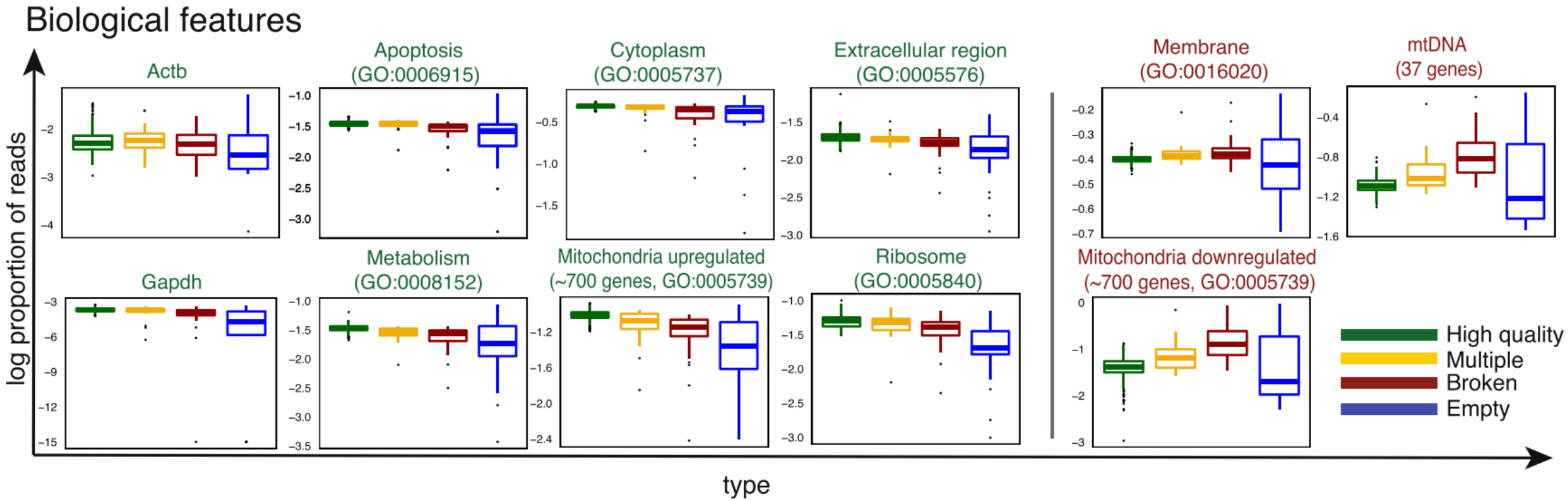
Classification of low-quality cells

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

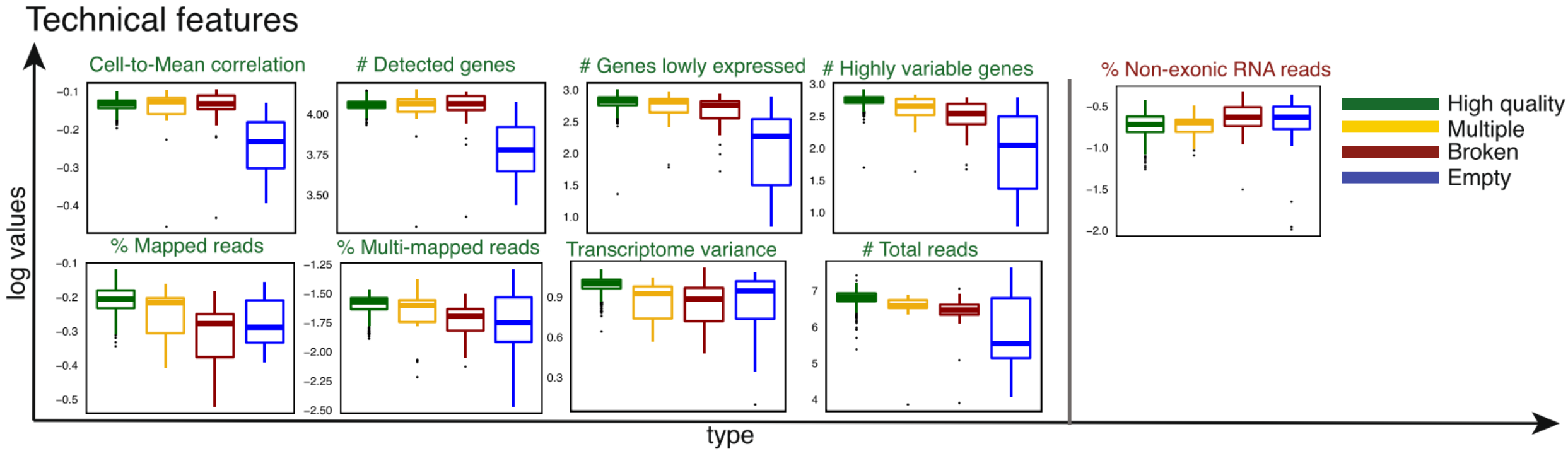
Classification of low-quality cells



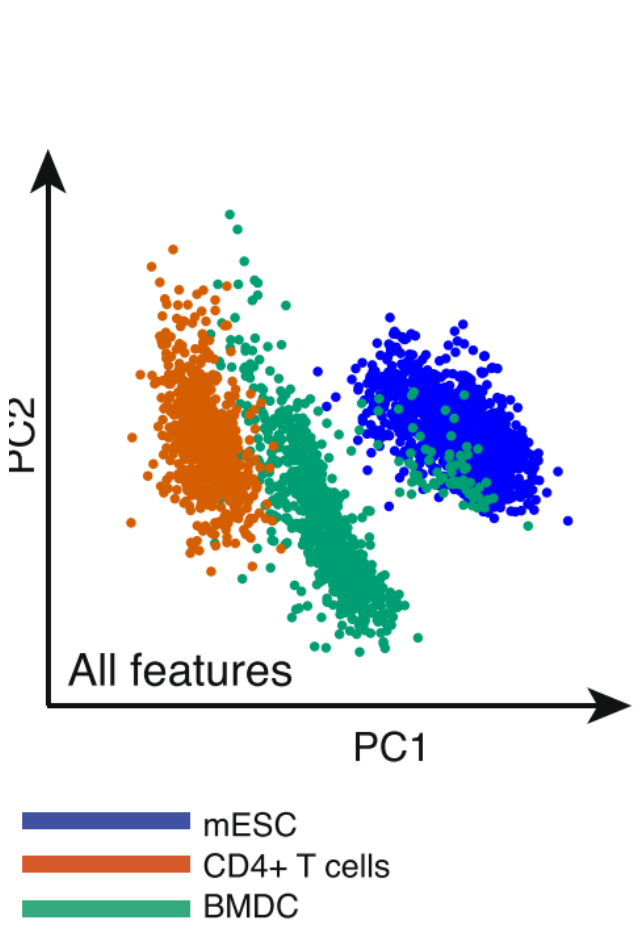
Feature selection

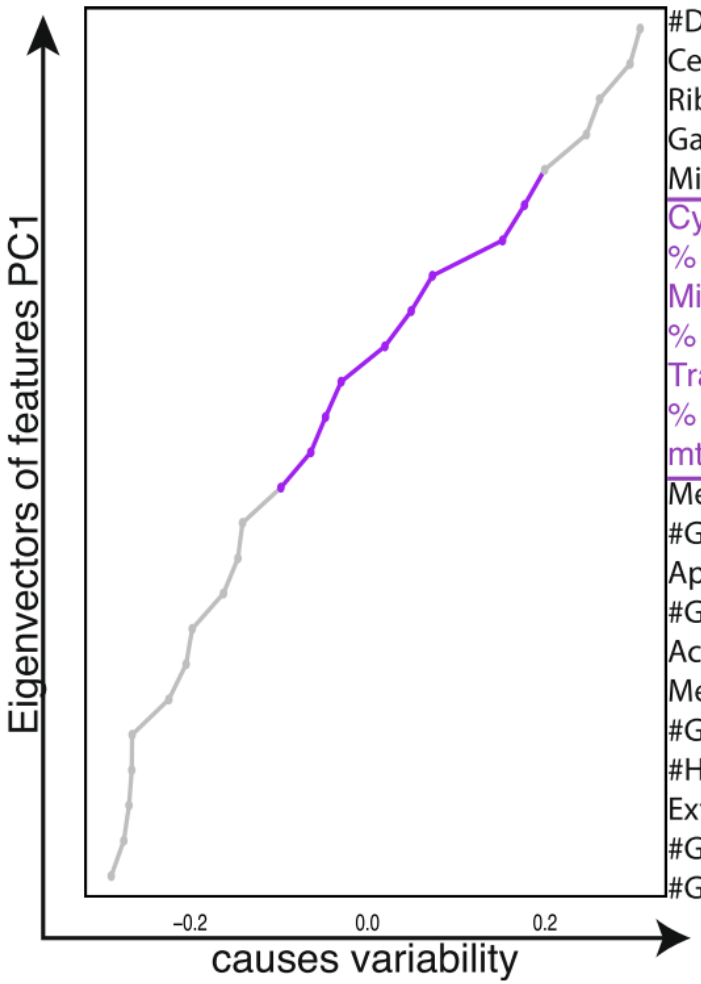


Feature selection

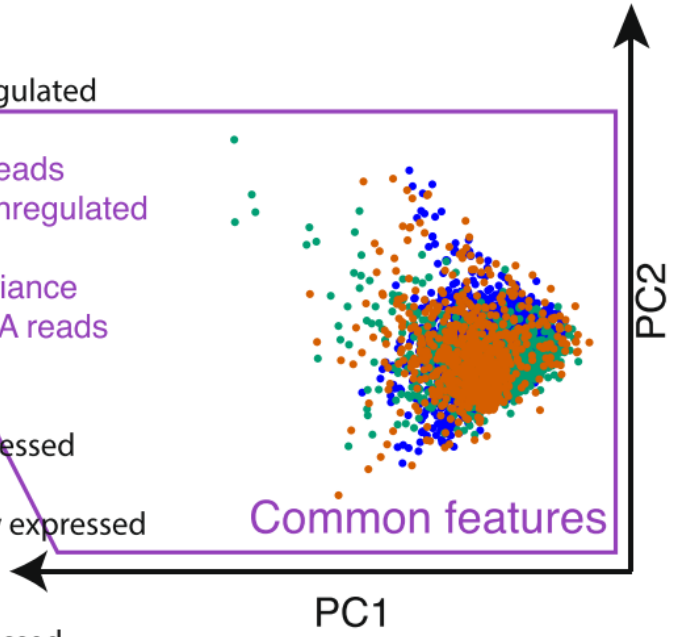


Feature selection

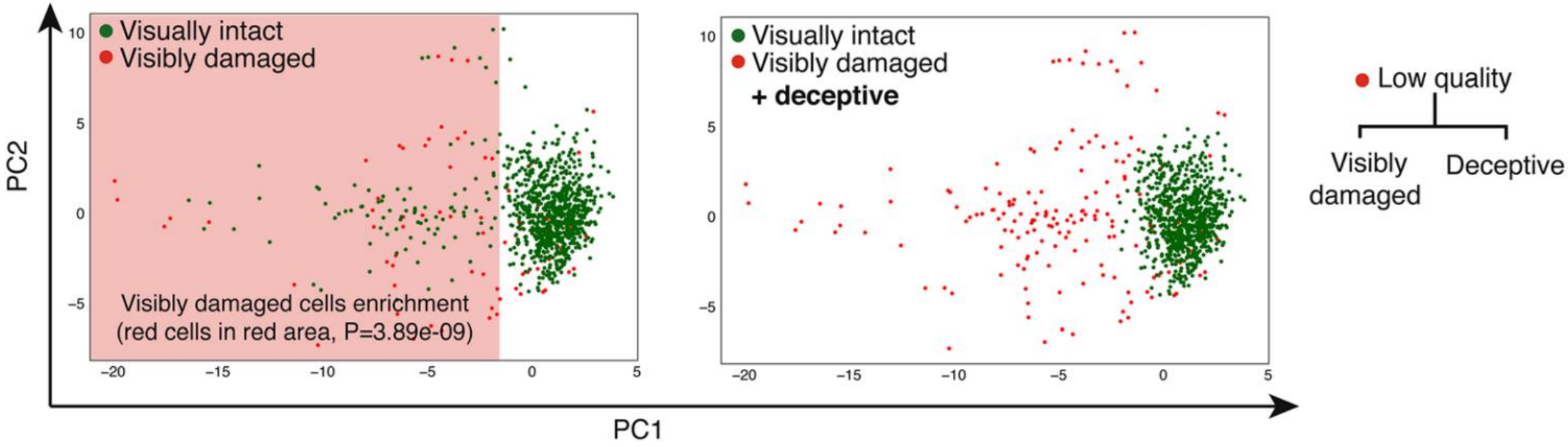




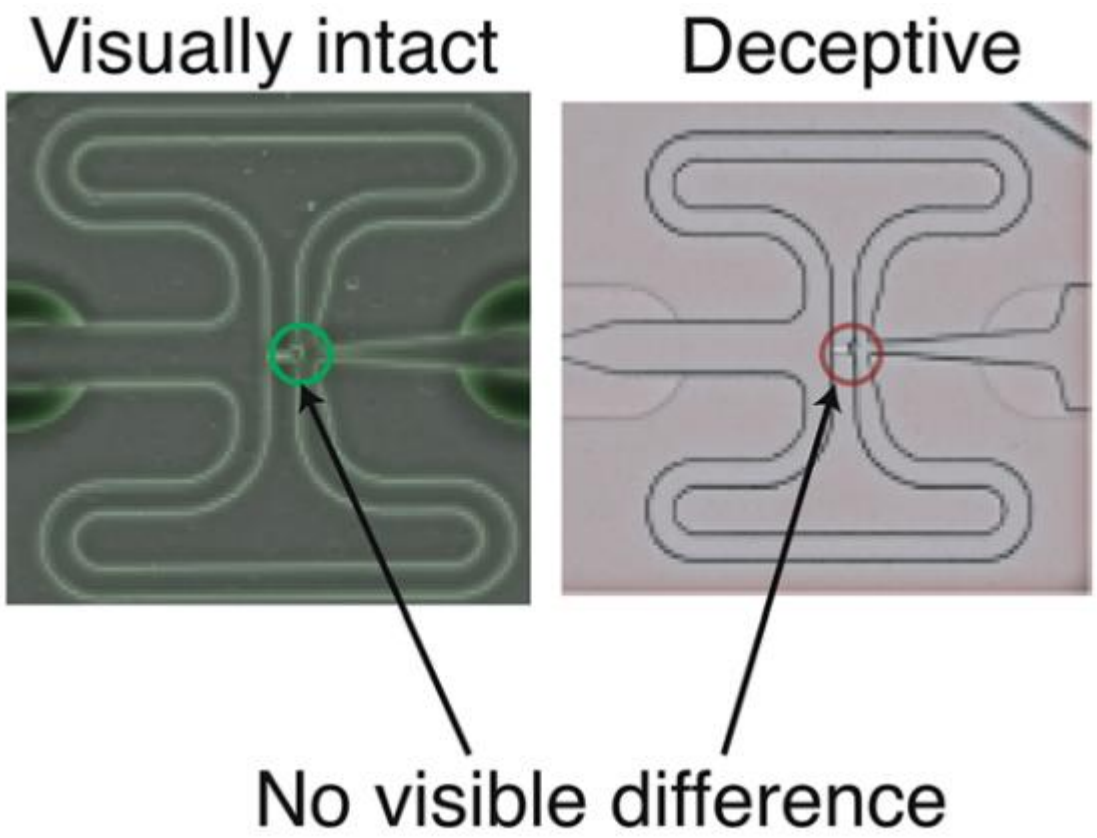
#Detected genes
Cell-to-Mean correlation
Ribosomes
Gapdh
Mitochondria upregulated
Cytoplasm
% Multi-mapped reads
Mitochondria downregulated
% Mapped reads
Transcriptome variance
% Non-exonic RNA reads
mtDNA
Metabolism
#Genes highly expressed
Apoptosis
#Genes moderately expressed
Actb
Membrane
#Genes lowly expressed
#Highly expressed and variable genes
Extracellular region
#Genes very highly expressed
#Genes very lowly expressed

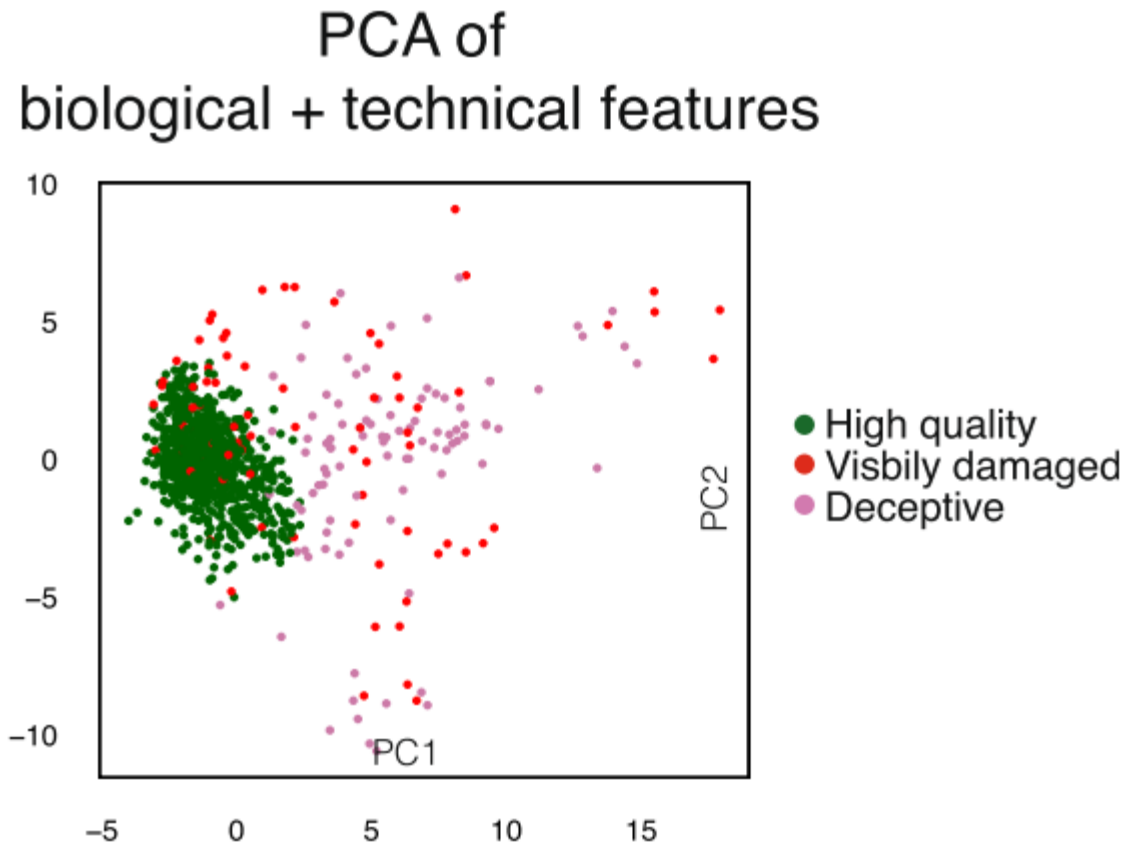


Hidden bad-quality cells

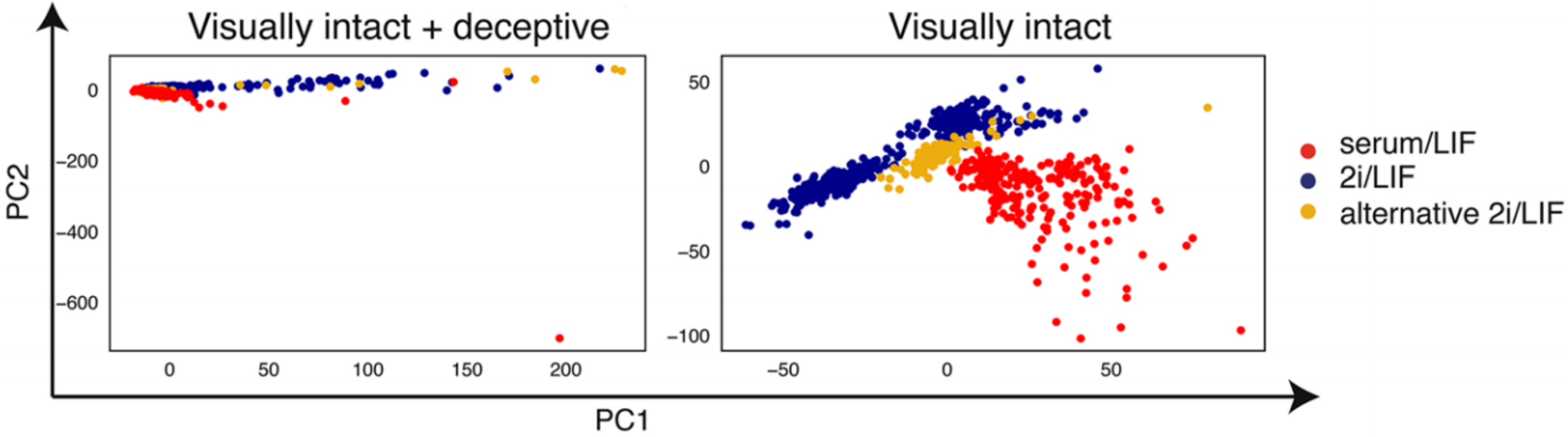


Hidden bad-quality cells





Hidden bad-quality cells



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. Simple calculation of many **quality control metrics** from the expression data;

3. Many tools for visualising scRNA-seq data, especially **diagnostic plots for quality control**;

4. Subsetting and many other methods for **filtering out problematic cells and features**;

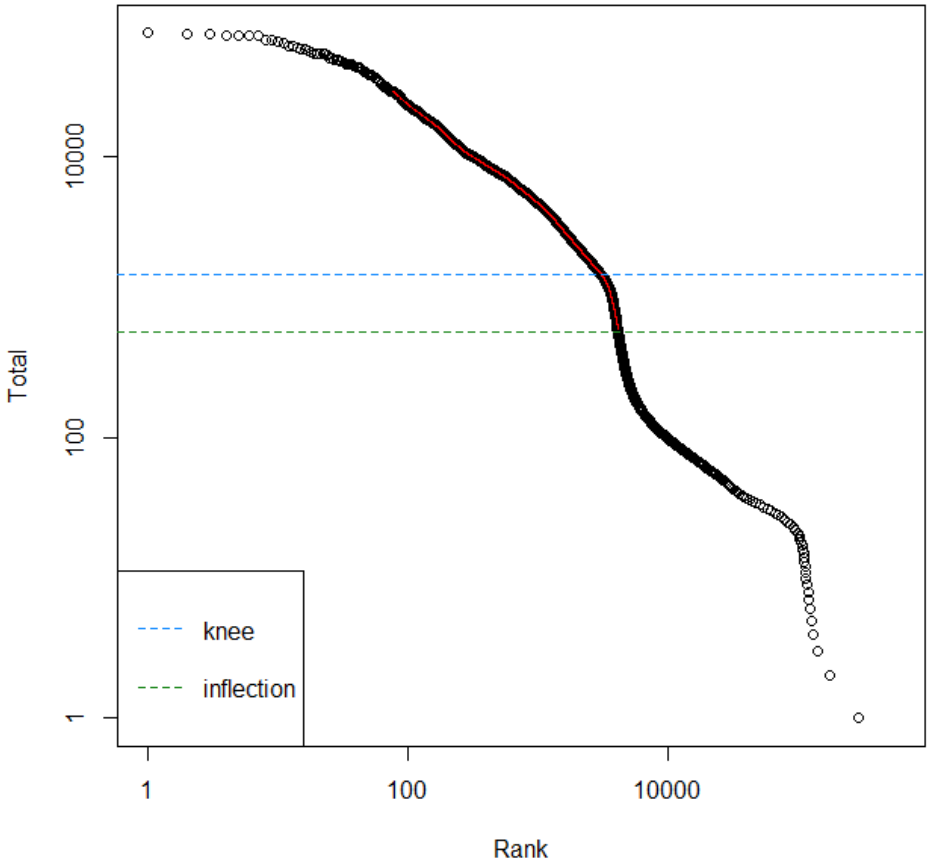
5. Methods for identifying important experimental variables and normalising data ahead of downstream statistical analysis and modeling.

