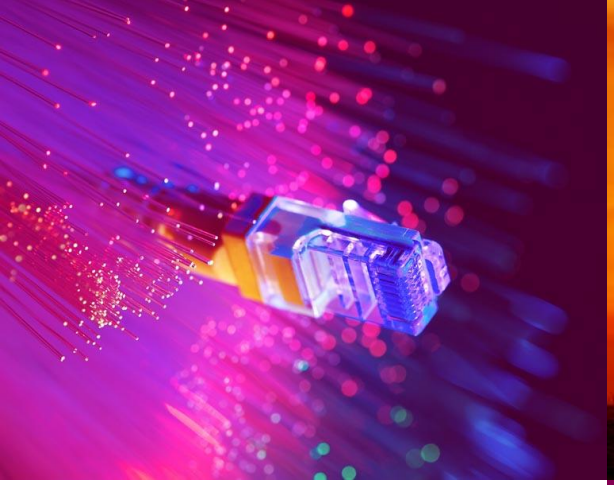


CSC

Single-cell RNA-seq data analysis using Chipster

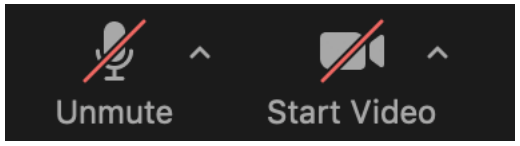
March 2025

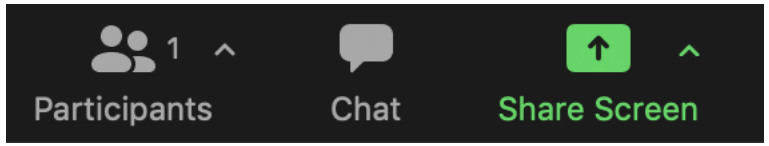
Eija Korpelainen, Maria Lehtivaara

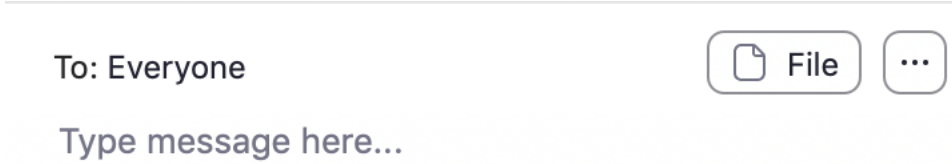
CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

Virtual coffee & get together

- Test your mic & tell us a little bit about yourself (in the google doc)!
  - Name
  - Institute
  - Do you already have scRNA-seq data and what kind (e.g. 10X)
  - Best thing about winter? (make sure to enjoy it after the course to load your batteries!)
- Zoom etiquette
  - When you are not talking, please keep your mic muted
  - You can find all the controls (mic, video, chat, screen sharing) at the bottom of the Zoom window
  - In chat: write your questions to Everyone, not just to Hosts
  - Please use a headset to avoid the echo







Understanding data analysis - why?

- Bioinformaticians might not always be available when needed
- Biologists know their own experiments best
  - Biology involved (e.g. genes, pathways, etc)
  - Potential batch effects etc
- Allows you to design experiments better → less money wasted
- Allows you to discuss more easily with bioinformaticians

What will I learn?

- Analysis of single-cell RNA-seq (scRNA-seq) data
  - Find clusters of cells (cell types) and marker genes for them
  - Compare multiple samples (e.g. treatment vs control)
    - Identify cell types that are present in both samples
    - Obtain cell type markers that are conserved in both samples
    - Compare the samples to find cell-type specific responses to treatment
- Note: you can use the same analysis pipeline for single-nuclei RNA-seq (snRNA-seq)
- How to operate the Chipster software

Introduction to single-cell RNA-seq data analysis