Bioinformatics (2017) 33:1179

CellQC (DropletUtils): build UMI barcode rank (Dataset 1)

```
sce <- read10xCounts(dir10x)
rownames(sce) = uniquifyFeatureNames(rowData(sce)$ID, rowData(sce)$Symbol)
colnames(sce) = sce$Barcode

my.counts=counts(sce)

# barcode rank by total UMI counts
br.out <- barcodeRanks(my.counts)

# Making a plot.
plot(br.out$rank, br.out$total, log="xy", xlab="Rank", ylab="Total")
o <- order(br.out$rank)
lines(br.out$rank[o], br.out$fitted[o], col="red")

abline(h=metadata(br.out)$knee, col="dodgerblue", lty=2)
abline(h=metadata(br.out)$inflection, col="forestgreen", lty=2)
legend("bottomleft", lty=2, col=c("dodgerblue", "forestgreen"),
       legend=c("knee", "inflection"))
```

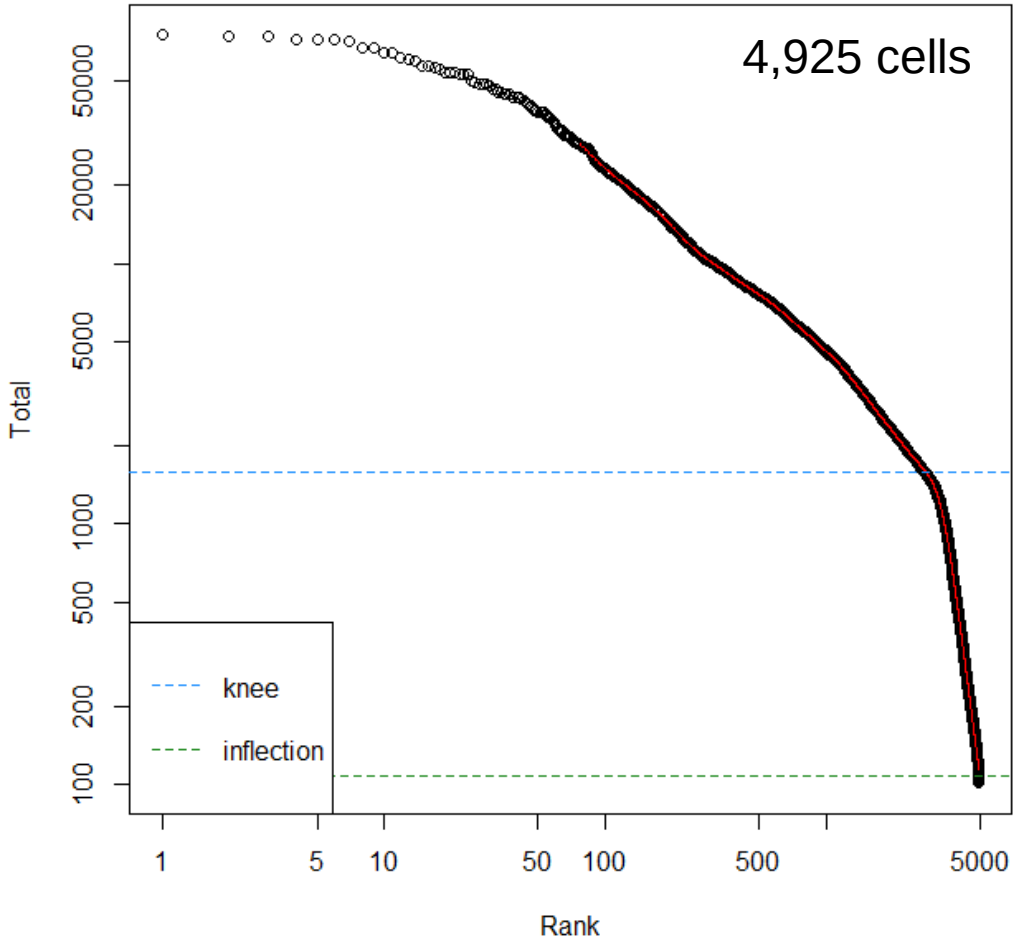
737,280 droplets



CellQC (DropletUtils): detect empty droplets (Dataset 1)

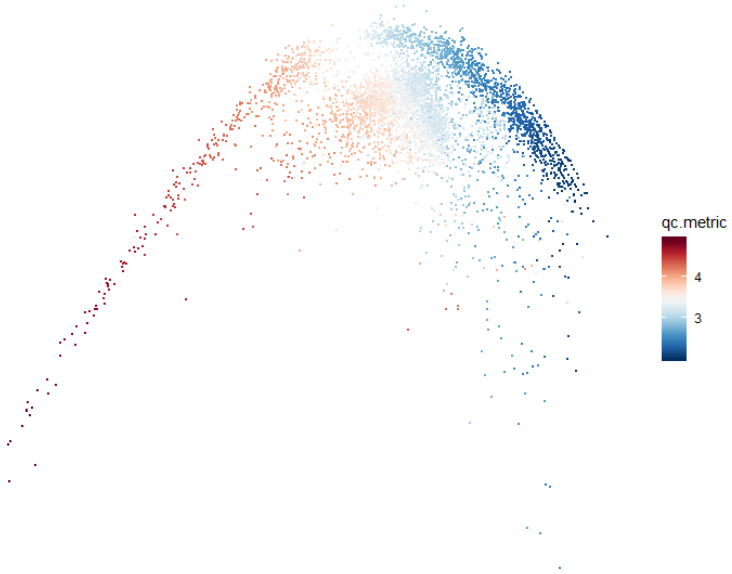
```
set.seed(100)
e.out <- emptyDrops(my.counts)
e.out

is.cell <- e.out$FDR < 0.01
sum(is.cell, na.rm=TRUE)

# remove empty droplet
sce_ne=sce[,which(is.cell)]
sce_ne=sce_ne[rowSums(counts(sce_ne))>0,]
```

```
class: SingleCellExperiment
dim: 19574 4925
metadata(1): Samples
assays(1): counts
rownames(19574): Xkr4 Rp1 ... CAAA01118383.1 CAAA01147332.1
rowData names(3): ID Symbol NA
colnames(4925): AAACCTGAGACTTTCG-1 AAACCTGAGAGACTAT-1 ... TTT
colData names(2): Sample Barcode
reducedDimNames(0):
spikeNames(0):
altExpNames(0):
```

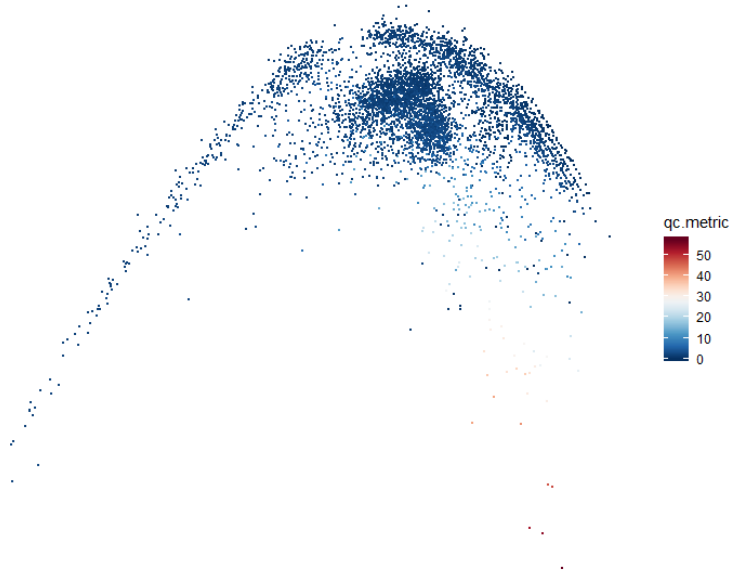


CellQC (Scater): Removing of low quality cell

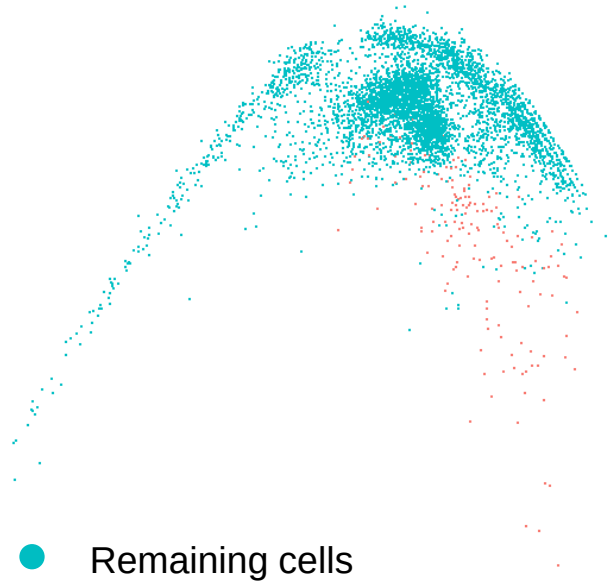
log10 (Total UMIs+1) > 100



Mitochondrial mapped reads percentage > 10



Remaining cells: 4,760 cells



Single cell RNA-seq

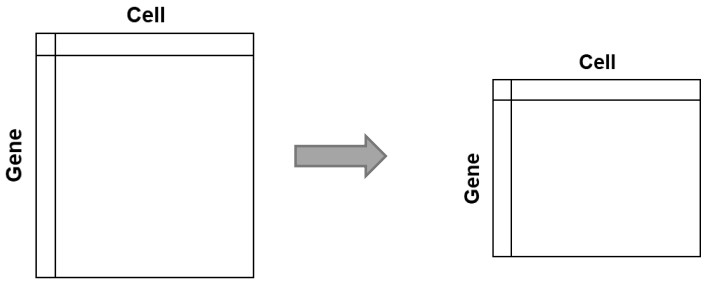
Data processing

Jong Kyoung Kim

Overview

Data processing

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



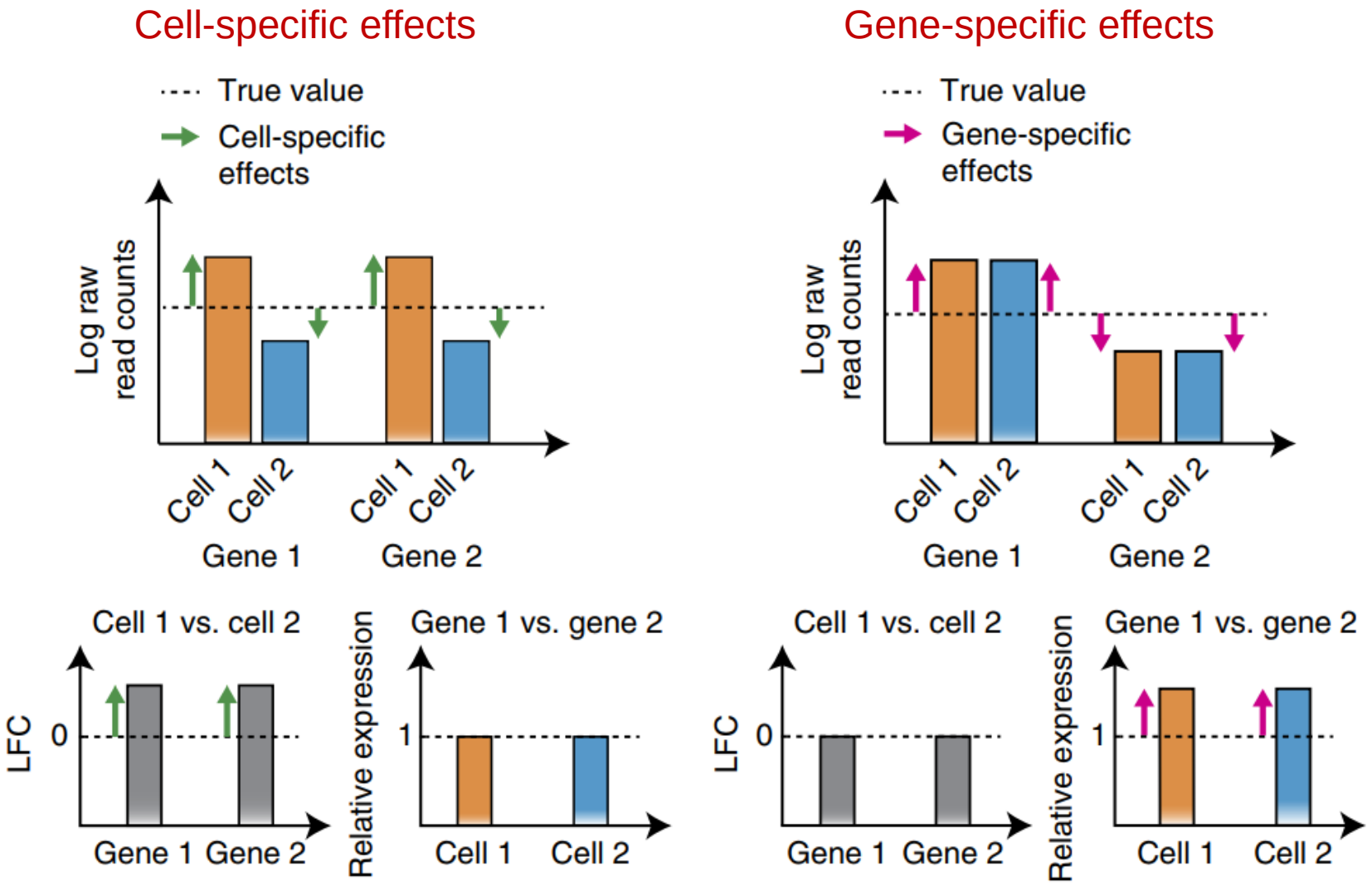
Normalization

- 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



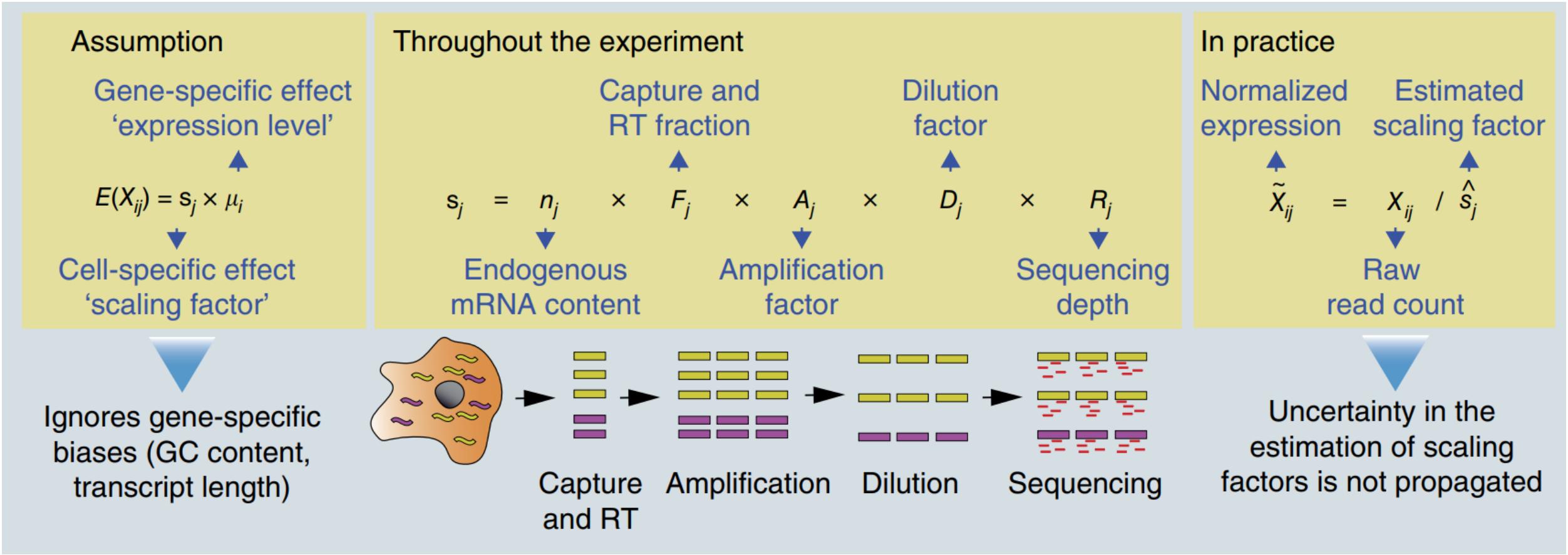
Nat Methods (2017) 14:565

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 (2017) 14:565

Global scaling factor



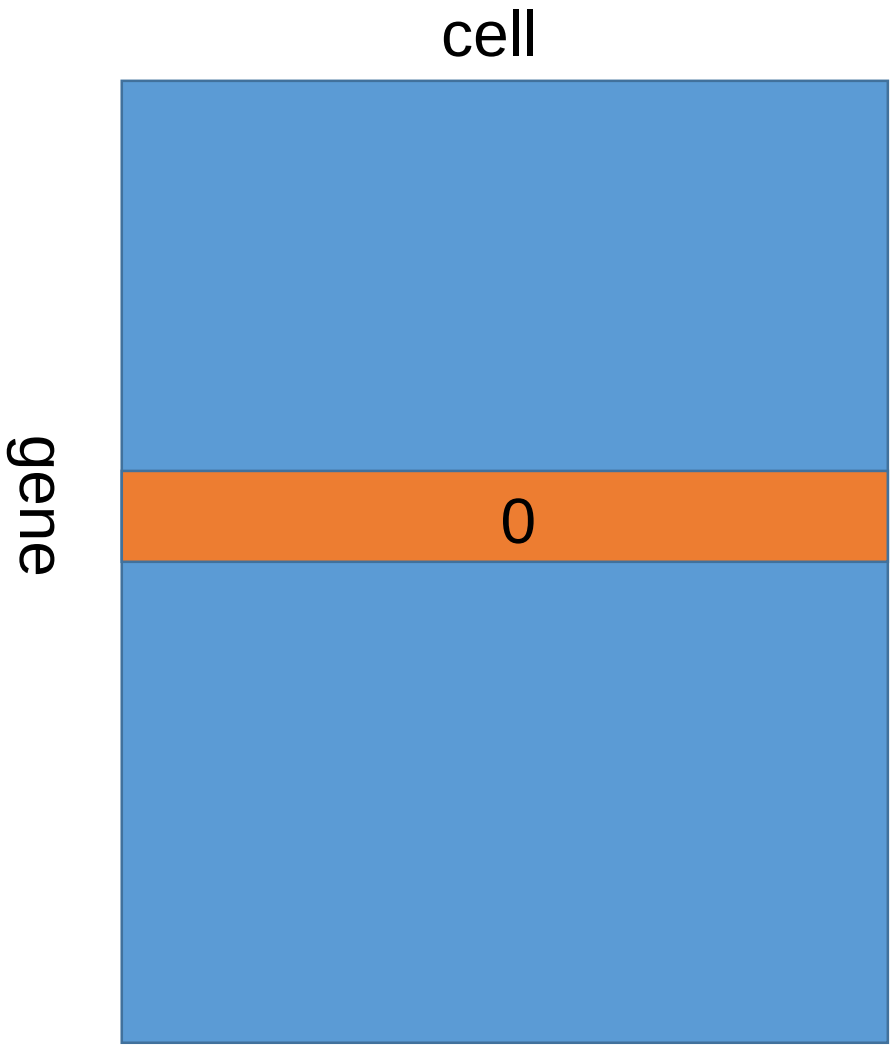
Nat Methods (2017) 14:565

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

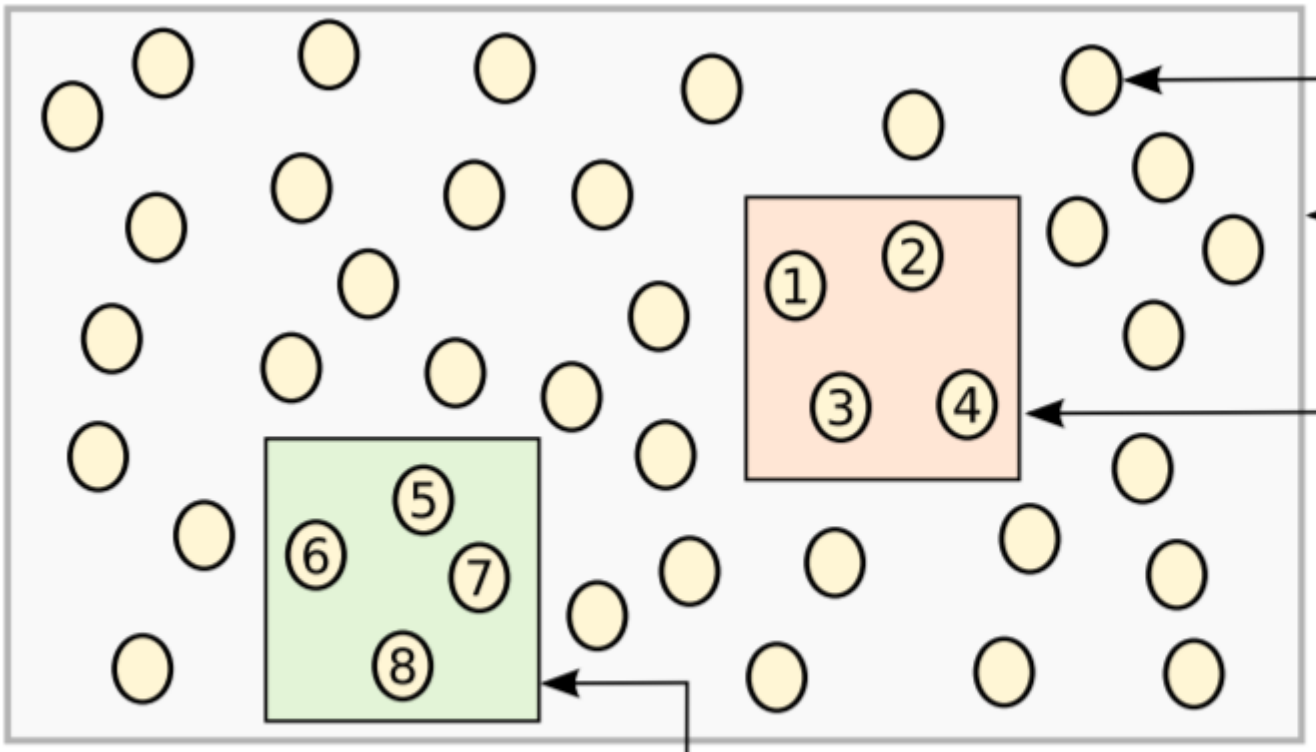


NA

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:

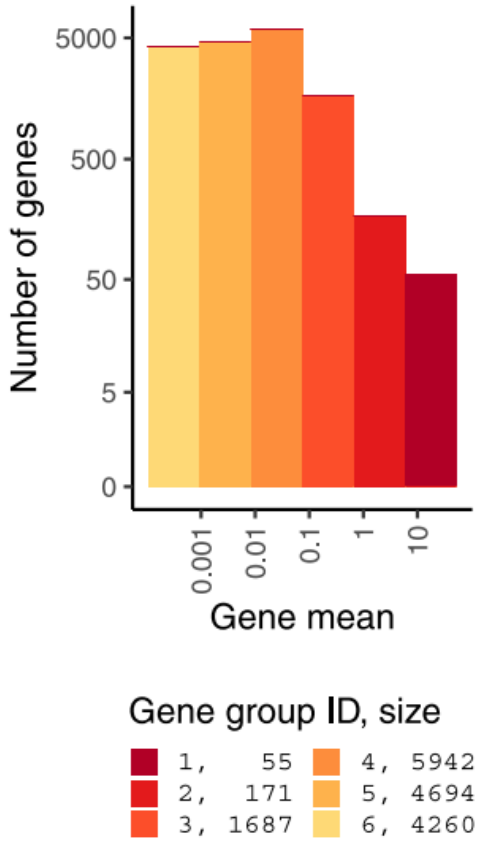
$$\begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,\ldots \\ 0\,0\,0\,0\,1\,1\,1\,1\,\ldots \\ 1\,0\,1\,0\,1\,0\,1\,0\,\ldots \\ 0\,1\,1\,0\,1\,1\,0\,0\,\ldots \\ \ldots \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \ldots \end{bmatrix} = \begin{bmatrix} \theta_A \\ \theta_B \\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

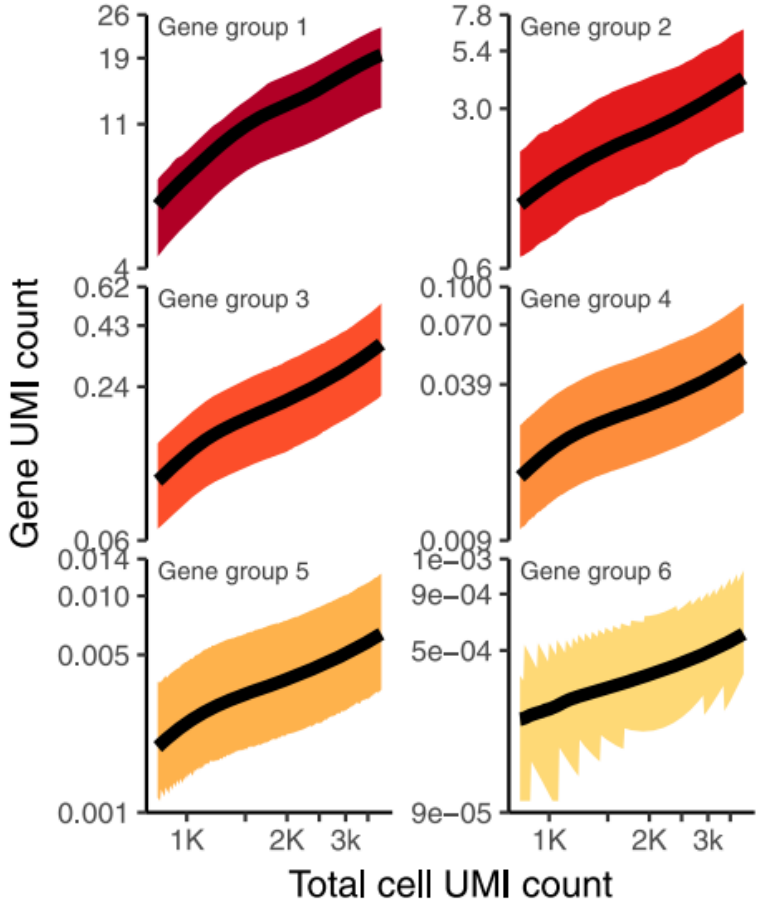
Genome Biol (2016) 17:75

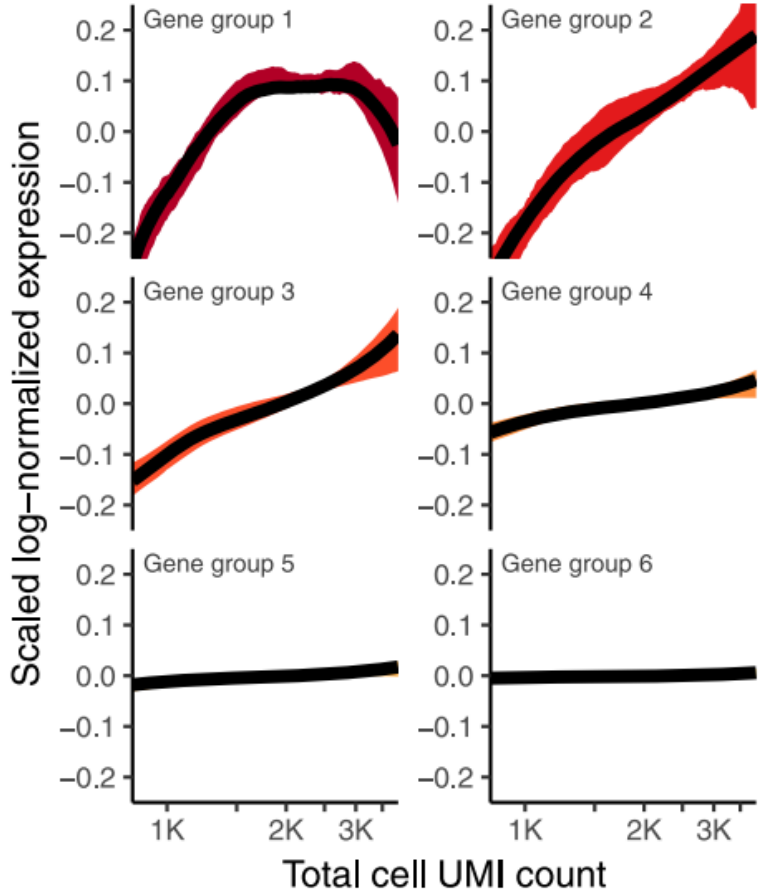
scran

- 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

SCTransform: motivation (not a default)







Genome Biol (2019) 20:296

33,148 PBMC dataset from 10X Genomics

SCTransform: regularized NB regression

$$\log(\mathbb{E}(x_i)) = \beta_0 + \beta_1 \log_{10} m$$

x_i: the vector of UMI counts assigned to gene i, $x_i = \sum_j x_{ij}$

m: the vector of molecules assigned to cells, $m_j = \sum_i x_{ij}$

Learning procedures:

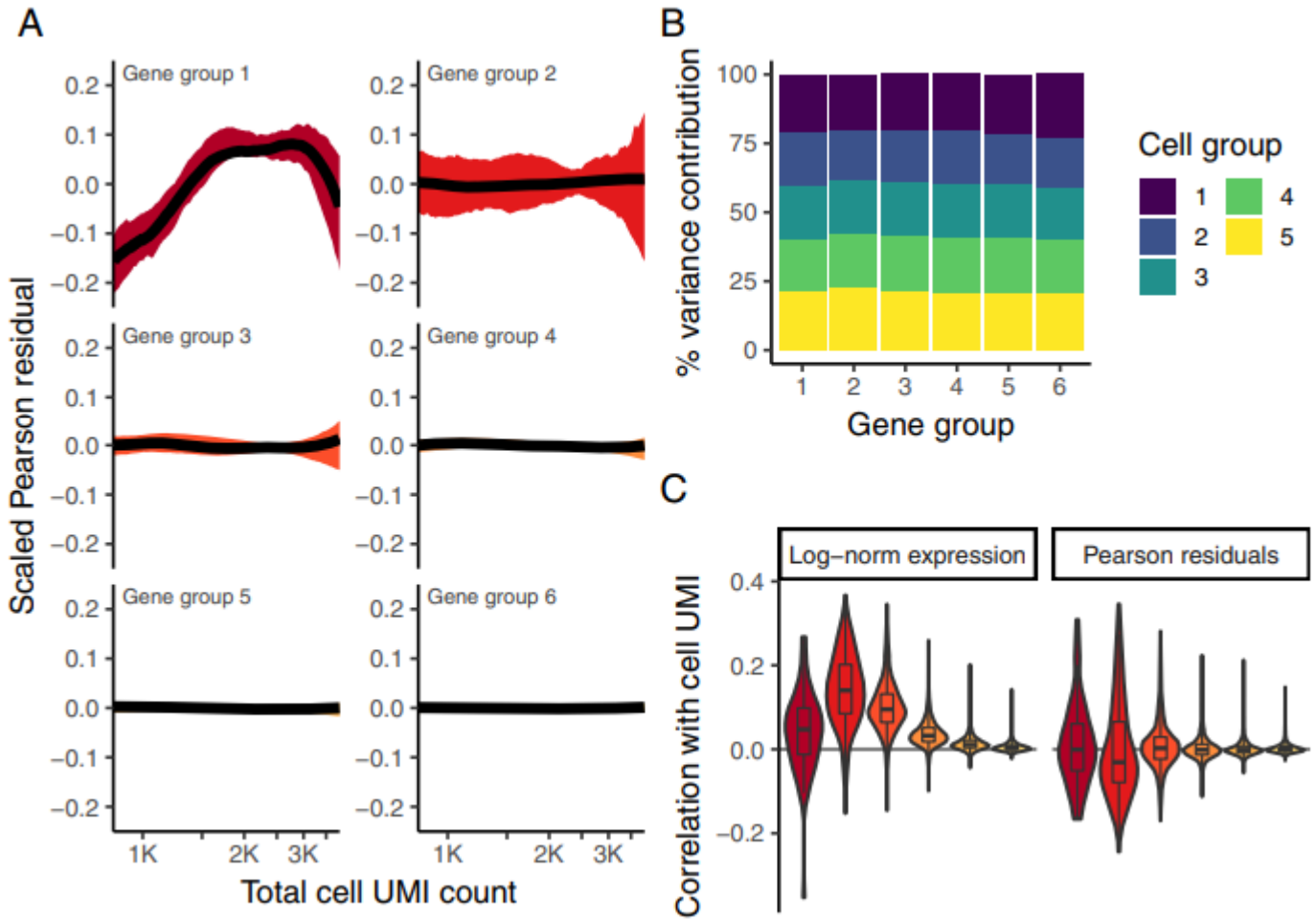
1. We fit independent regression models per gene
2. We exploit the relationship of model parameter values and gene mean to learn global trends in the data (using ksmooth function in R). We perform independent regularizations for all parameters.
3. We use the regularized regression parameters to define an affine function that transforms UMI counts into Pearson residuals:

$$z_{ij} = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}},$$

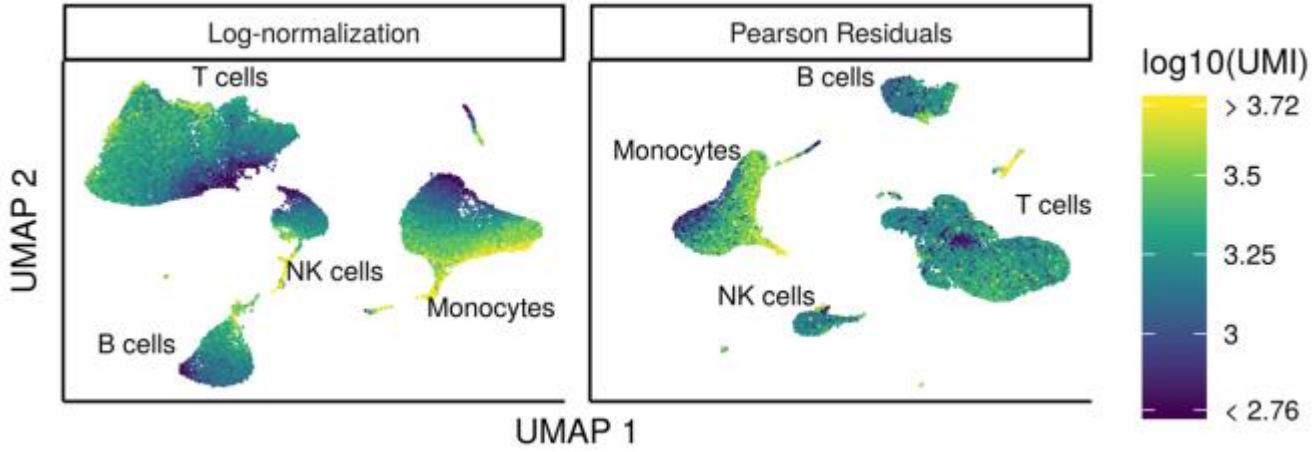
$$\mu_{ij} = \exp\left(\beta_{0_i} + \beta_{1_i} \log_{10} m_j\right),$$

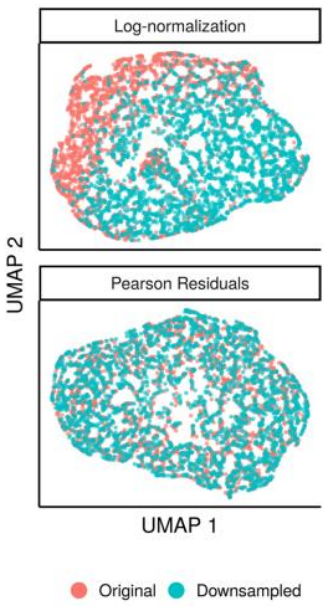
$$\sigma_{ij} = \sqrt{\mu_{ij} + \frac{\mu_{ij}^2}{\theta_i}},$$

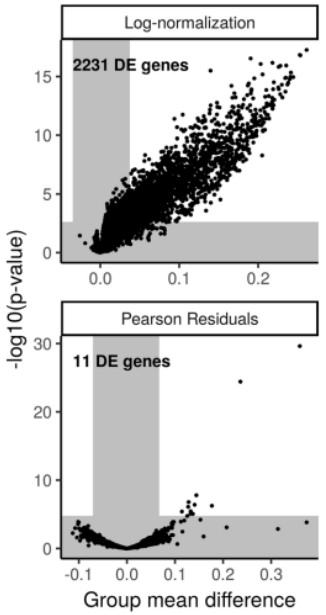
SCTransform: regularized NB regression

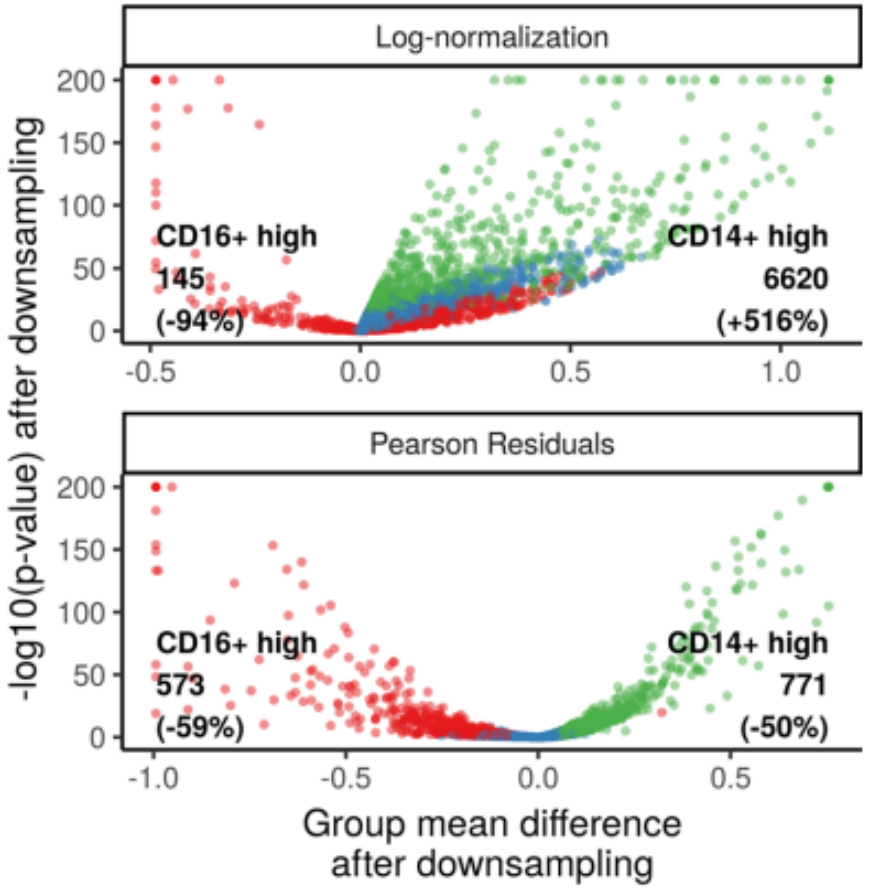


SCTransform: downstream analysis





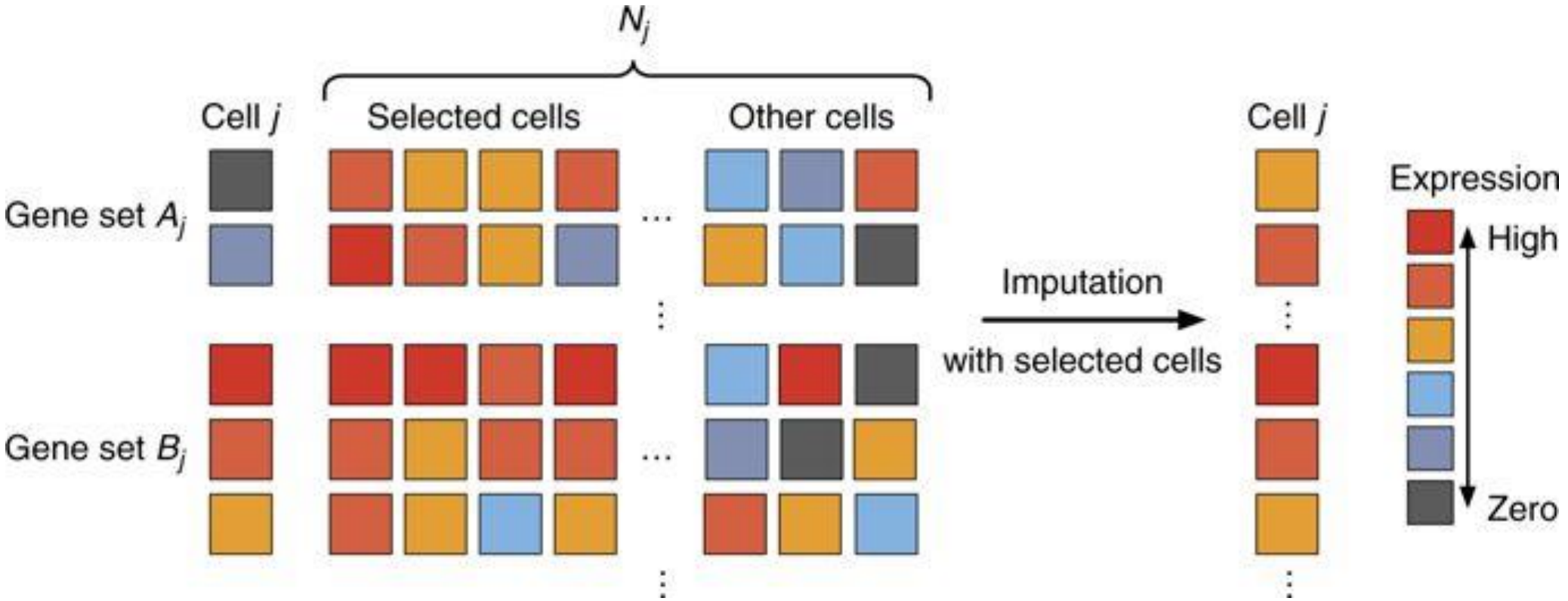




Imputation

- A high frequency of zero counts is driven by:
  - Stochastic gene expression
  - Low mRNA capture efficiency
  - Low sequencing depth
- This sparseness leads to high technical variability in gene expression.
- Because 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 (2018) 9:997

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.

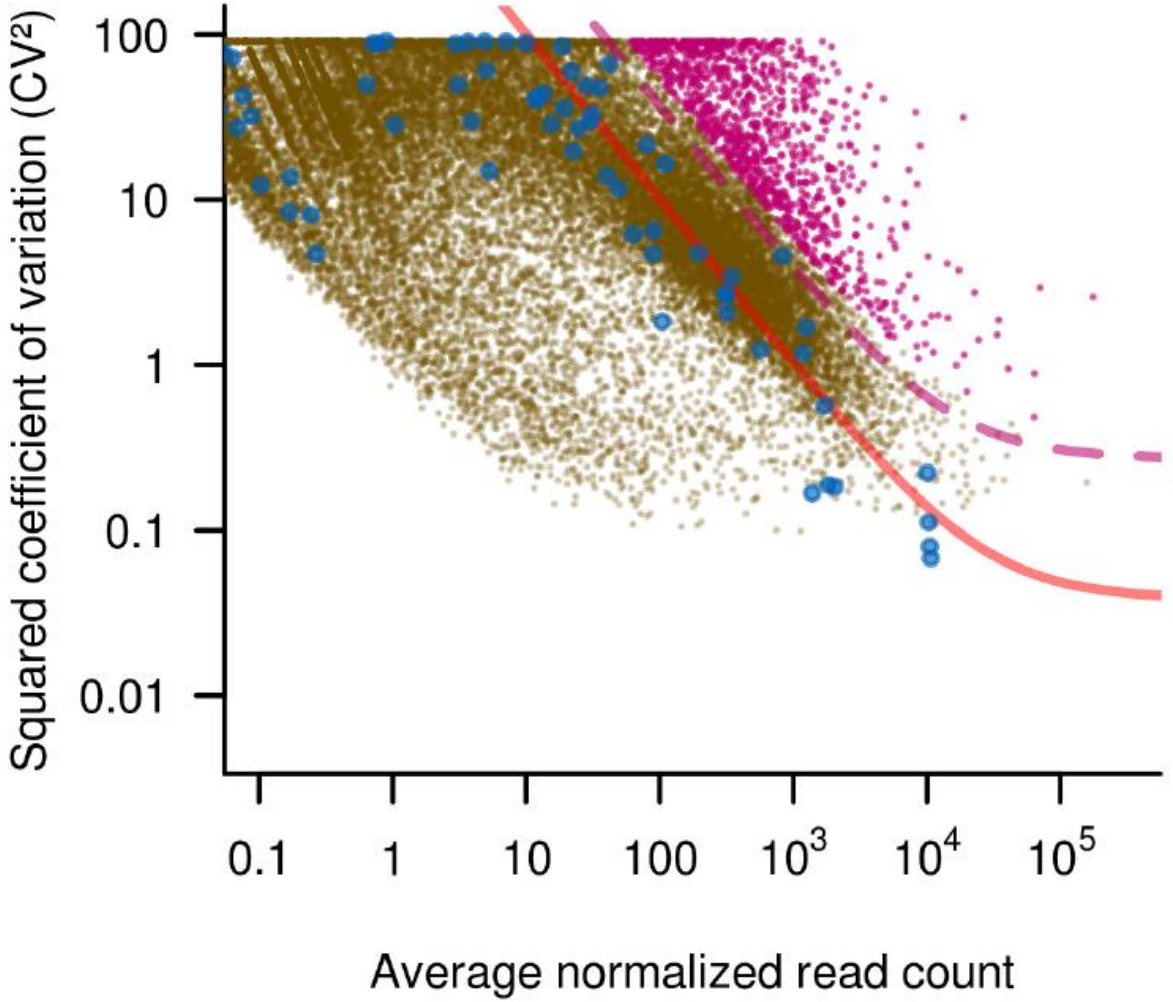
- Imputed expression values can be used to
  - Recover regulatory interactions between genes
  - Increase the accuracy of estimates of cell-to-cell variability in gene expression
  - Improve cell clustering and differential gene expression analysis.

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 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

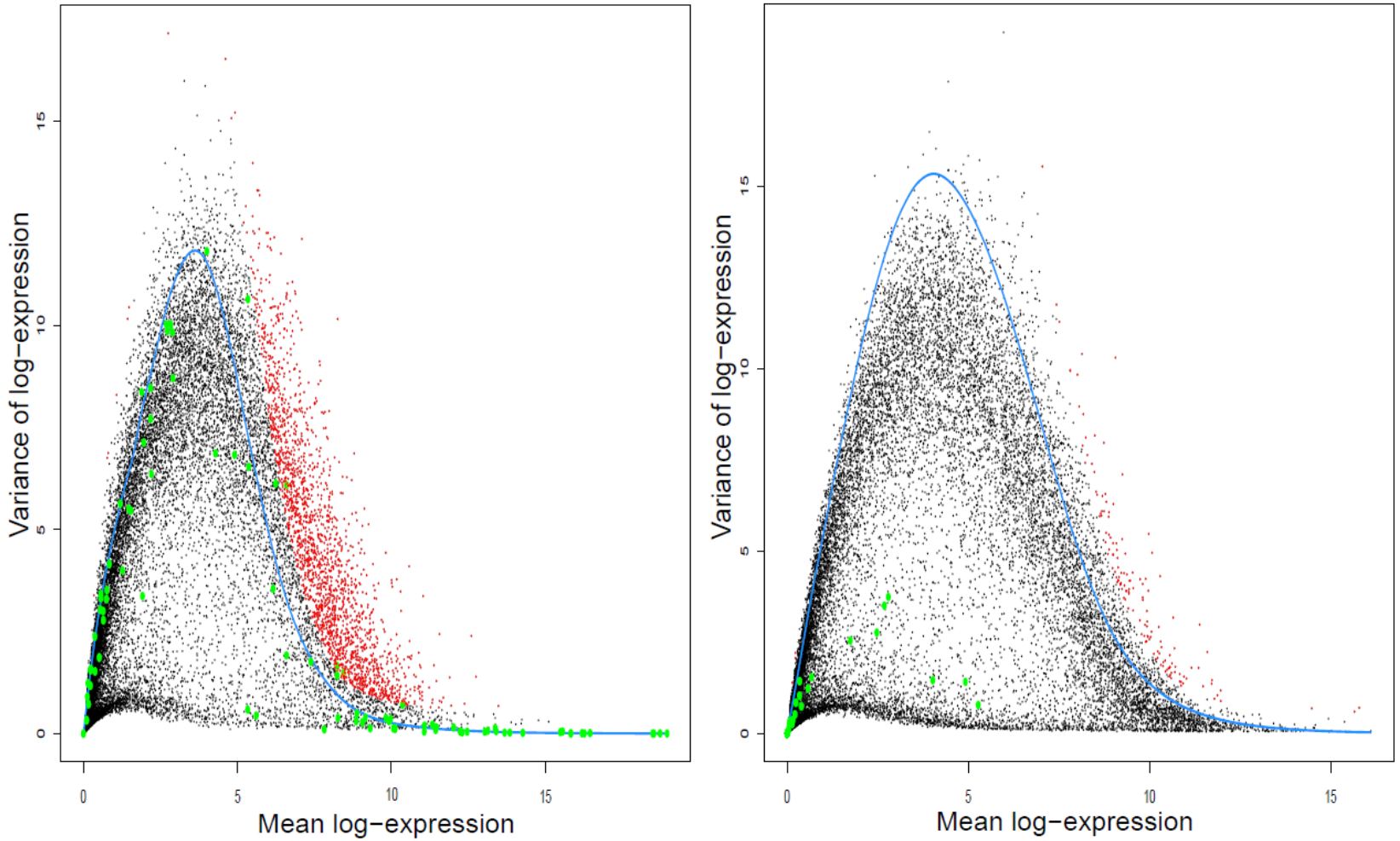
- If spike-ins have poor quality or are not available, we directly estimate the mean-variance trend of technical noise from endogenous genes by assuming that most genes are dominated by technical variability.

- 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.

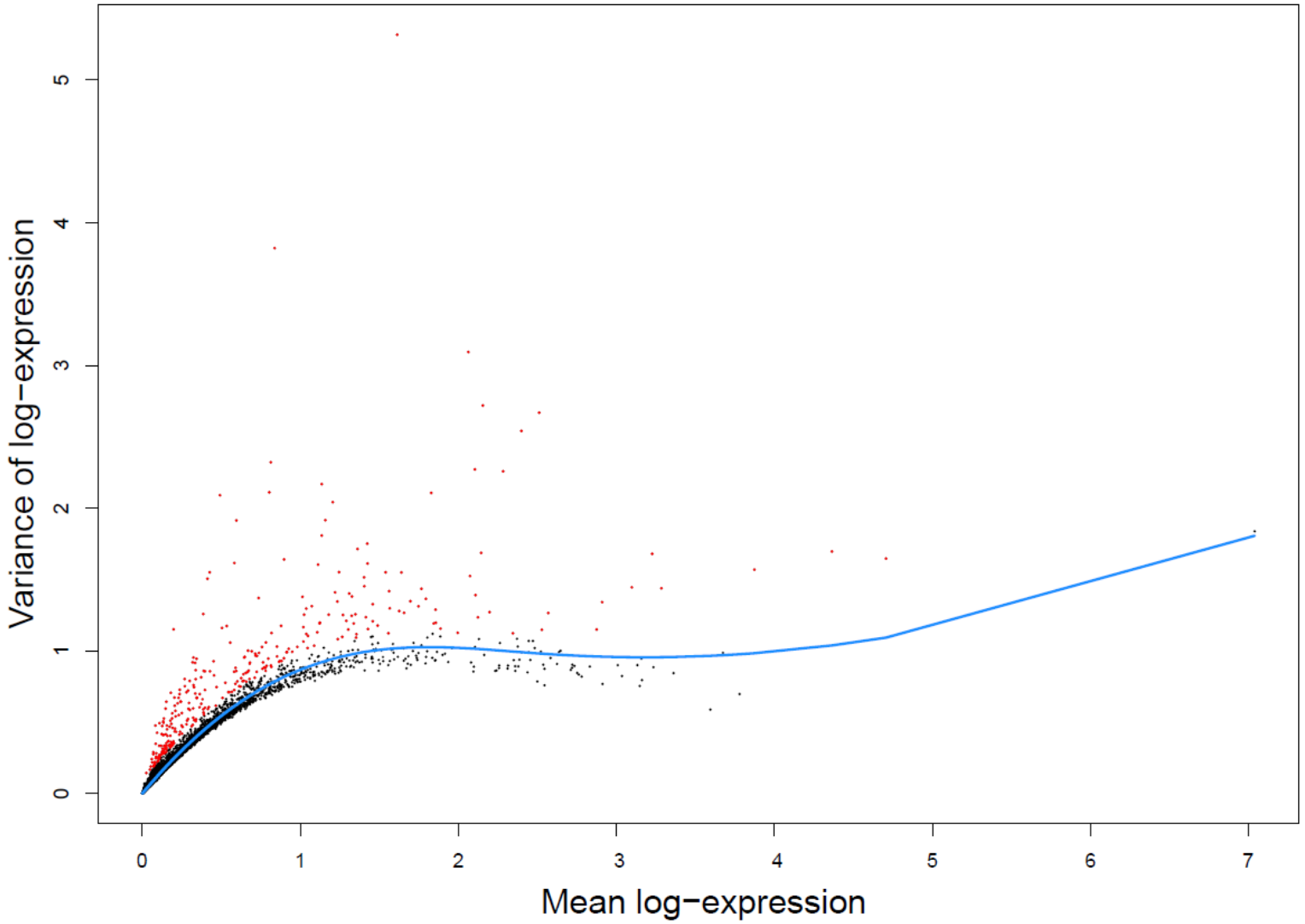
- The test statistic under the null hypothesis follows a chi-squared distribution if we assume that the normalized counts follow a normal distribution.

- 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

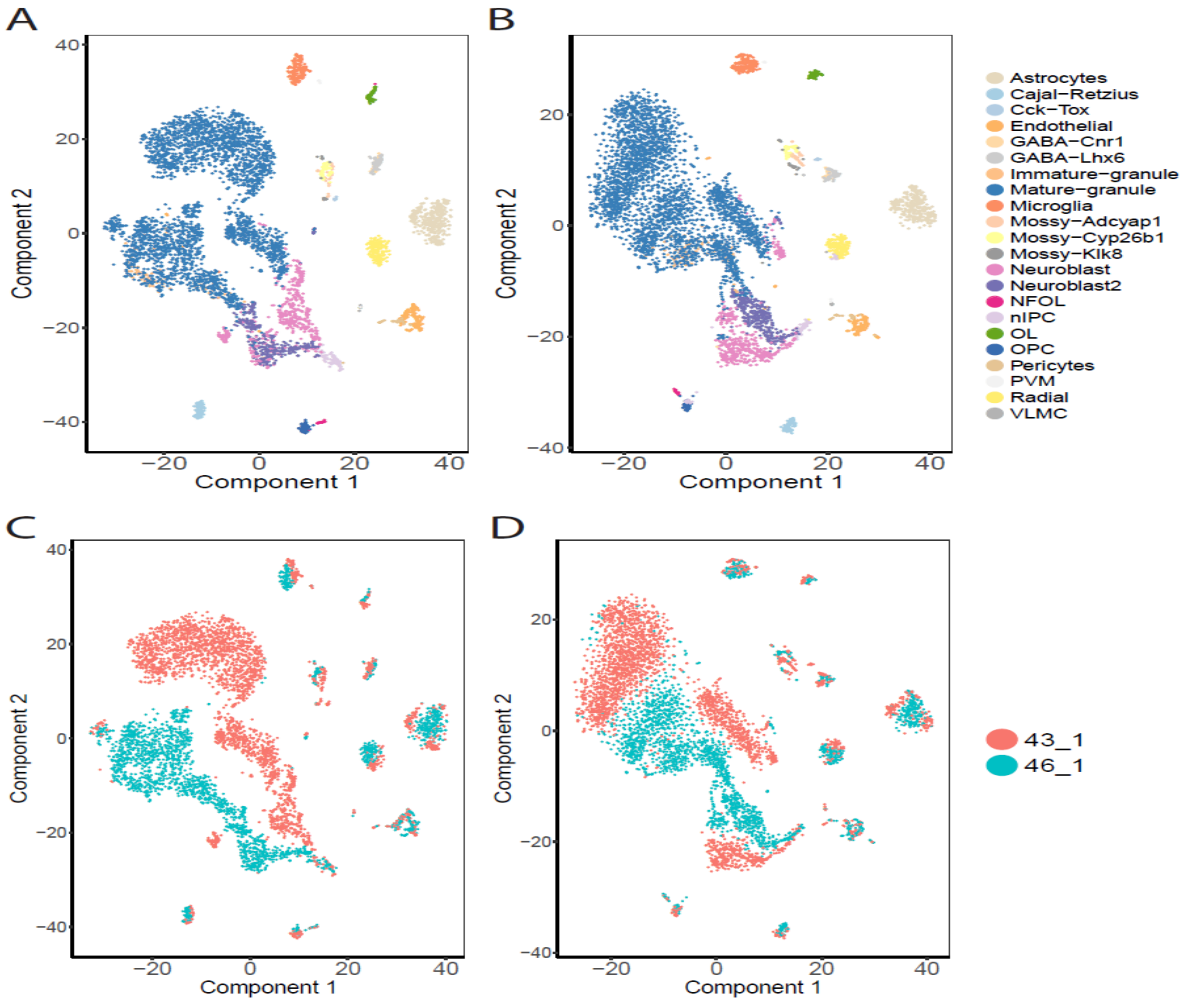


Mean-variance plot



Visualization with HVGs

All genes HVGs



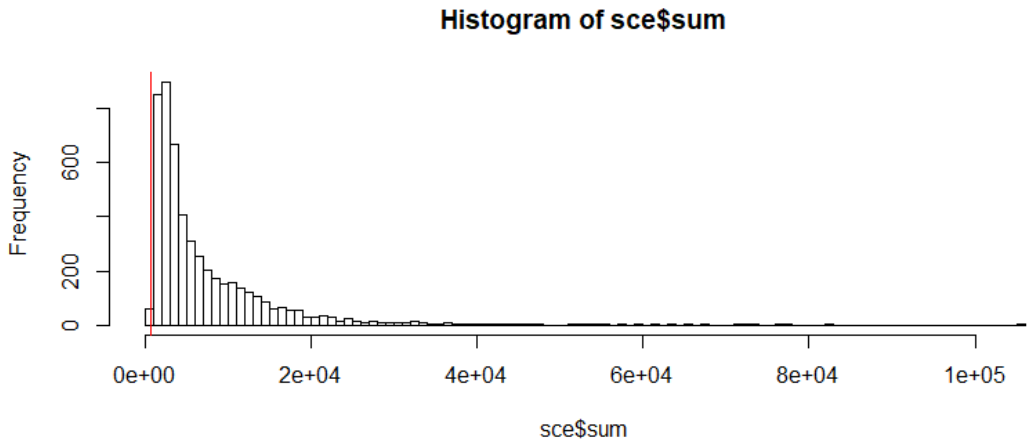
Load Raw Count Data (Dataset 2)

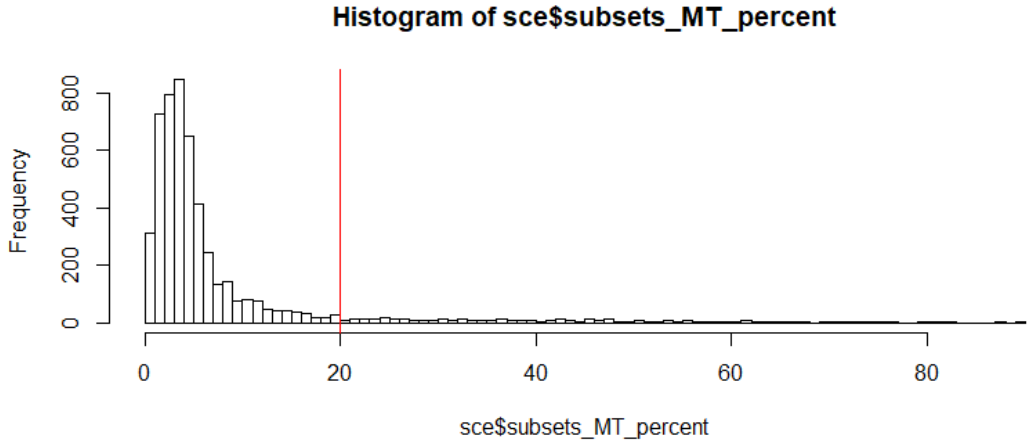
```r
# load sample info
sampleInfo <- read.table(paste0(dir, "samples.txt"), header=TRUE, sep="\t")
head(sampleInfo)

sampleInfo$ID <- sapply(sampleInfo$Sample %>% as.character(),
                        function(x) {strsplit(x, split="_")[[1]][3]})
sampleInfo$ID <- gsub("LCP", "", sampleInfo$ID)

orig.ids = sampleInfo$ID %>% as.factor() %>% levels()
new.ids = c("H18", "H21", "H23", "C25", "C26", "H28",
            "C29", "H30", "C35", "H37", "H38", "C39")
for(i in orig.ids){
  new.id = new.ids[grep(i, new.ids)]
  sampleInfo[grep(i, sampleInfo$ID),]$ID = new.id
}
sampleInfo$Diagnosis = sampleInfo$ID
for(i in c("H", "C")){
  if(i == "H"){
    sampleInfo[grep(i, sampleInfo$ID),]$Diagnosis = "HCC"
  }else{
    sampleInfo[grep(i, sampleInfo$ID),]$Diagnosis = "iCCA"
  }
}
```

```r
#load scRNAseq data
sce <- read10xCounts(
  samples = dir,
  type="sparse",
  col.names = TRUE
)
counts(sce) = counts(sce) %>% as.matrix()
rownames(sce) = uniquifyFeatureNames(rowData(sce)$ID, rowData(sce)$Symbol)

sce$Sample = sampleInfo$Sample
sce$Diagnisis = sampleInfo$Diagnosis
sce$ID = sampleInfo$ID
sce$Type = sampleInfo$Type
```

Cell Quality Control (Dataset 2)

```r
# QC
library(scater)
mtgenes = rowData(sce)[grep("MT-", rowData(sce)$Symbol),]$Symbol
is.mito = rownames(sce) %in% mtgenes
table(is.mito)
sce <- addPerCellQC(
  sce,
  subsets = list(MT=mtgenes),
  percent_top = c(50, 100, 200, 500),
  detection.limit = 5
)

sce$log10_sum = log10(sce$sum + 1)
sce$log10_detected = log10(sce$detected + 1)

umi = 700
mtpct = 20

hist(sce$sum, breaks = 100)
abline(v = umi, col="red")

hist(sce$subsets_MT_percent, breaks=100)
abline(v=mtpct, col="red")

filter_by_total_counts = sce$sum > umi
filter_by_mt_percent = sce$subsets_MT_percent < mtpct

sce$use <- (
  filter_by_total_counts &
    filter_by_mt_percent
)
sce = sce[,sce$use]
```

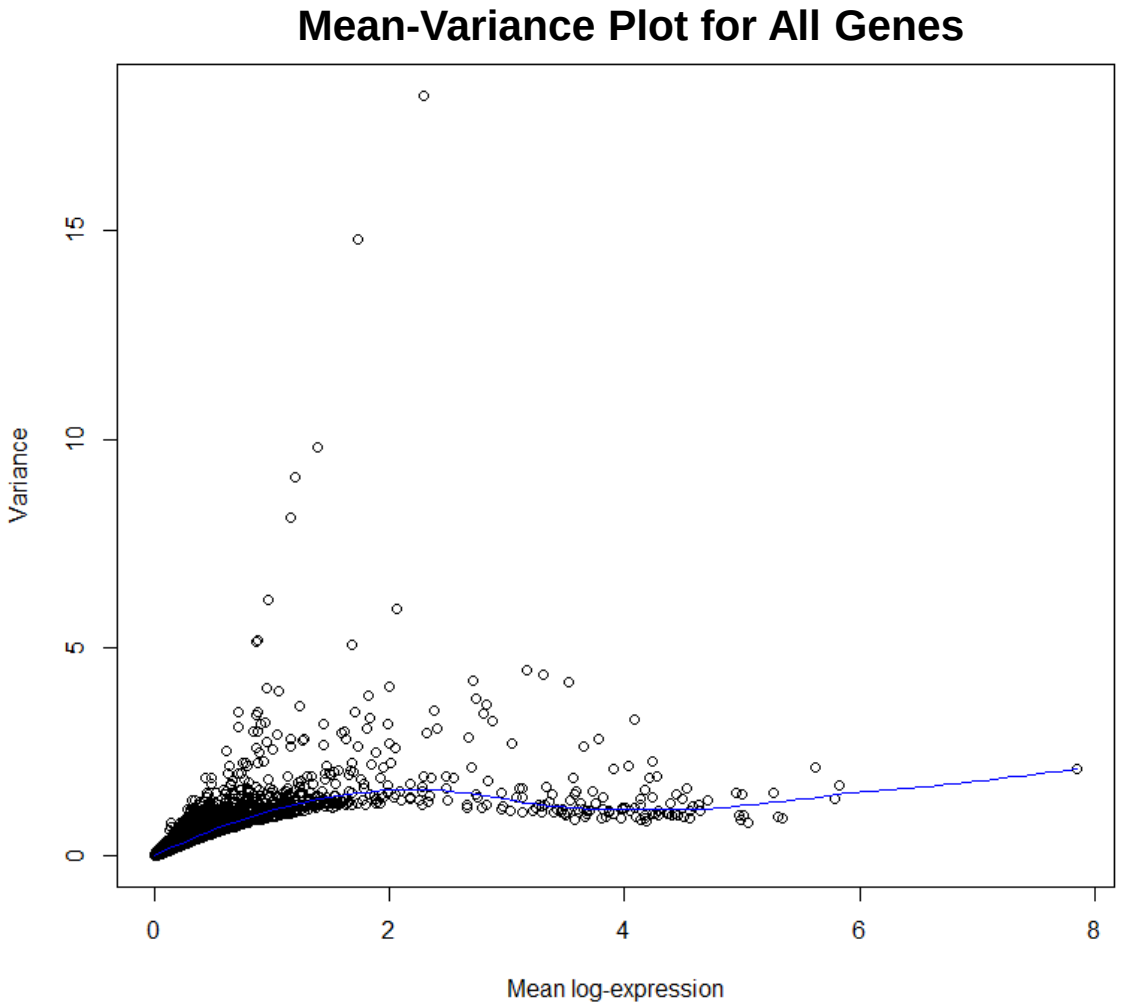




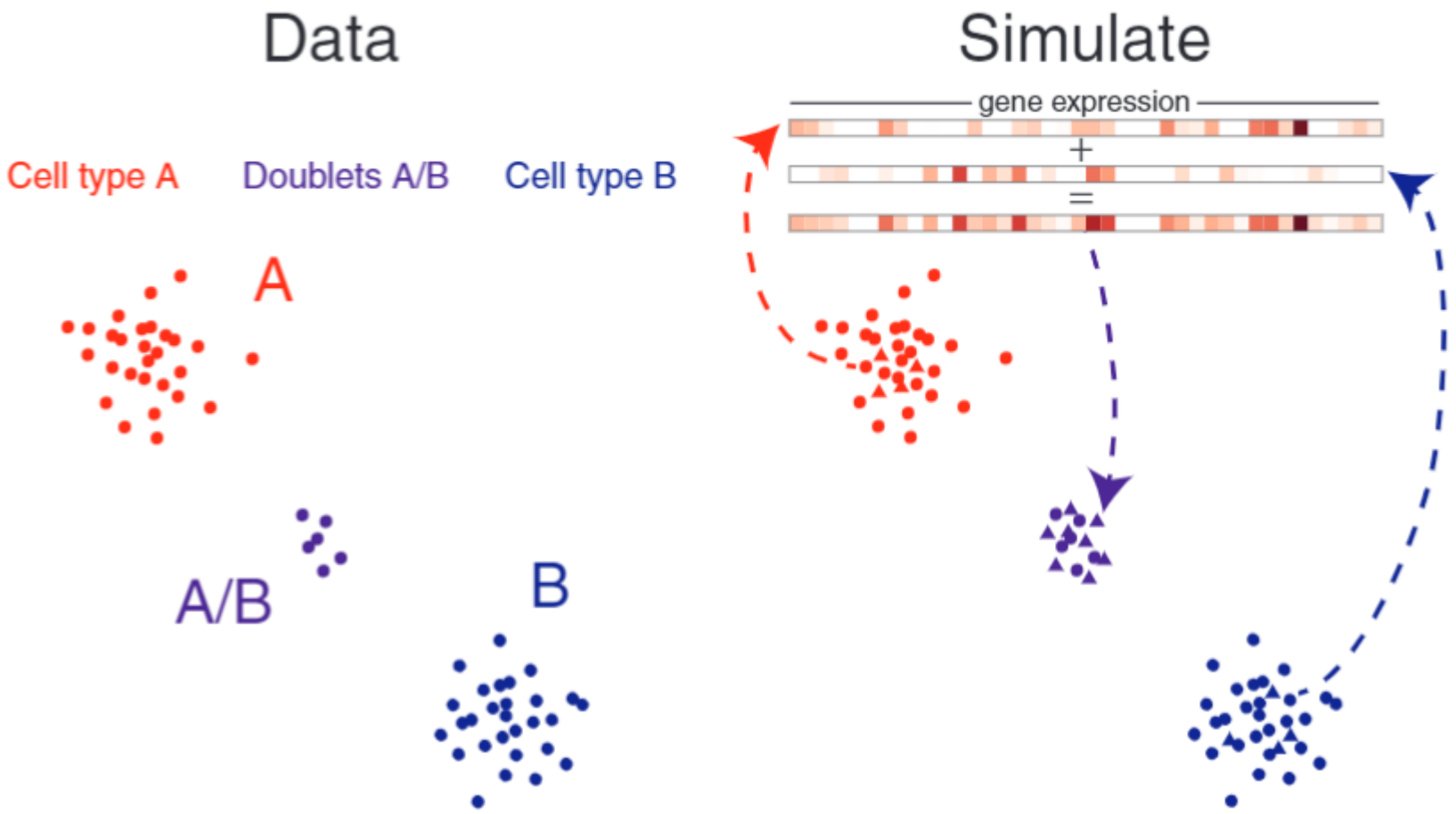
Normalization/HVG Selection (Dataset2)

```r
# normalization
library(scran)
clusters <- quickCluster(sce)
sce <- computeSumFactors(sce, clusters = clusters)
print(summary(sizeFactors(sce)))

sce.norm <- logNormCounts(sce)
```

```r
# highly variable genes
dec <- modelGeneVar(sce.norm)
plot(dec$mean, dec$total,
     xlab="Mean log-expression", ylab="Variance")
curve(metadata(dec)$trend(x), col="blue", add=TRUE)

hvg.norm <- getTopHVGs(dec, fdr.threshold = 0.05)
length(hvg.norm) # 551 genes
```



Doublets identification

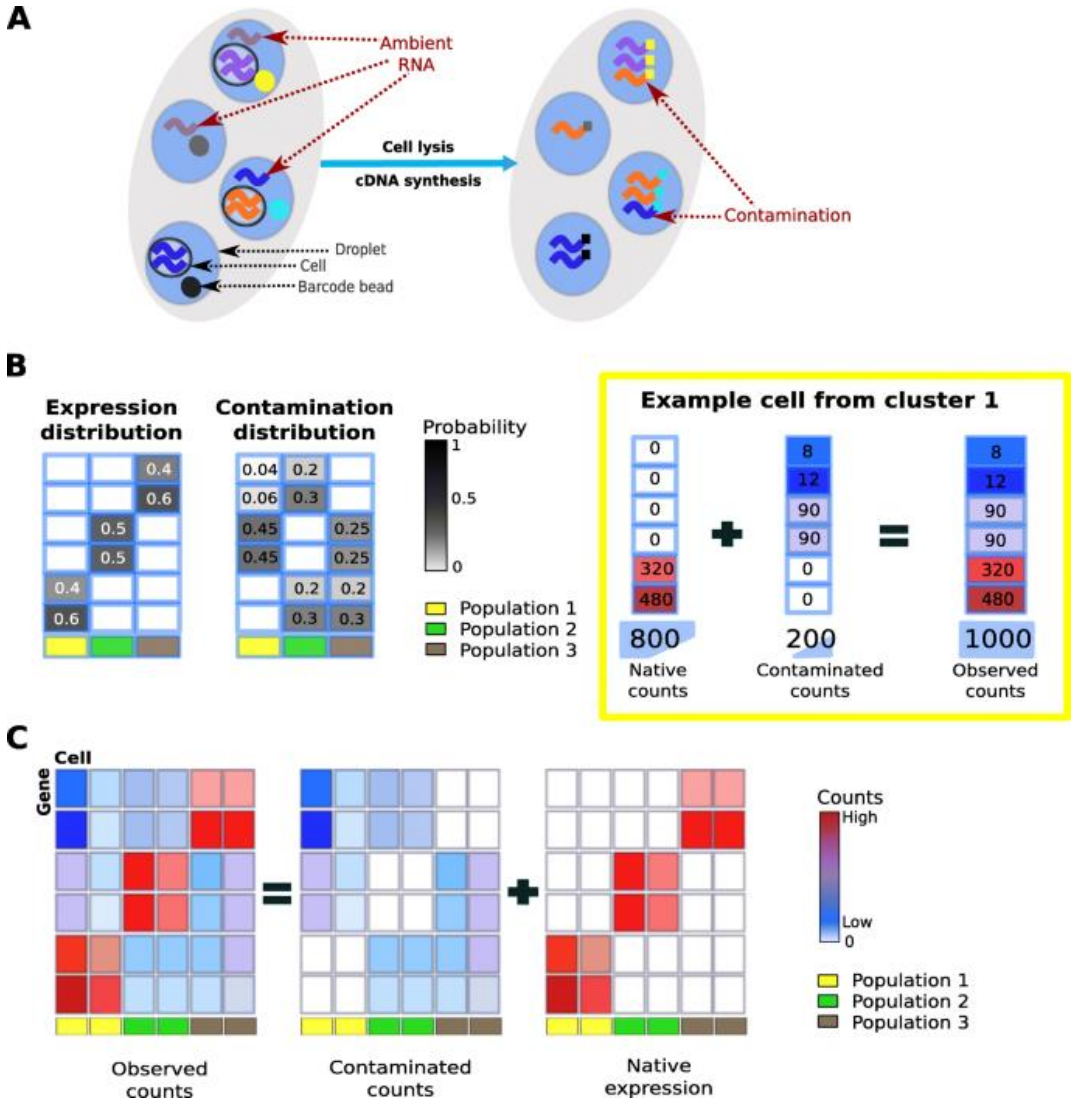


Cell Systems (2020) 11:95

Doublets identification

- Scrublet: https://github.com/AllonKleinLab/scrublet
- DoubletFinder: https://github.com/chris-mcginnis-ucsf/DoubletFinder
- Solo: https://github.com/calico/solo

Ambient RNA contamination



Genome Biol (2020) 21:57

Ambient RNA contamination

- DecontX: https://github.com/campbio/celda
- SoupX: https://github.com/constantAmateur/SoupX

Single cell RNA-seq

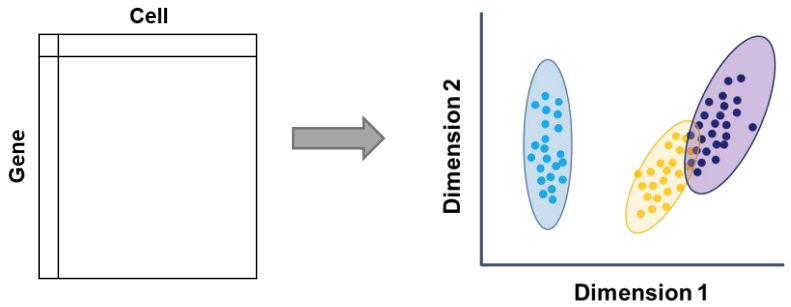
Exploratory analysis

Jong Kyoung Kim

Overview

Exploratory Analysis

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



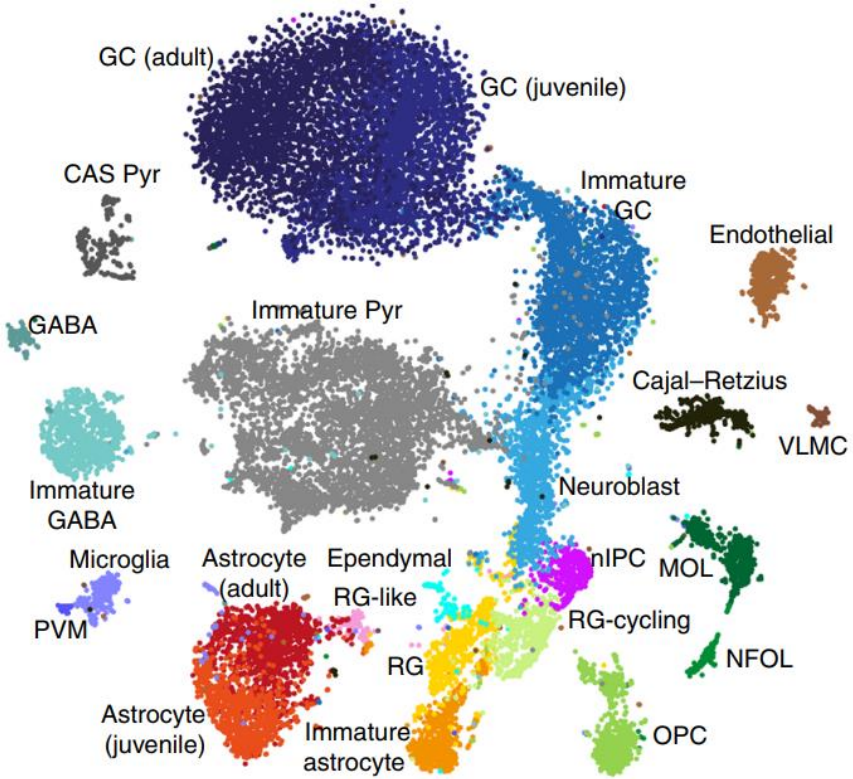
Dimensionality reduction

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.

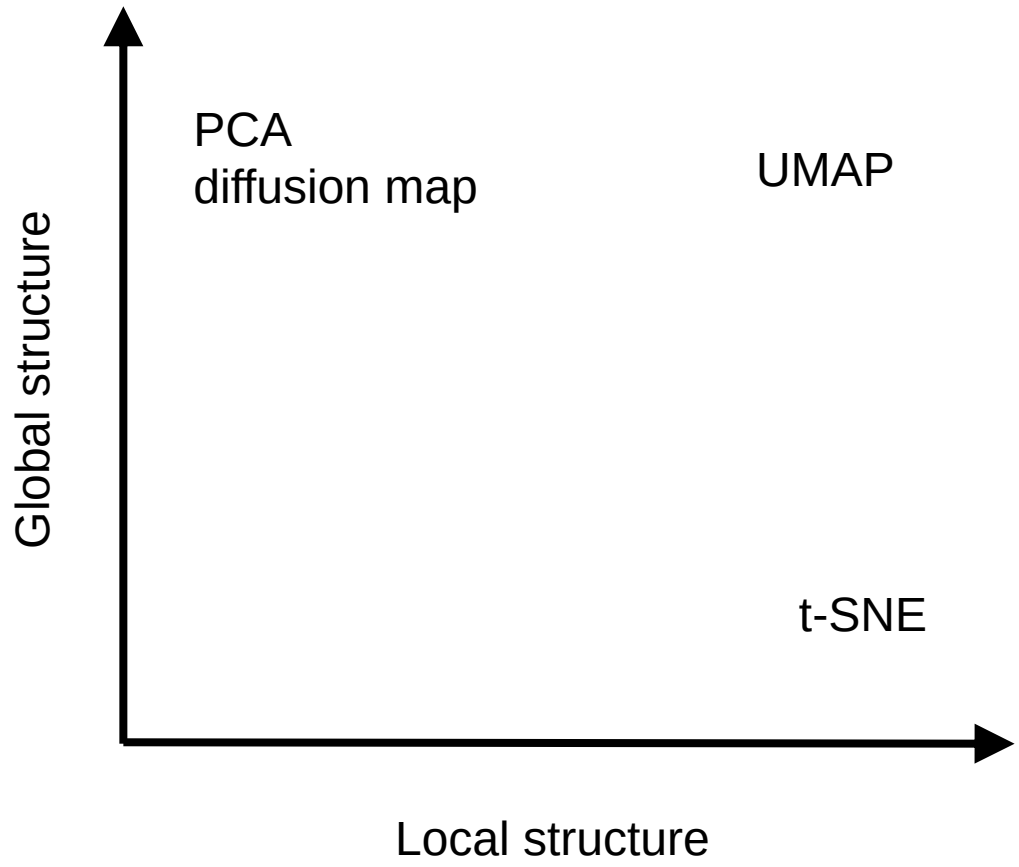
- Two different relevant structures are considered
  - A local structure that preserves cell-to-cell distance within a local neighborhood of cells
  - A global structure that preserves cell-to-cell distance on the low-dimensional manifold associated with the underlying biological process

Local and global structure

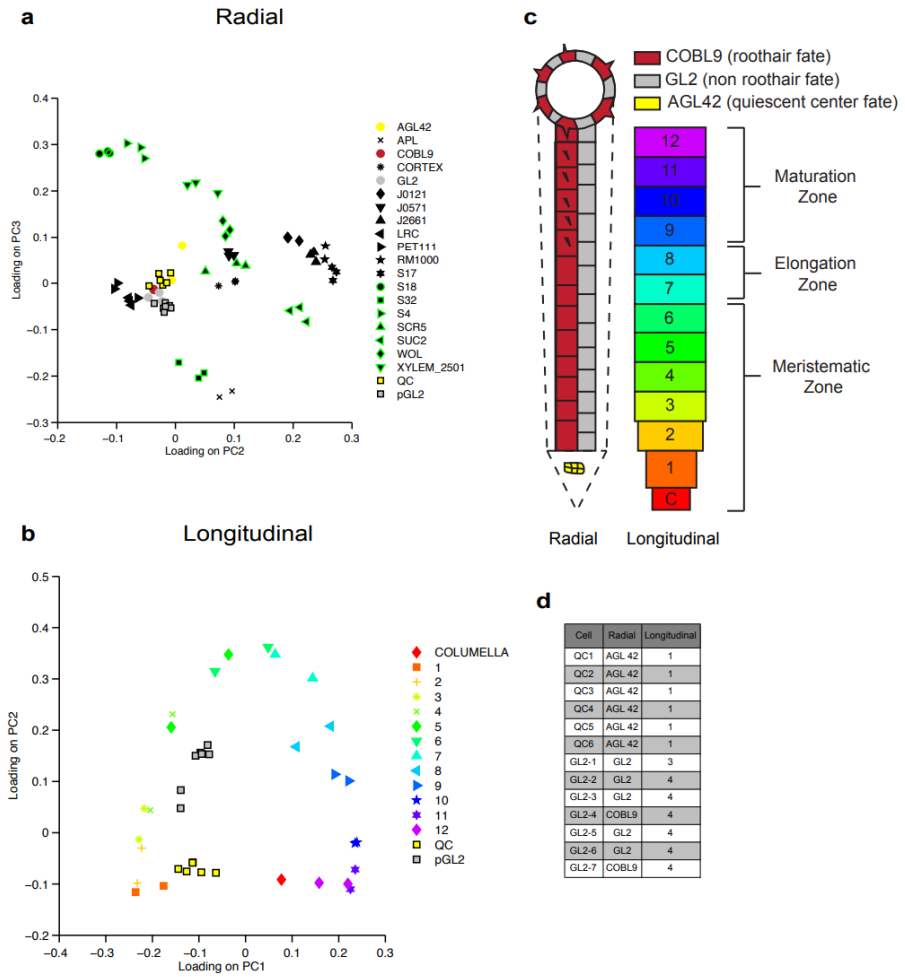
- Capturing local structure in a low-dimensional representation is important for clustering cells of the same type or state close together.

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

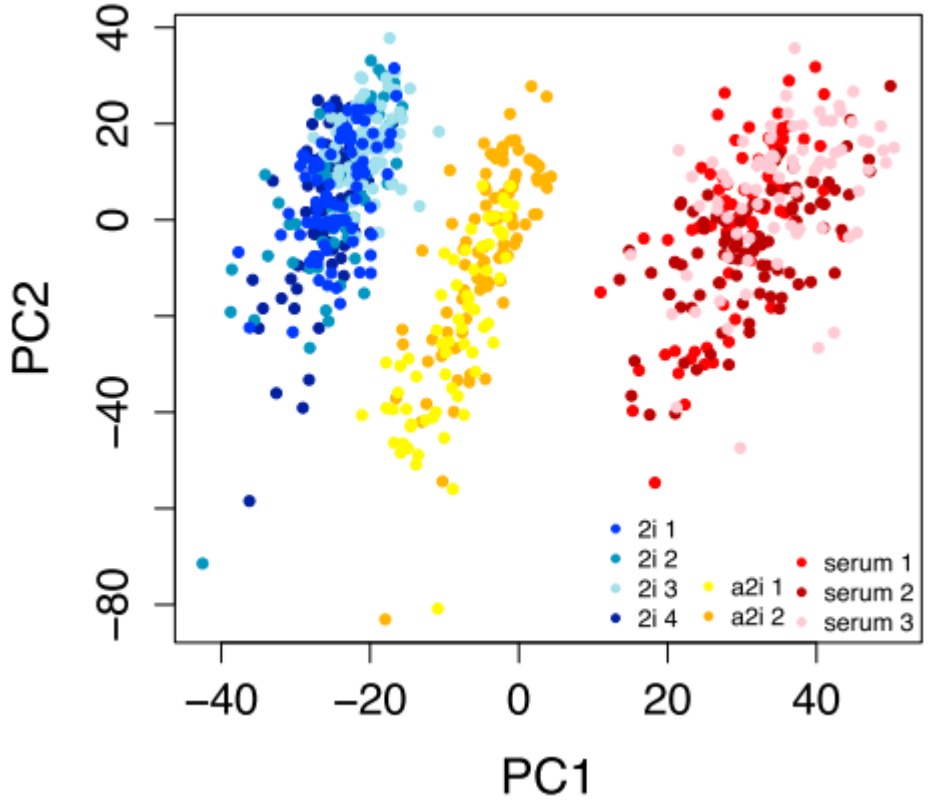
Algorithms


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 (2015)* ***Cell Stem Cell***

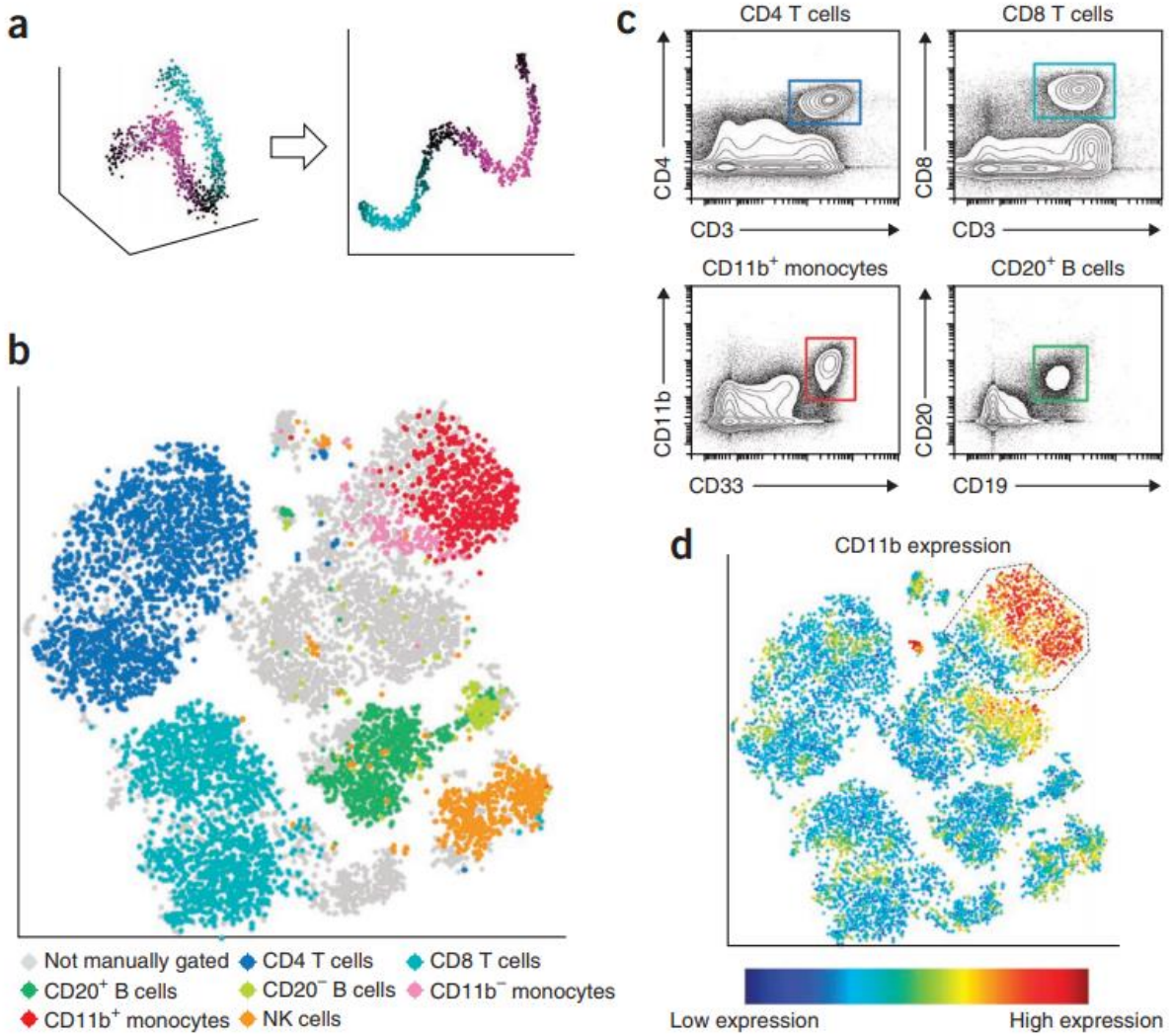
Limitations of PCA

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.

t-SNE for single-cell data

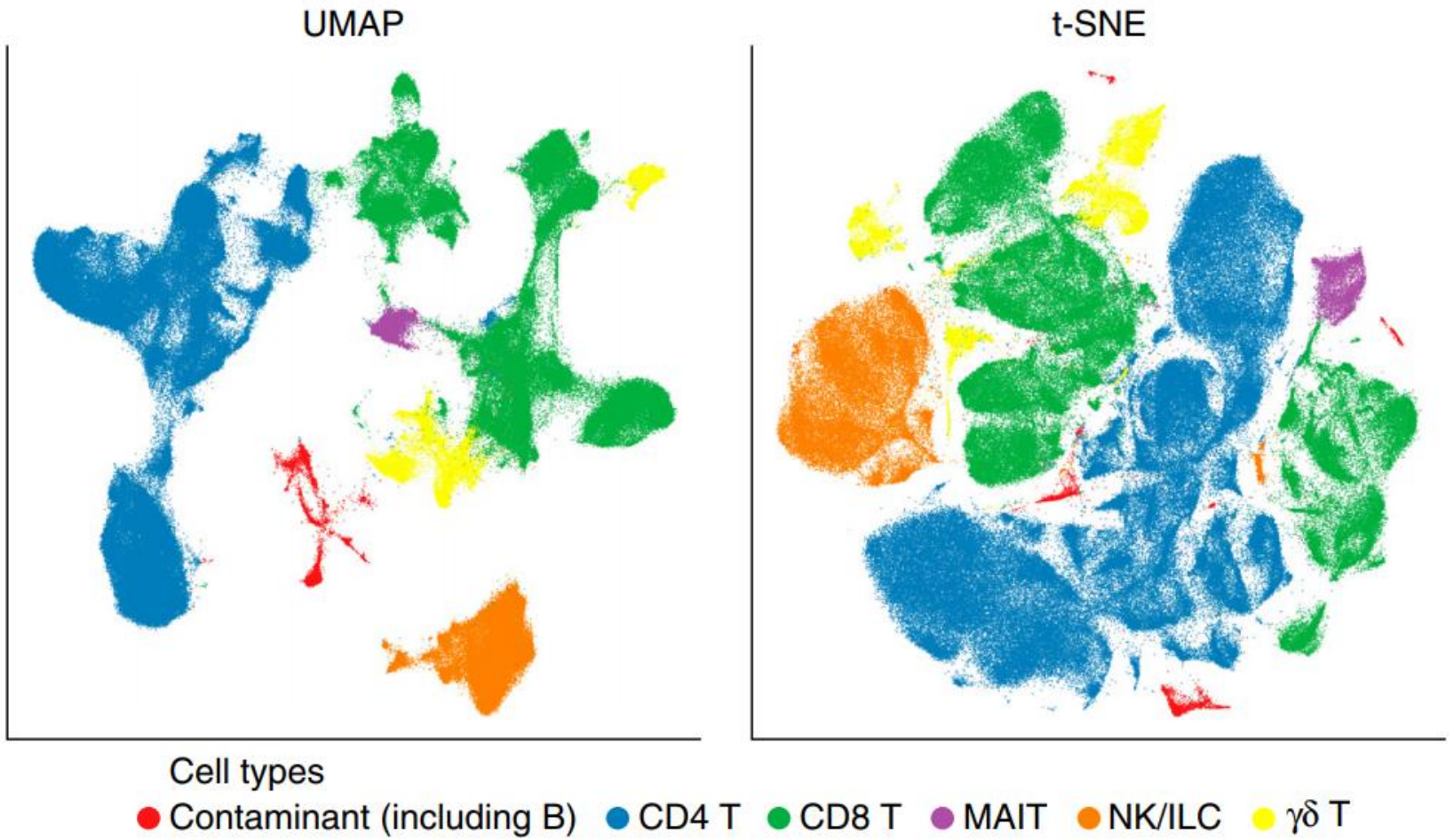


Nat Biotechnol (2013) 31:545

t-SNE for single-cell data

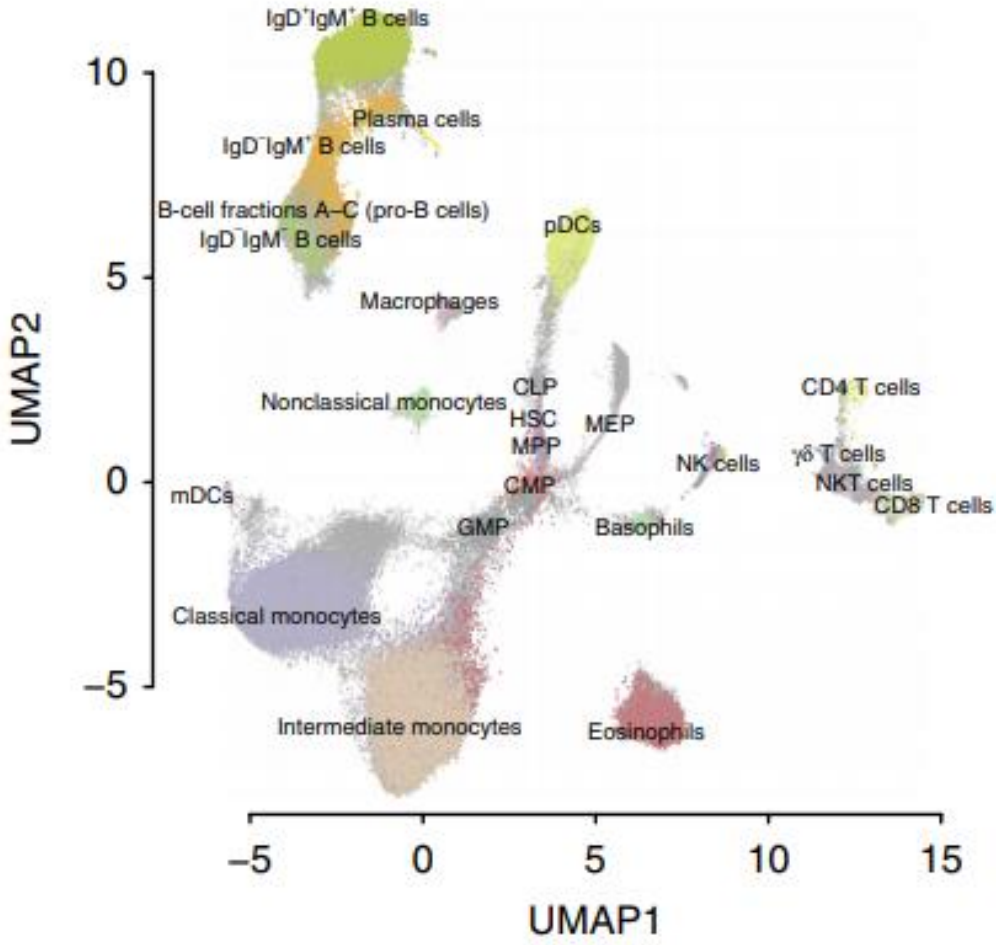
- “The art of using t-SNE for single-cell transcriptomics” Nature Commun 10:5416 (2019)

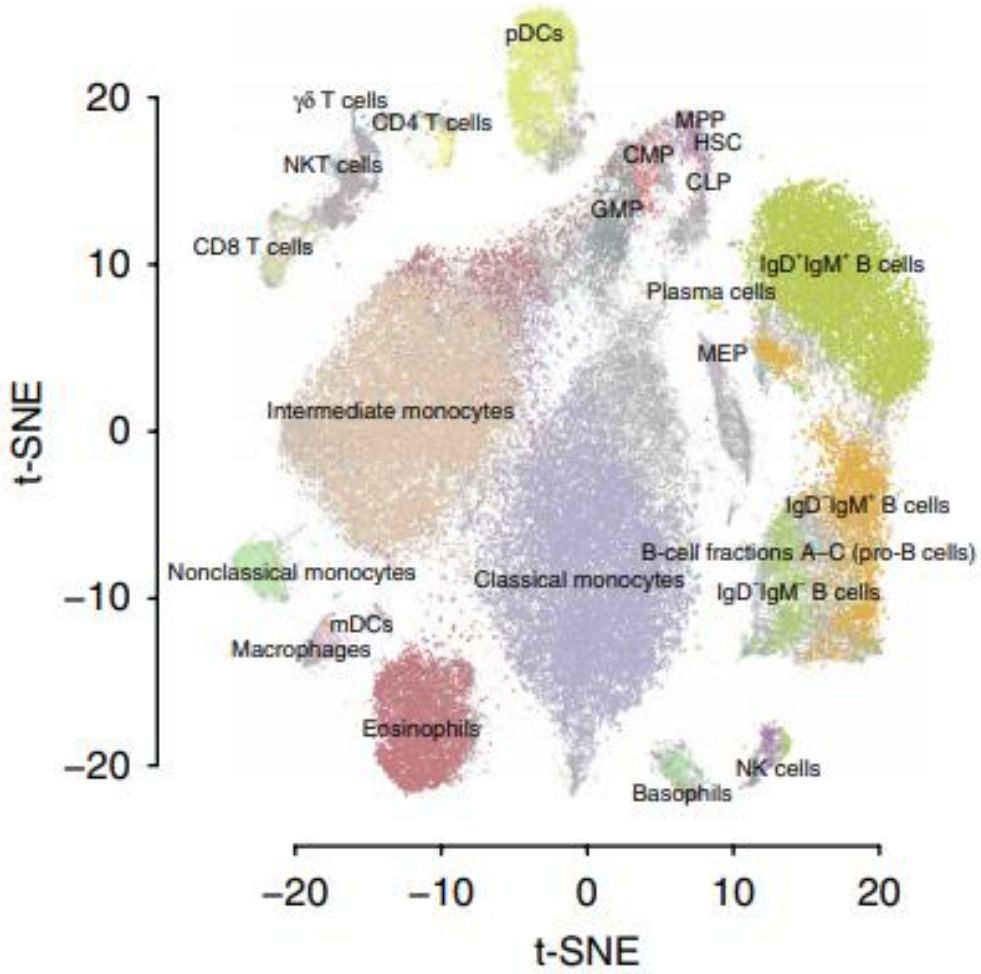
UMAP

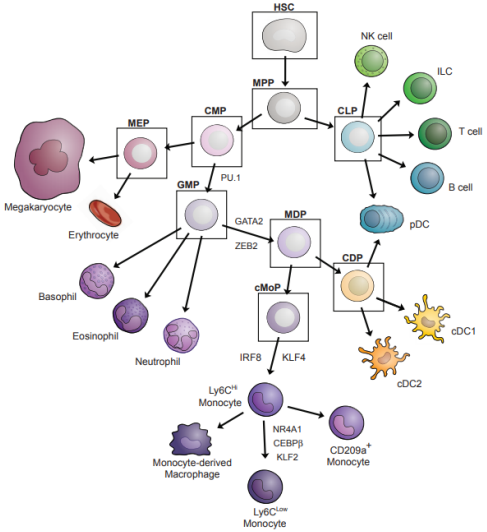


Nat Biotechnol (2019) 37:38

UMAP







Nat Biotechnol (2019) 37:38

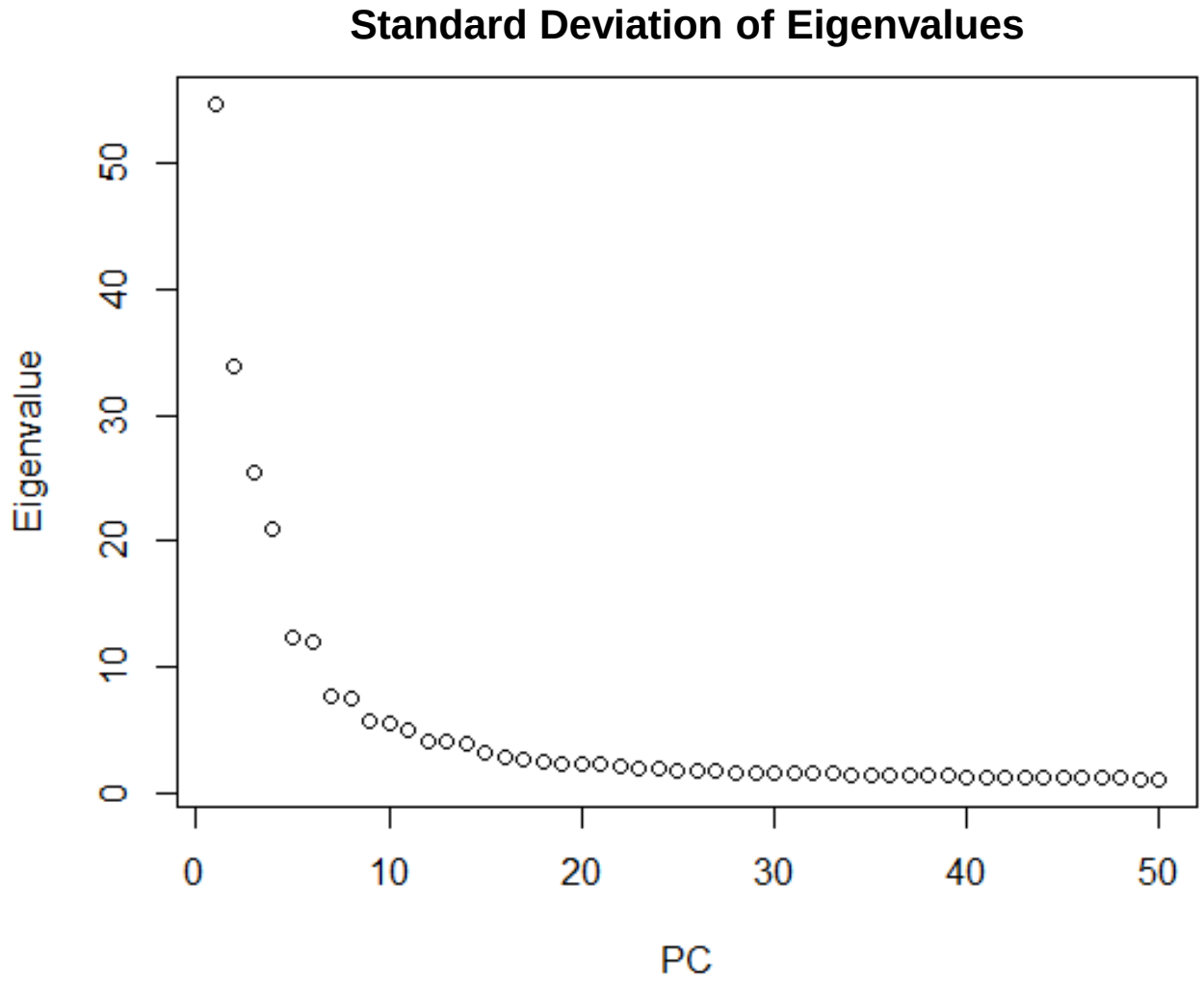
Visualization: Comparison

- “Tuning parameters of dimensionality reduction methods for single-cell RNA-seq analysis” Genome Biology 21:212 (2020)

- “Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis” Genome Biology 20:269 (2020)

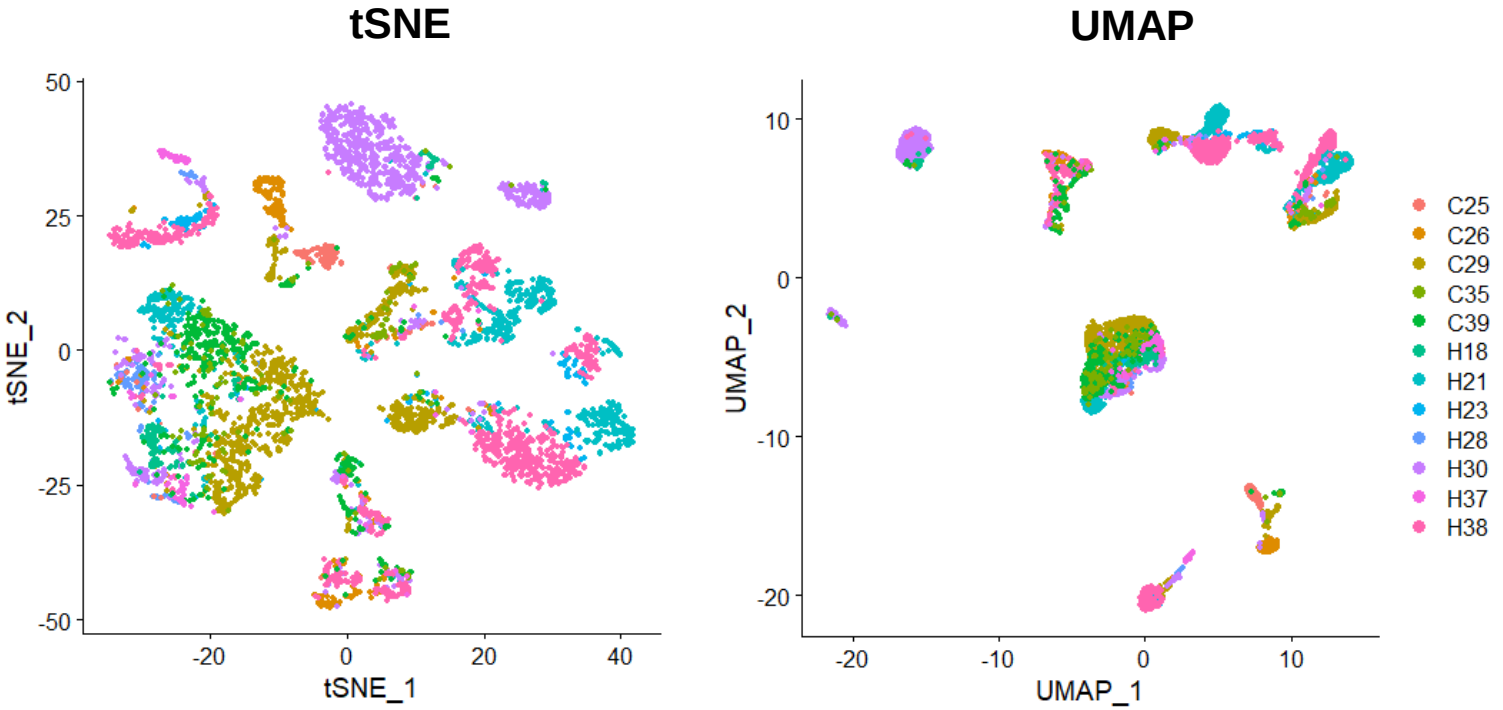
Dimension Reduction (Dataset 2)

```r
# dimension reduction
library(Seurat)
seurat <- as.Seurat(sce.norm,
                    counts = "counts",
                    data = "logcounts",
                    assay = "RNA")
VariableFeatures(seurat) = hvg.norm

all.genes = rownames(seurat)
seurat <- ScaleData(seurat, features = all.genes)

seurat <- RunPCA(seurat,
                 assay = "RNA",
                 npcs = 50,
                 features = hvg.norm,
                 reduction.key = "pca_",
                 verbose = FALSE)
plot((seurat@reductions$pca@stdev)^2,
     xlab = "PC",
     ylab = "Eigenvalue")
```



Dimension Reduction (Dataset 2)

```r
PCA=15 # select the number of PC
seurat <- FindNeighbors(seurat, dims=1:PCA)
seurat <- FindClusters(seurat, resolution = 0.8)

seurat <- RunTSNE(seurat,
                  dims = 1:PCA,
                  check_duplicates = FALSE)
seurat <- RunUMAP(seurat,
                  dims = 1:PCA)

TSNEPlot(seurat, group.by = "ID", pt.size = 1)
UMAPPlot(seurat, group.by = "ID", pt.size = 1)
```



Single cell RNA-seq

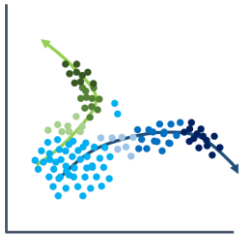
Heterogeneity analysis

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: 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.

Clustering: cell type markers

- https://www.labome.com/method/Cell-Markers.html
- SHOGoiN: https://stemcellinformatics.org/cell_marker/literatures
- PanglaoDB: https://panglaodb.se/

Clustering: cell type classifiers

- Garnett: https://cole-trapnell-lab.github.io/garnett/
- MARS: https://github.com/snap-stanford/mars
- Cell BLAST: https://cblast.gao-lab.org/
- scClassify: https://github.com/SydneyBioX/scClassify
- SingleR: https://bioconductor.org/packages/release/bioc/html/SingleR.html

Clustering: challenges

- “Challenges in unsupervised clustering of single-cell RNA-seq d ata” Nature Reviews Genetics 20:273 (2019)

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.

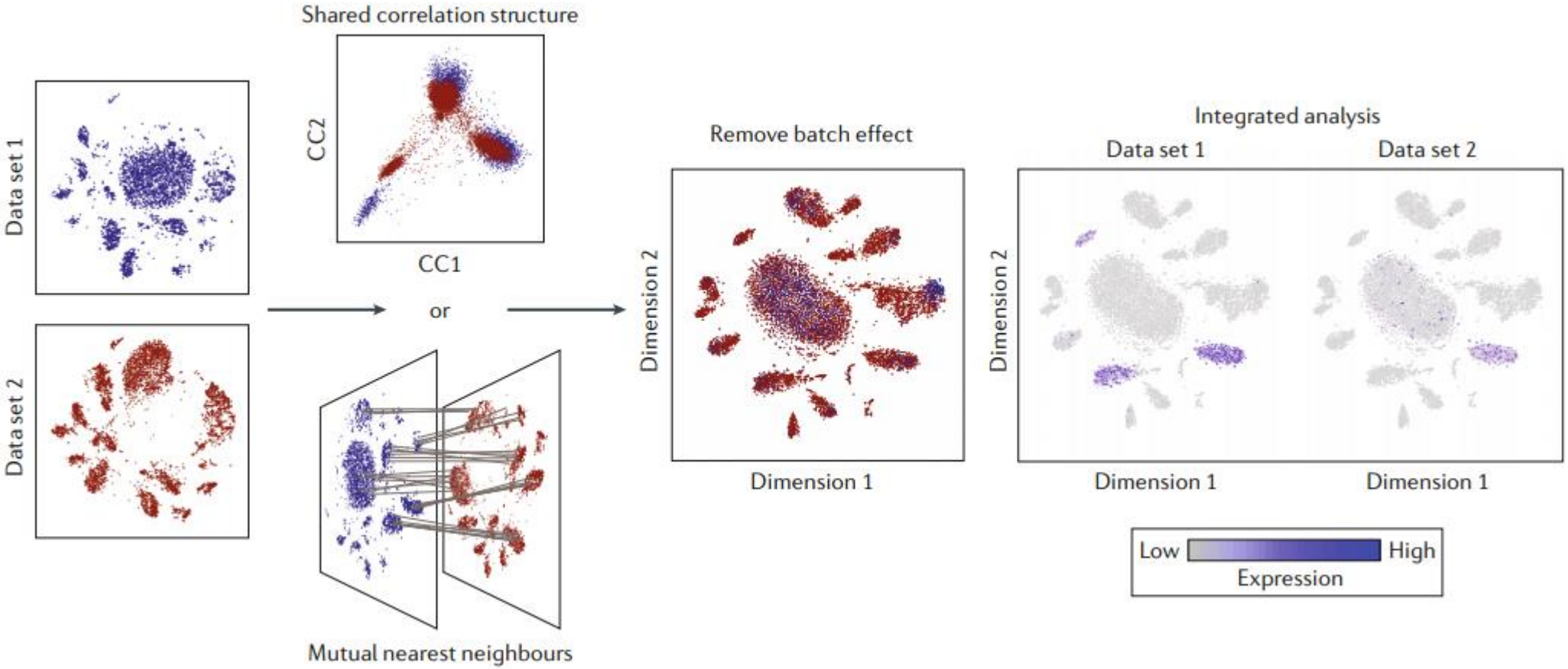
Batch correction

- 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.

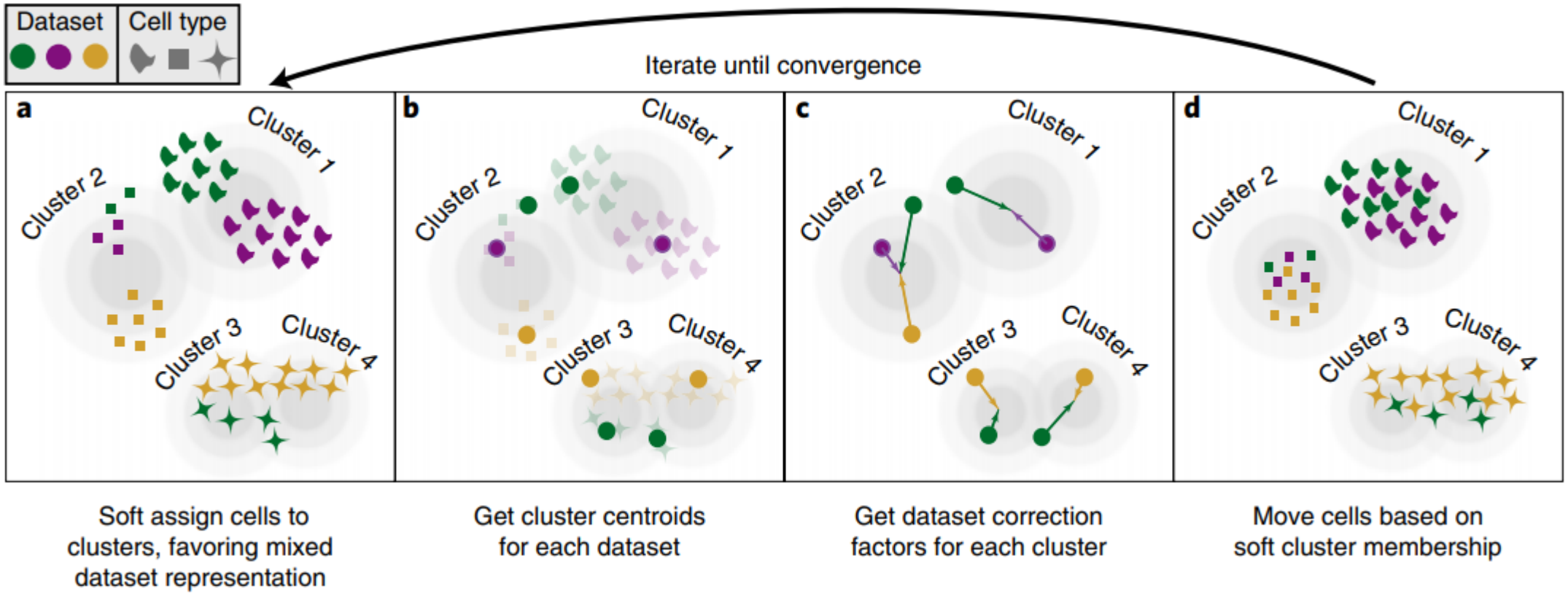
- A more general strategy is to merge multiple scRNA-seq data with shared subpopulations using canonical correlation analysis or by identifying mutual nearest neighbors.

Batch correction: CCA and mNN



Nat Rev Genet (2019) 20:257

Batch correction: Harmony



Nat Methods (2019) 16:1289

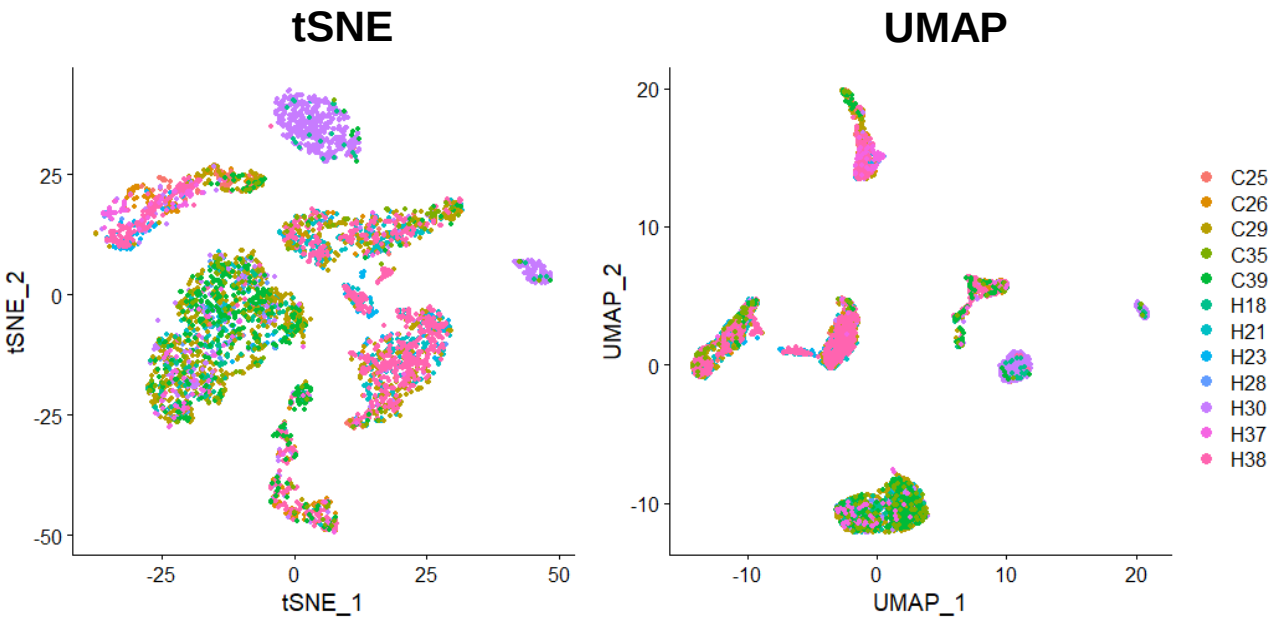
Batch correction: Comparison

- “A benchmark of batch-effect correction methods for single-cell RNA sequencing data” Genome Biology 21:12 (2020)

- “Benchmarking atlas-level data integration in single-cell genomics” bioRxiv

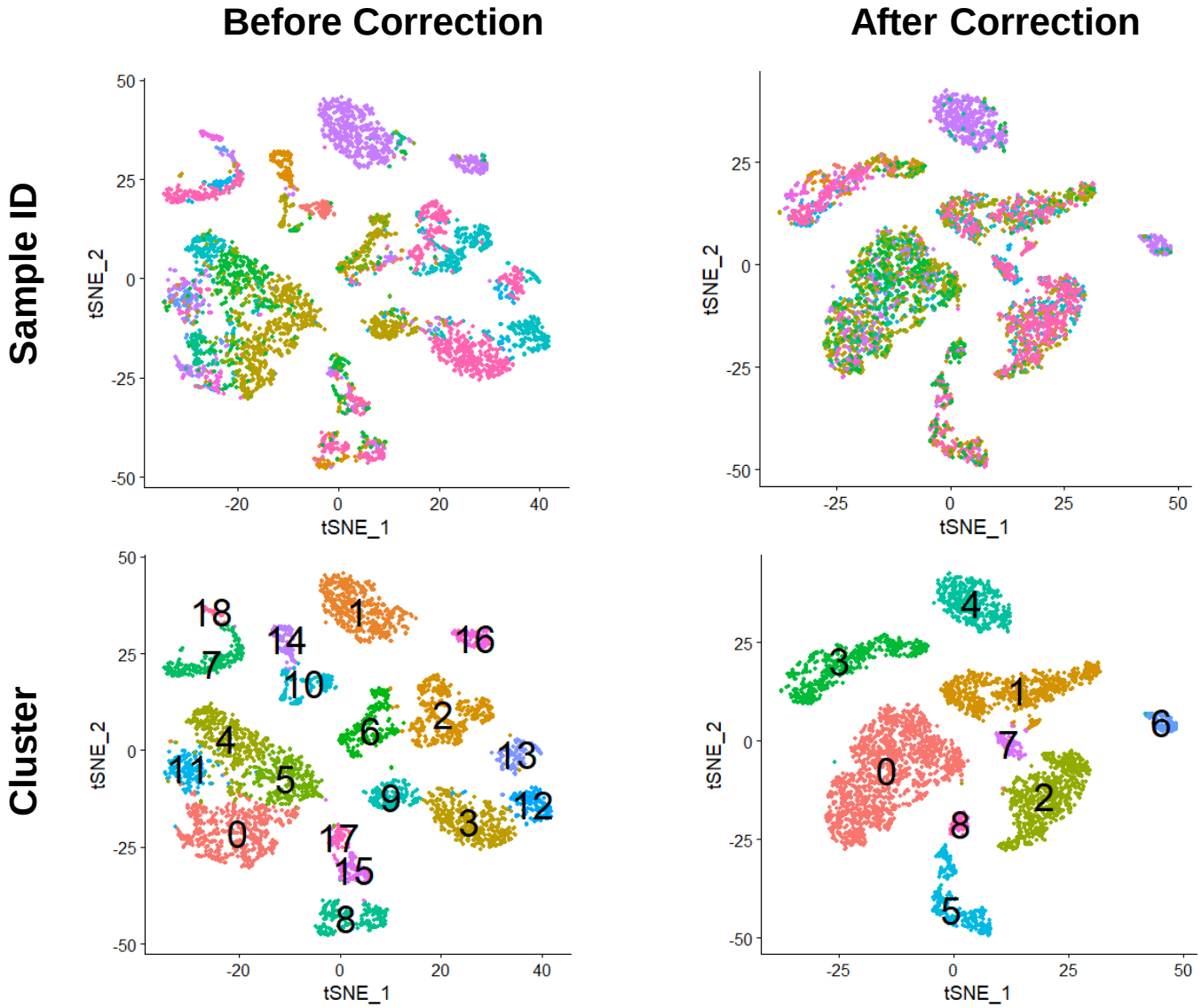
Batch correction with Harmony (Dataset 2)

```
# batch correction
library(harmony)
seurat <- RunHarmony(seurat, "ID", plot_convergence = TRUE)

nComp = 15
seurat <- FindNeighbors(seurat,
                        reduction = "harmony",
                        dims=1:nComp)
seurat <- FindClusters(seurat, resolution = 0.2)

seurat <- RunTSNE(seurat,
                  reduction = "harmony",
                  dims = 1:nComp,
                  check_duplicates = FALSE)
seurat <- RunUMAP(seurat,
                  reduction = "harmony",
                  dims = 1:nComp)

# plot by Sample ID
DimPlot(seurat, reduction = "umap", group.by = "ID", pt.size = 1)
DimPlot(seurat, reduction = "tsne", group.by = "ID", pt.size = 1)
```



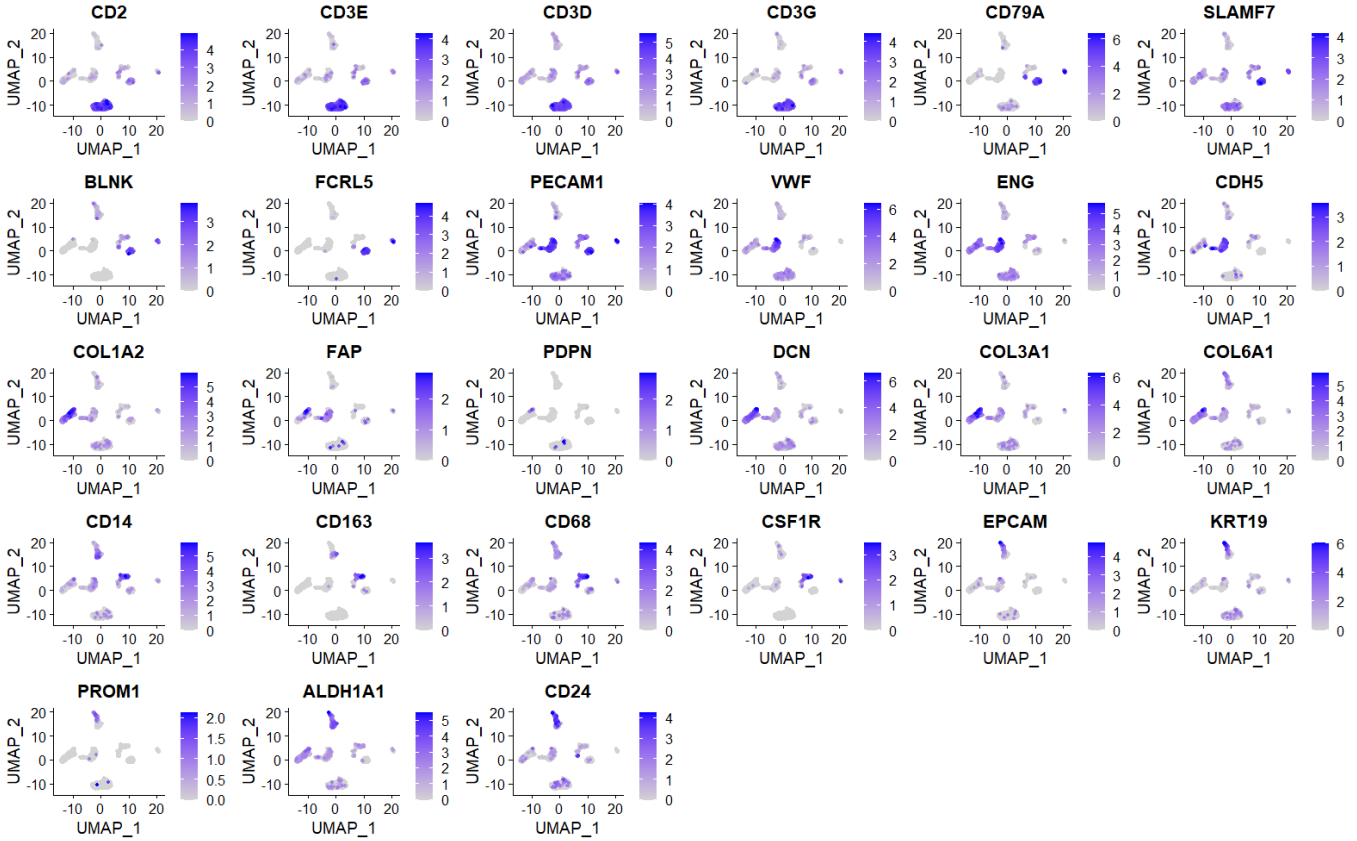
Batch Correction

```
# plot by Clusters
DimPlot(seurat, reduction = "tsne",
        group.by = "seurat_clusters",
        pt.size=1, label=TRUE, label.size = 10)
DimPlot(seurat, reduction = “umap",
        group.by = "seurat_clusters",
        pt.size=1, label=TRUE, label.size = 10)
```



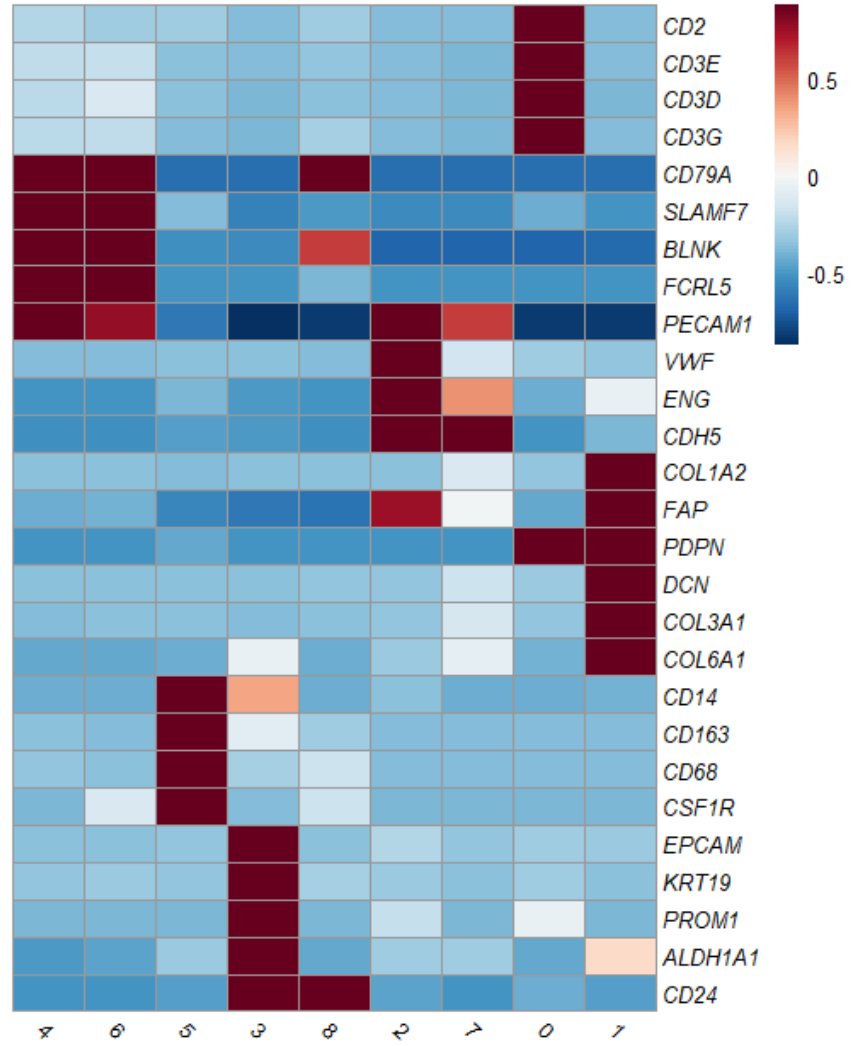
Cell Type Annotation (Dataset 2)

```
markers = list(T.cells = c("CD2", "CD3E",
                           "CD3D", "CD3G"), #cluster 0
               B.cells = c("CD79A", "SLAMF7",
                           "BLNK", "FCRL5"), #cluster 4,6
               TECs = c("PECAM1", "VWF",
                           "ENG", "CDH5"), #cluster 2,7
               CAFs = c("COL1A2", "FAP", "PDPN",
                           "DCN", "COL3A1", "COL6A1"), #cluster 1
               TAMs = c("CD14", "CD163", "CD68", "CSF1R"), #cluster 5,8
               HPClike = c("EPCAM", "KRT19", "PROM1",
                           "ALDH1A1", "CD24") #cluster 3
)
#check marker expressions-scatterplot
FeaturePlot(seurat,
            features = unlist(markers),
            order = T,
            pt.size = 1,
            ncol = 6)
```



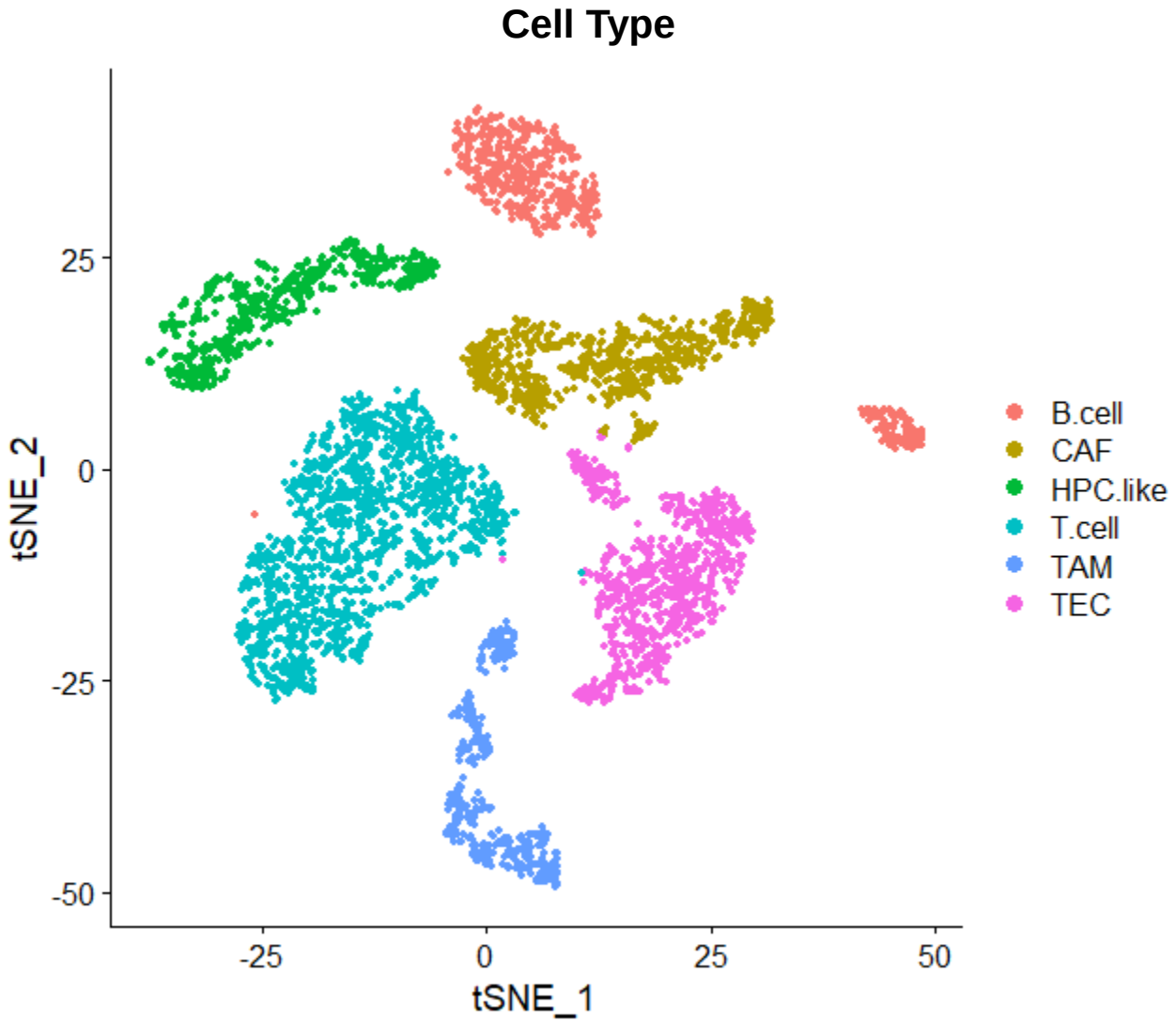
Cell Type Annotation

```r
#check marker expressions-heatmap
avgExprs <- AverageExpression(seurat,
                              features = unlist(markers),
                              assays = "RNA", slot = "data")

scaledExprs <- t(scale(t(avgExprs$RNA)))
scaledExprs[scaledExprs > -min(scaledExprs)] <- -min(scaledExprs)

library(RColorBrewer)
palette_length = 100
my_color = colorRampPalette(rev(brewer.pal(11, "RdBu")))(palette_length)

my_breaks <- c(seq(min(scaledExprs), 0,
                   length.out=ceiling(palette_length/2) + 1),
               seq(max(scaledExprs)/palette_length,
                   max(scaledExprs),
                   length.out=floor(palette_length/2)))

#draw heatmap
pheatmap(scaledExprs,
         cluster_cols = T, cluster_rows = F, clustering_method = "ward.D2",
         treeheight_col = 0,
         breaks = my_breaks, color=my_color,
         labels_row = as.expression(lapply(rownames(scaledExprs), function(a) bquote(italic(.(a))))),
         angle_col = 315
)
```



Cell Type Annotation

```r
# cell type annotation
seurat$celltype = seurat$seurat_clusters

seurat$celltype = gsub(0, "T.cell", seurat$celltype)
seurat$celltype = gsub(1, "CAF", seurat$celltype)
seurat$celltype = gsub(2, "TEC", seurat$celltype)
seurat$celltype = gsub(3, "HPC.like", seurat$celltype)
seurat$celltype = gsub(4, "B.cell", seurat$celltype)
seurat$celltype = gsub(5, "TAM", seurat$celltype)
seurat$celltype = gsub(6, "B.cell", seurat$celltype)
seurat$celltype = gsub(7, "TEC", seurat$celltype)
seurat$celltype = gsub(8, "TAM", seurat$celltype)

DimPlot(seurat,
        reduction="tsne",
        group.by="celltype",
        pt.size = 1)
```



Continuous latent variable approach

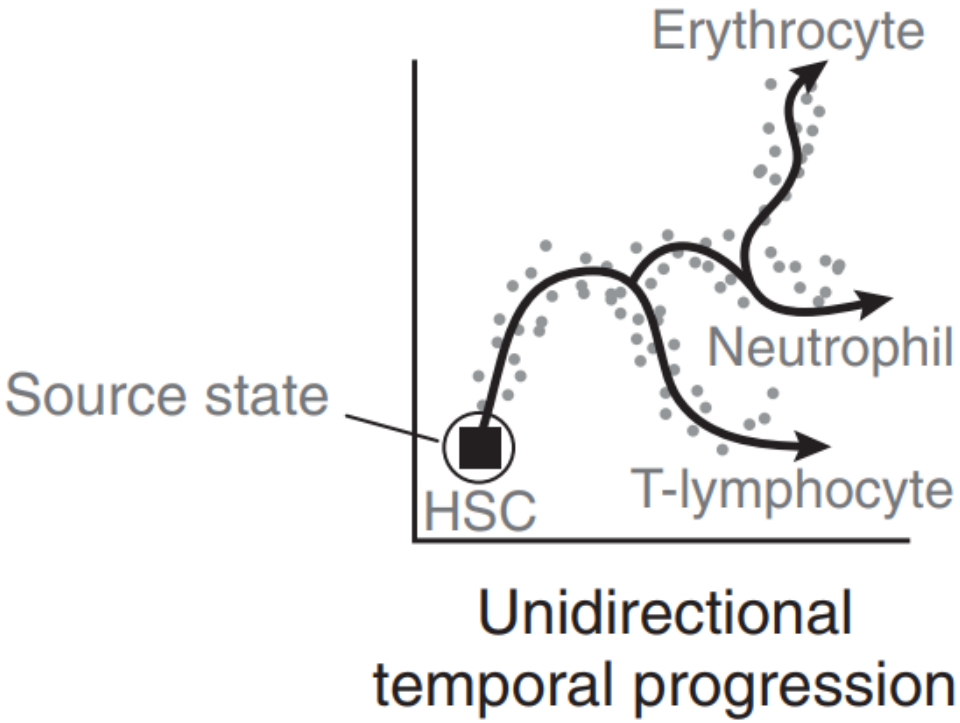
- Pioneered by Monocle.
- Trajectory inference or pseudo-temporal ordering.
- The main assumption underlying this approach is that there exists a dynamic cellular process that shapes the transcriptional landscape and each individual cell can be placed along the process.
- Many dynamic cellular processes,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

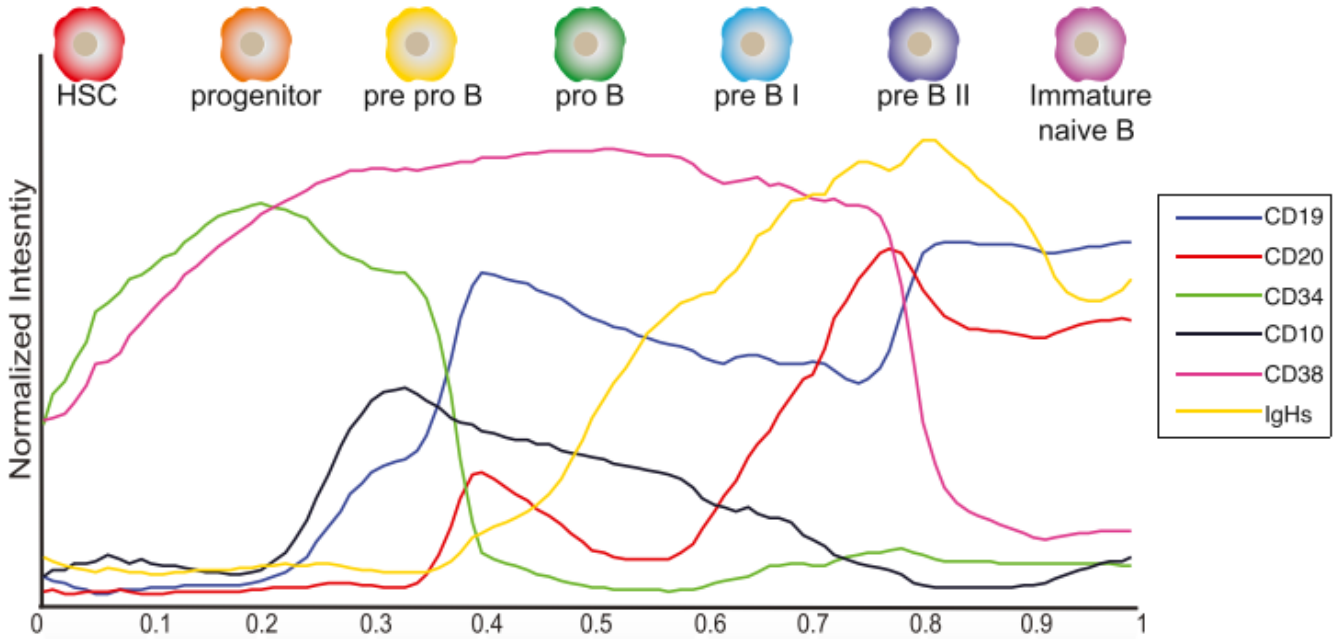
Continuous latent variable approach

- 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.

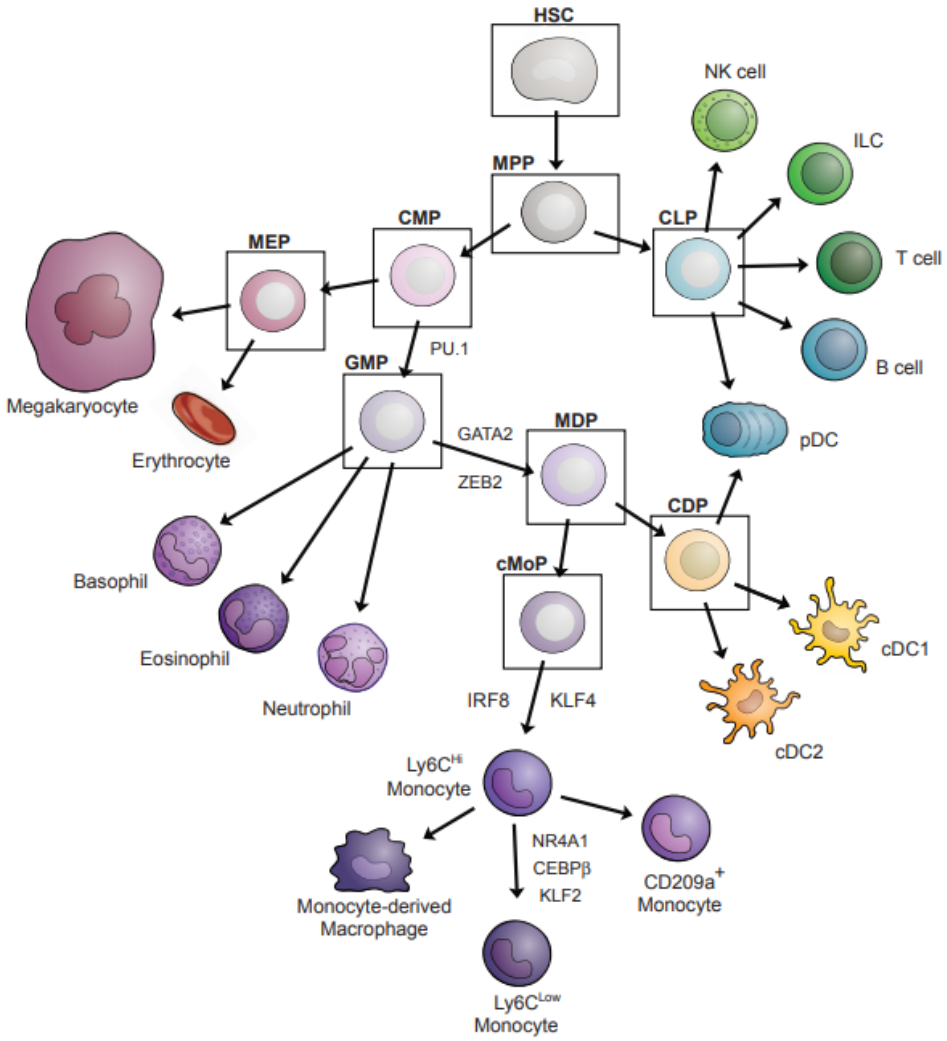
Trajectory inference



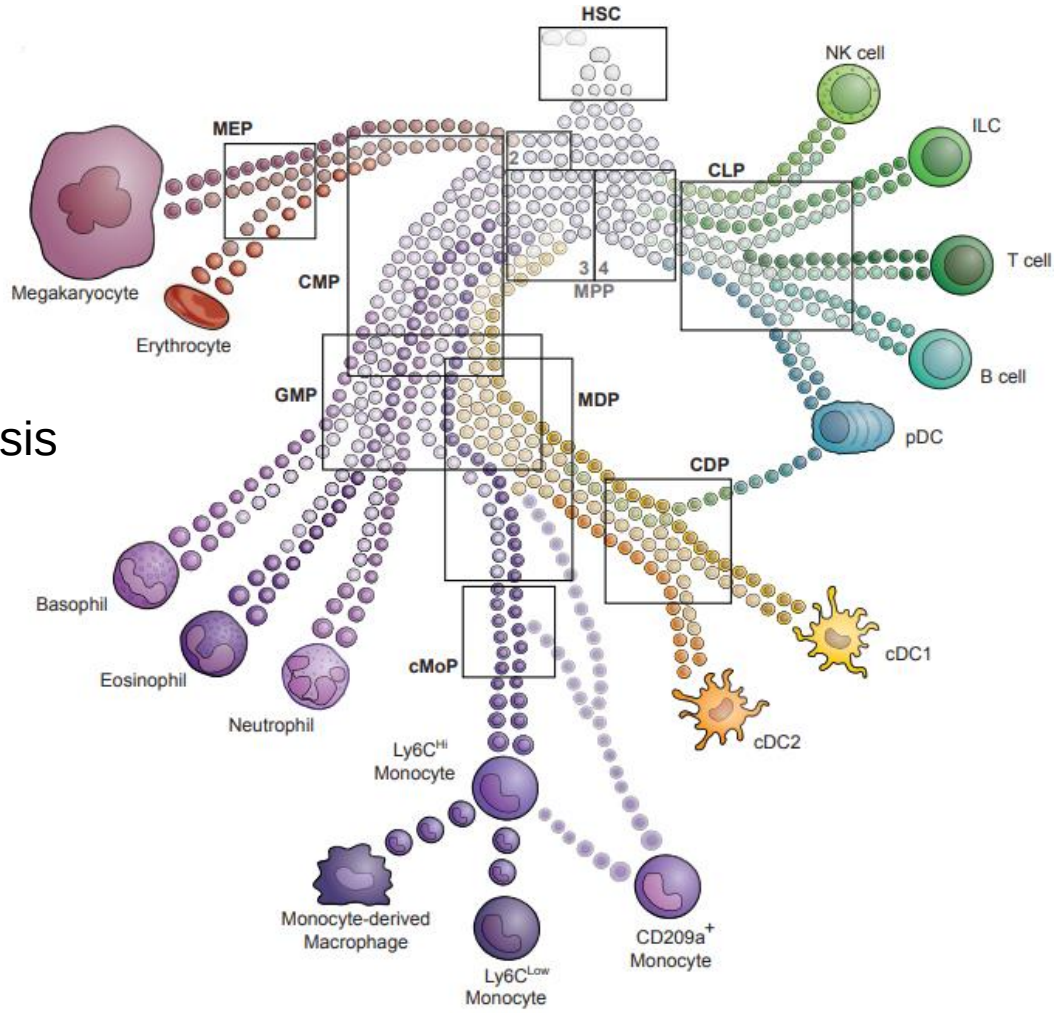


Nat Biotechnol (2016) 34:1145

A single-cell view of murine monocyte development



Single-cell analysis



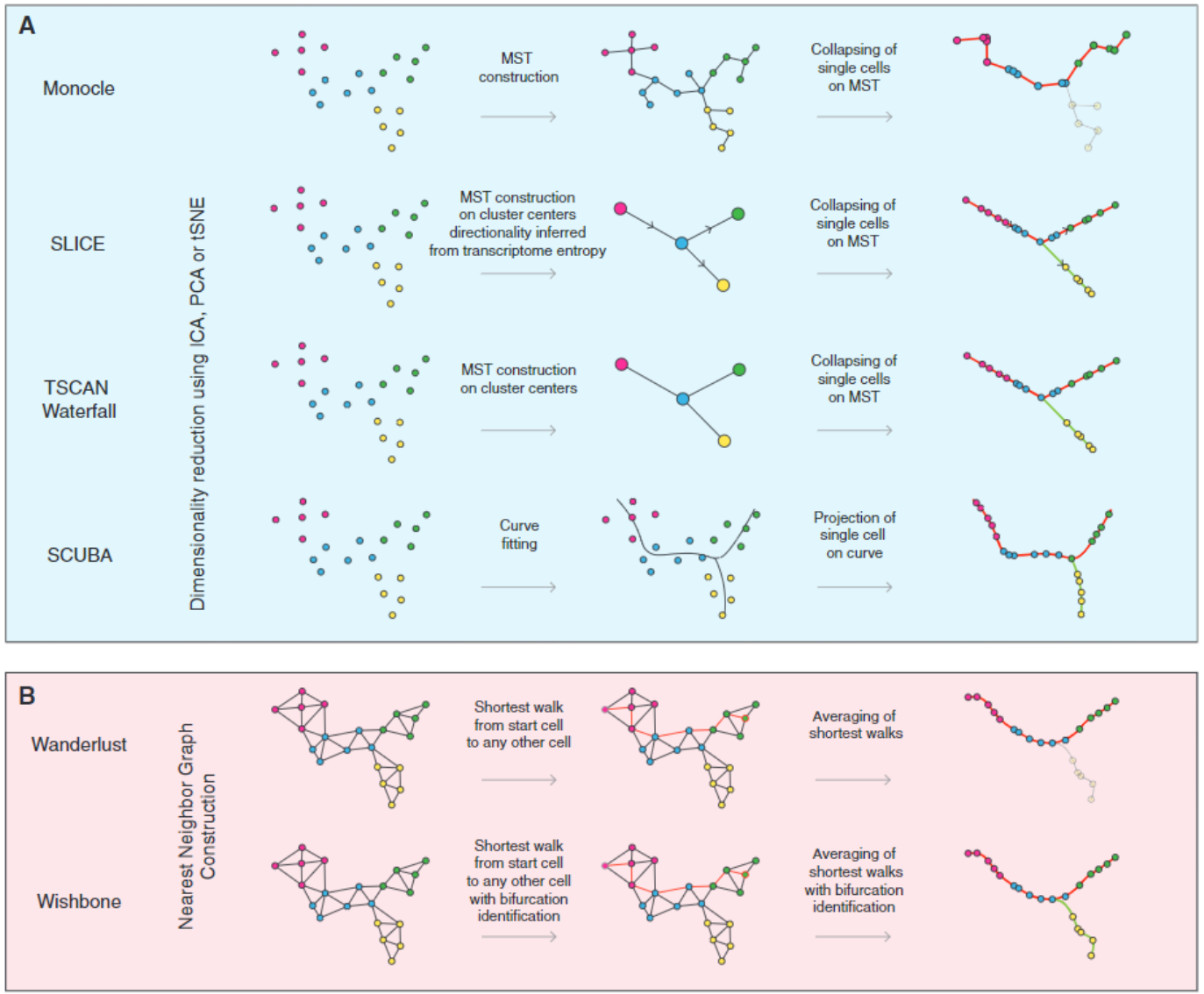
Immunity (2018) 49: 595-613

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 (2018) 23:166

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 (kNN) in which each cell is represented as a node and each node is linked its k-nearest 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-NN graph is usually constructed after projecting cells to the low-dimensional feature space using dimensionality reduction methods.

- For better visualization, k-NN graphs 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).

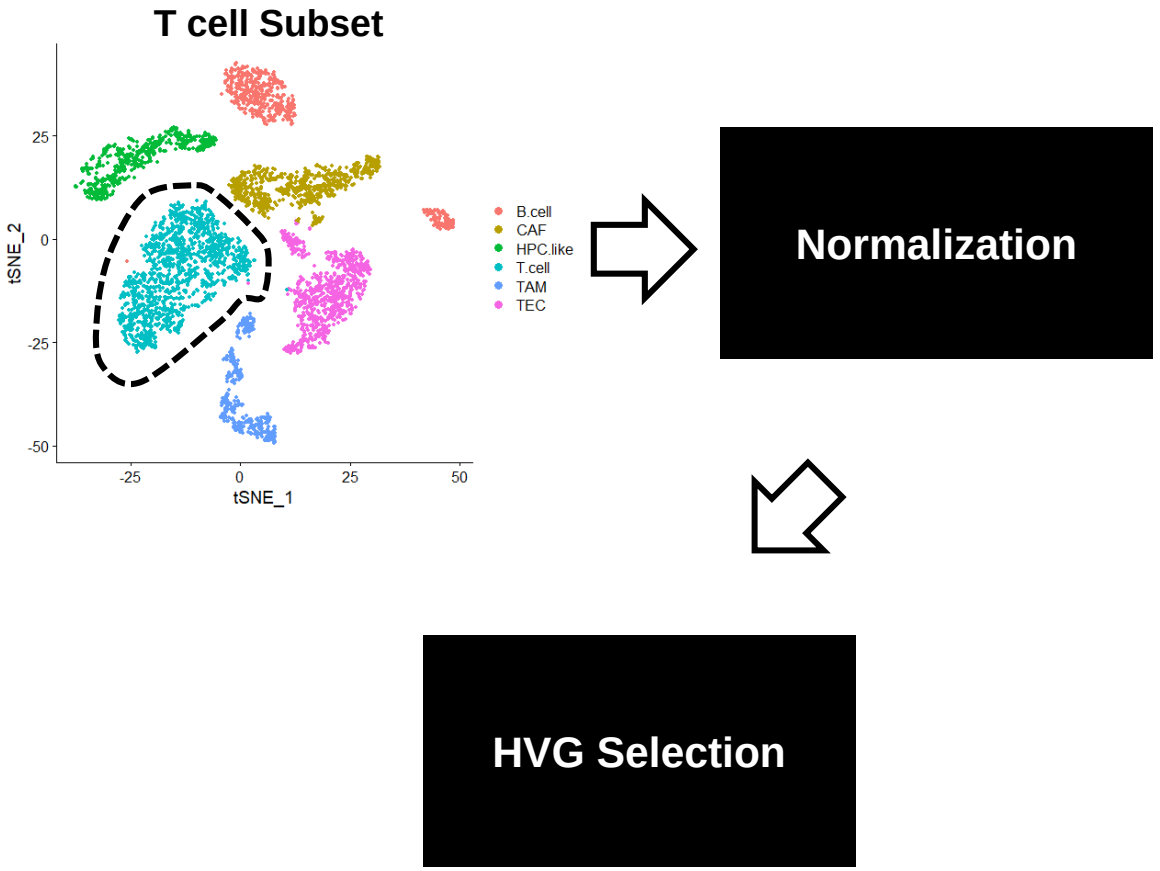
Trajectory inference: Comparison

- “A comparison of single-cell trajectory inference methods” Nature Biotechnology 37:547 (2019)

Trajectory Analysis for T cell

: Subset T cell/HVG Selection

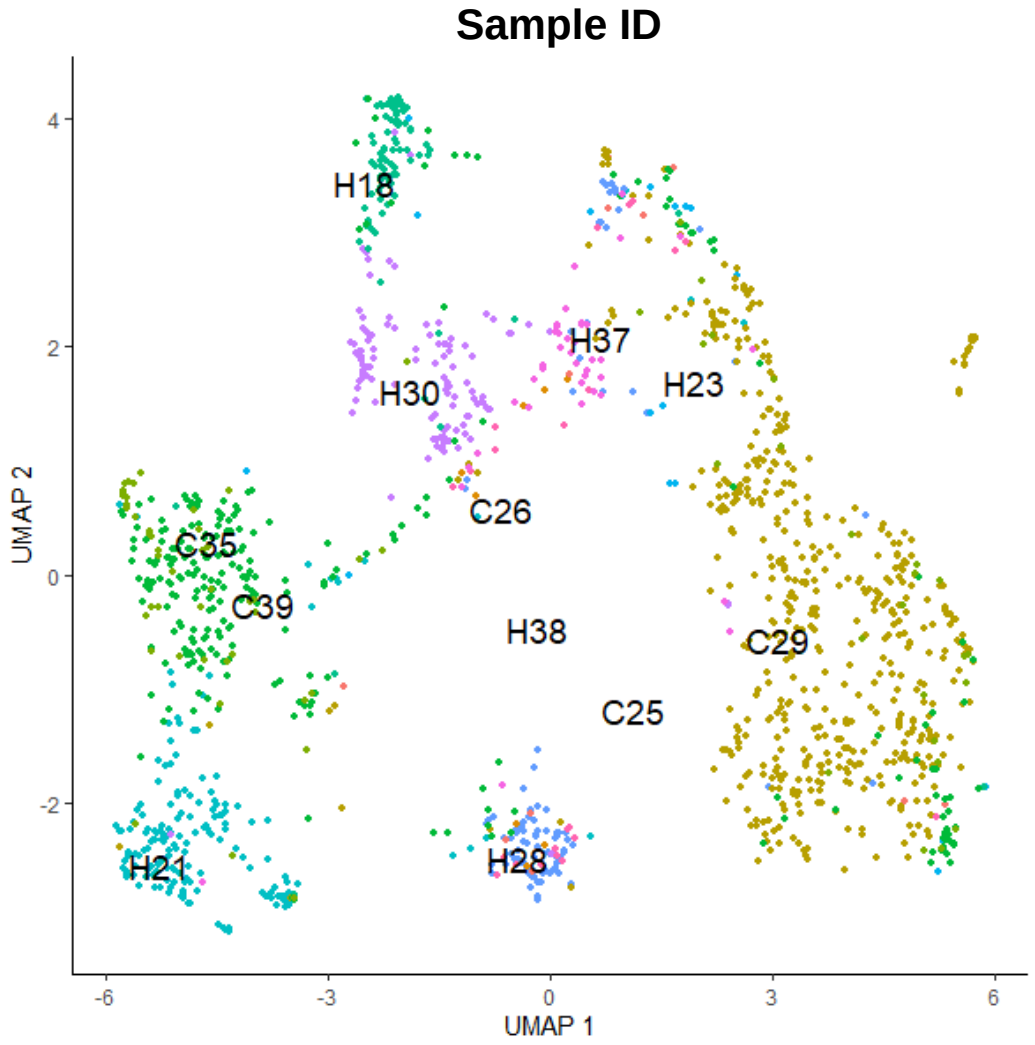
```r
#extract T cell population
seurat_t <- seurat[, seurat$celltype == "T.cell"]
seurat_t <- seurat_t[rowSums(seurat_t@assays$RNA@counts) != 0, ]

sce_t <- as.SingleCellExperiment(seurat_t)

clusters <- quickCluster(sce_t)
sce_t <- computeSumFactors(sce_t, clusters = clusters)
sce_t <- logNormCounts(sce_t)

#HVG selection for T cell
dec <- modelGeneVar(sce_t)
plot(dec$mean, dec$total, xlab="Mean log-expression", ylab="Variance")
curve(metadata(dec)$trend(x), col="blue", add=TRUE)

hvg.t <- getTopHVGs(dec, fdr.threshold = 0.05)
length(hvg.t) # 200 genes
```



Trajectory Analysis for T cell

: Dimension Reduction

```
# t cell trajectory
library(monocle3)
cell_metadata = colData(sce_t)
gene_metadata = data.frame(gene_short_name = rownames(sce_t),
                           row.names = rownames(sce_t))
# generate cell_data_set object
cds <- new_cell_data_set(sce_t@assays@data$logcounts,
                         cell_metadata = cell_metadata,
                         gene_metadata = gene_metadata)

hvg.t <- gsub("_", "-", hvg.t) #replaced character after hvg selection

#dimension reduction without batch correction
cds <- preprocess_cds(cds, "PCA", num_dim = 50,
                      norm_method = "none", use_genes = hvg.t)
monocle3::plot_pc_variance_explained(cds)

cds <- reduce_dimension(cds, reduction_method = "UMAP")
plot_cells(cds, color_cells_by = "ID",
           cell_size = 1,
           group_label_size = 5)
```

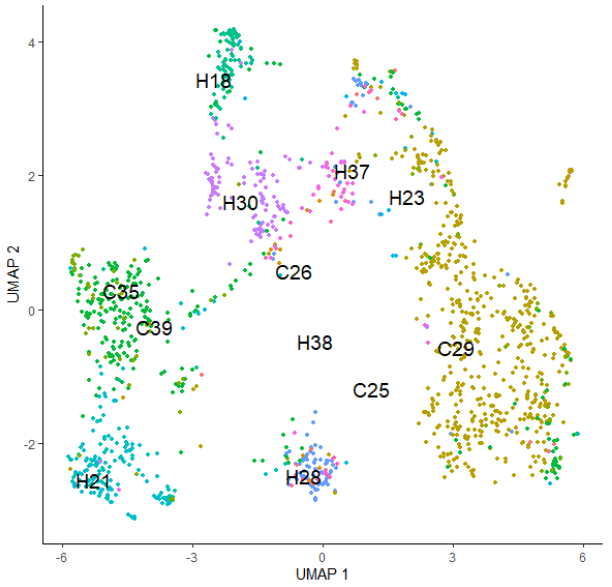


Trajectory Analysis for T cell

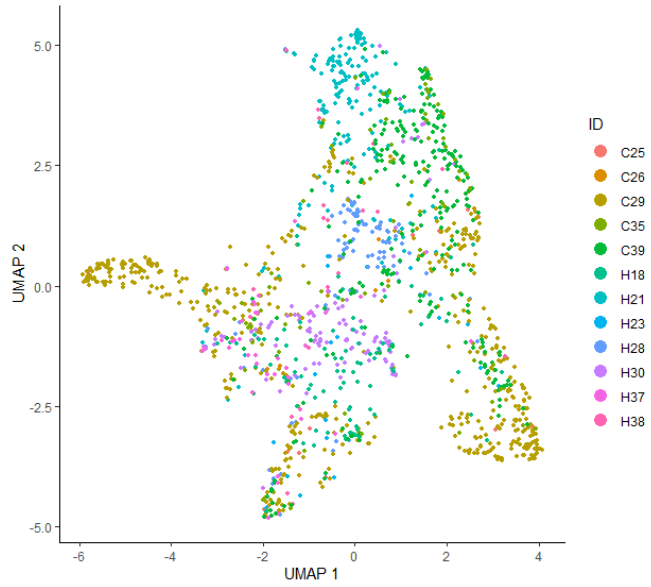
: Batch Correction (MNN)

```r
#batch correction with monocle3::align_cds
#(MNN implemented Batchelor R package)
cds_aligned <- preprocess_cds(cds, "PCA",
                              num_dim = 30,
                              norm_method = "none",
                              use_genes = hvg.t)

#batch correction
cds_aligned <- align_cds(cds_aligned,
                         alignment_group = "ID")
cds_aligned <- reduce_dimension(cds_aligned,
                                preprocess_method = "Aligned")
plot_cells(cds_aligned,
           color_cells_by = "ID",
           cell_size = 1,
           label_cell_groups = FALSE)
```

Before Correction



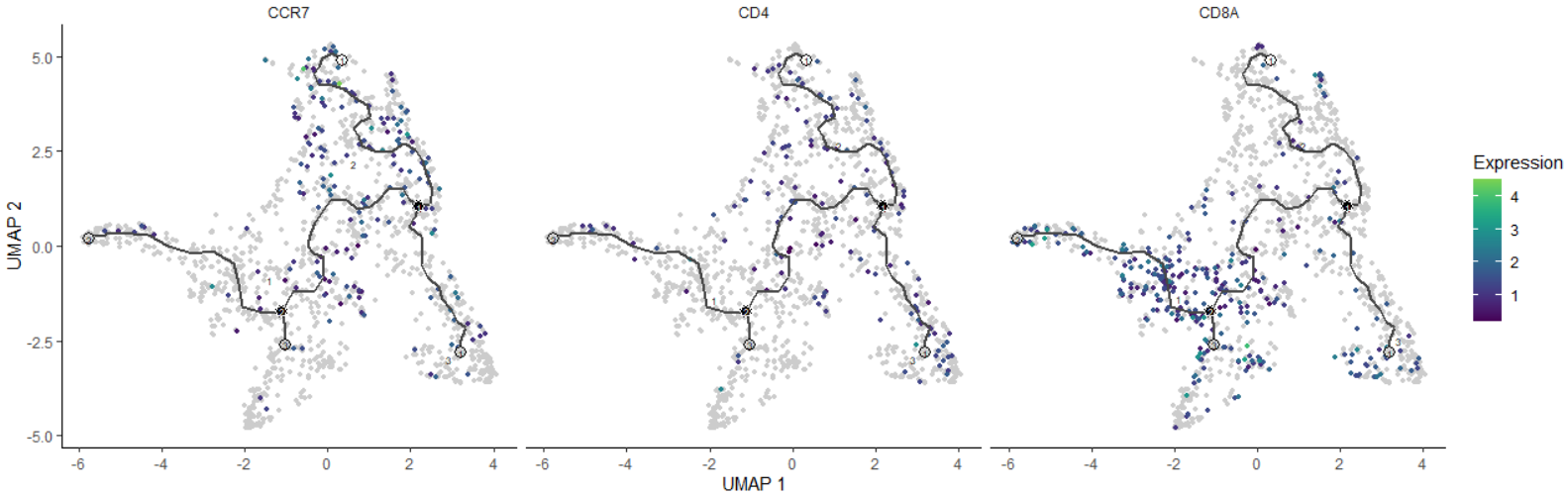
After Correction

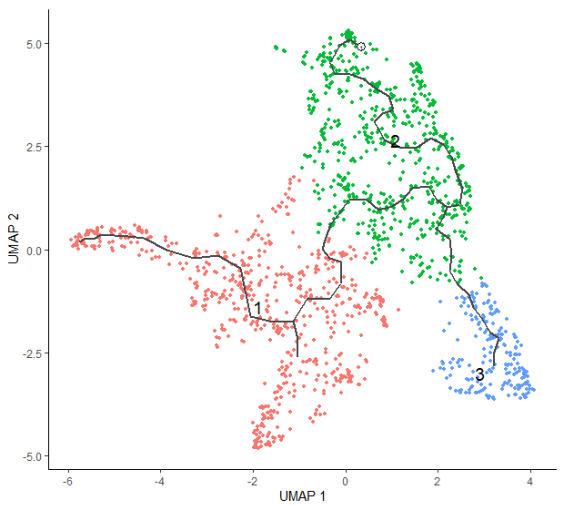


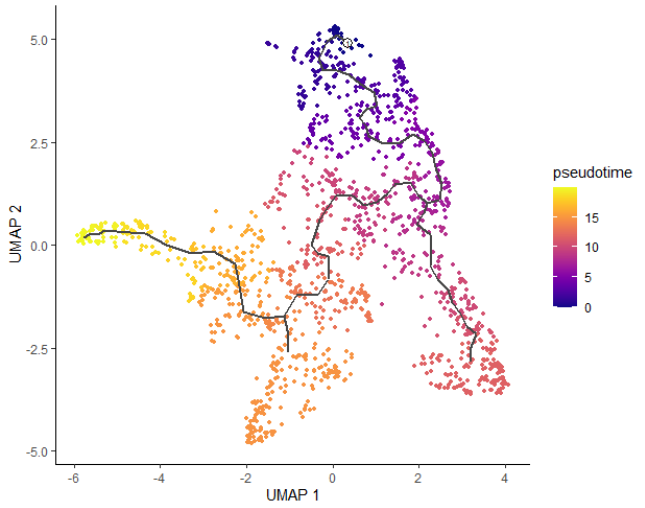
Trajectory Analysis for T cell

: Clustering/Trajectory Analysis

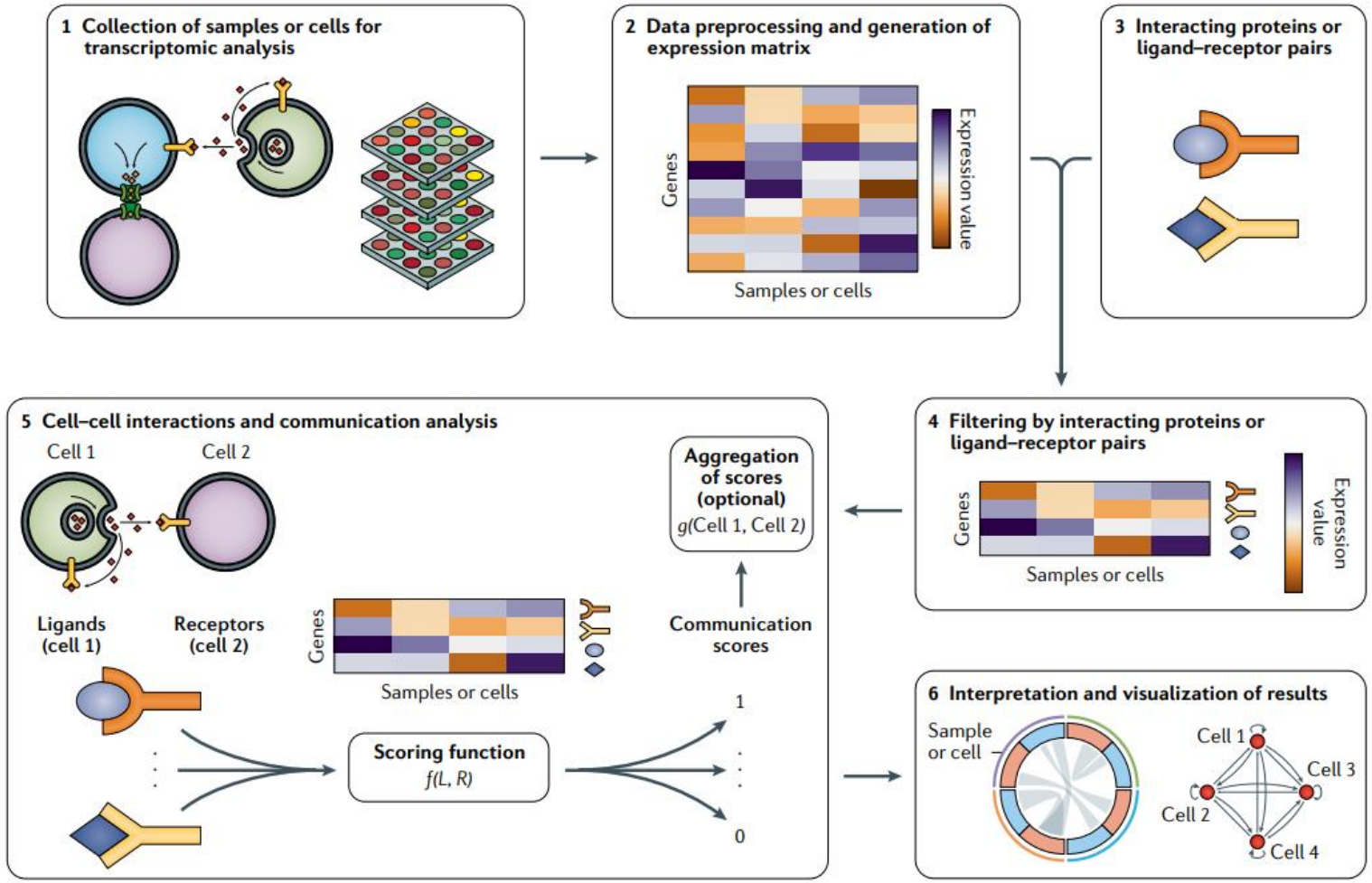
```
cds_aligned <- cluster_cells(cds_aligned,
                             resolution = 1e-3)
cds_aligned <- learn_graph(cds_aligned)
cds_aligned <- order_cells(cds_aligned)

plot_cells(cds_aligned,
           color_cells_by = "cluster",
           cell_size = 1,
           group_label_size = 5,
           label_leaves = FALSE,
           label_branch_points = FALSE)

plot_cells(cds_aligned,
           genes = c("CCR7", "CD4", "CD8A"),
           cell_size = 1,
           norm_method = "size_only")

plot_cells(cds_aligned,
           color_cells_by = "pseudotime",
           cell_size = 1,
           label_groups_by_cluster = FALSE,
           label_leaves = FALSE,
           label_branch_points = FALSE)
```





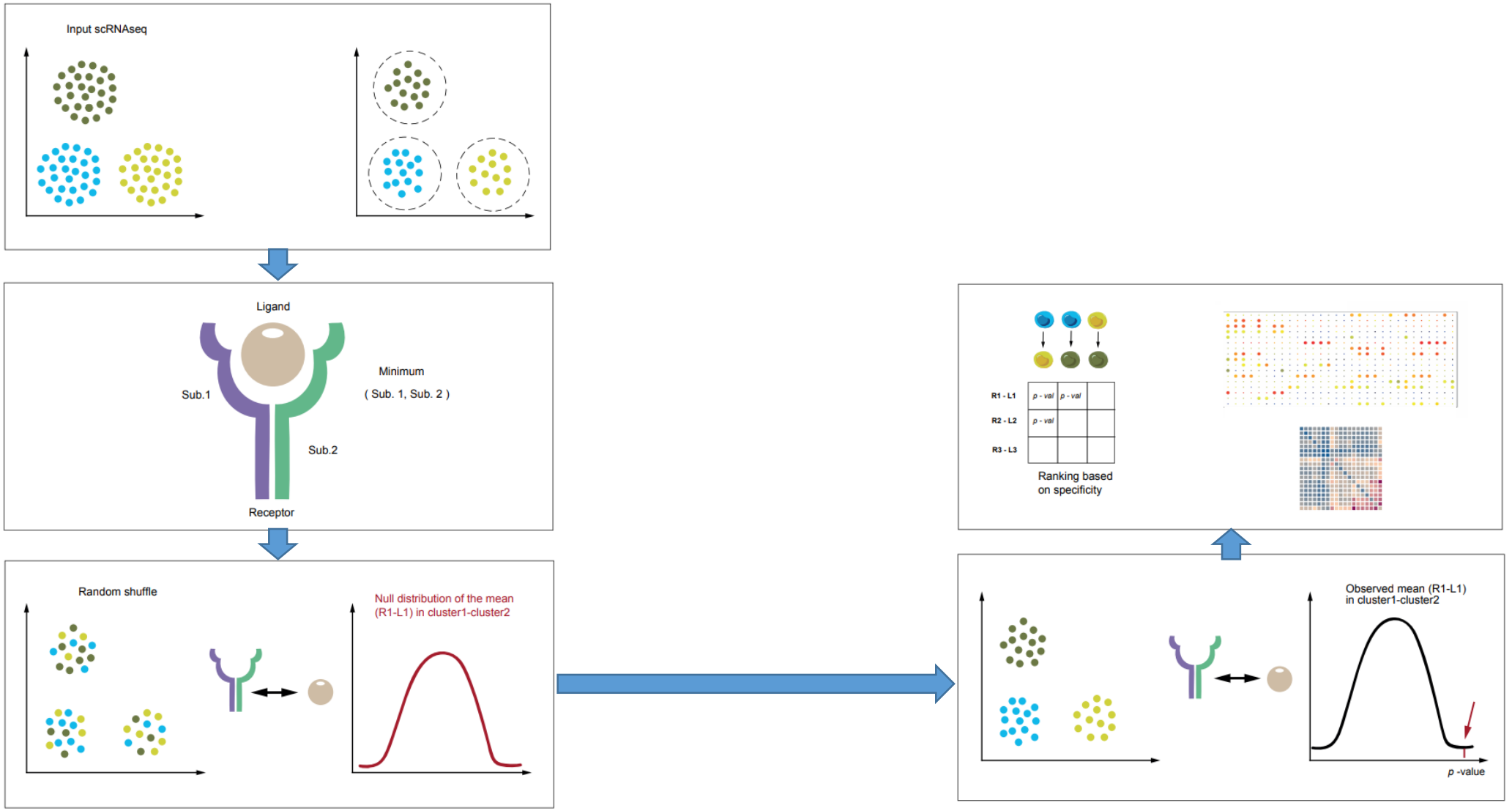


Intercellular communication



Nat Rev Genet (2020) 22:71

Intercellular communication: CellPhoneDB



Nature (2018) 536:347; Nat Protocols (2020) 15:1484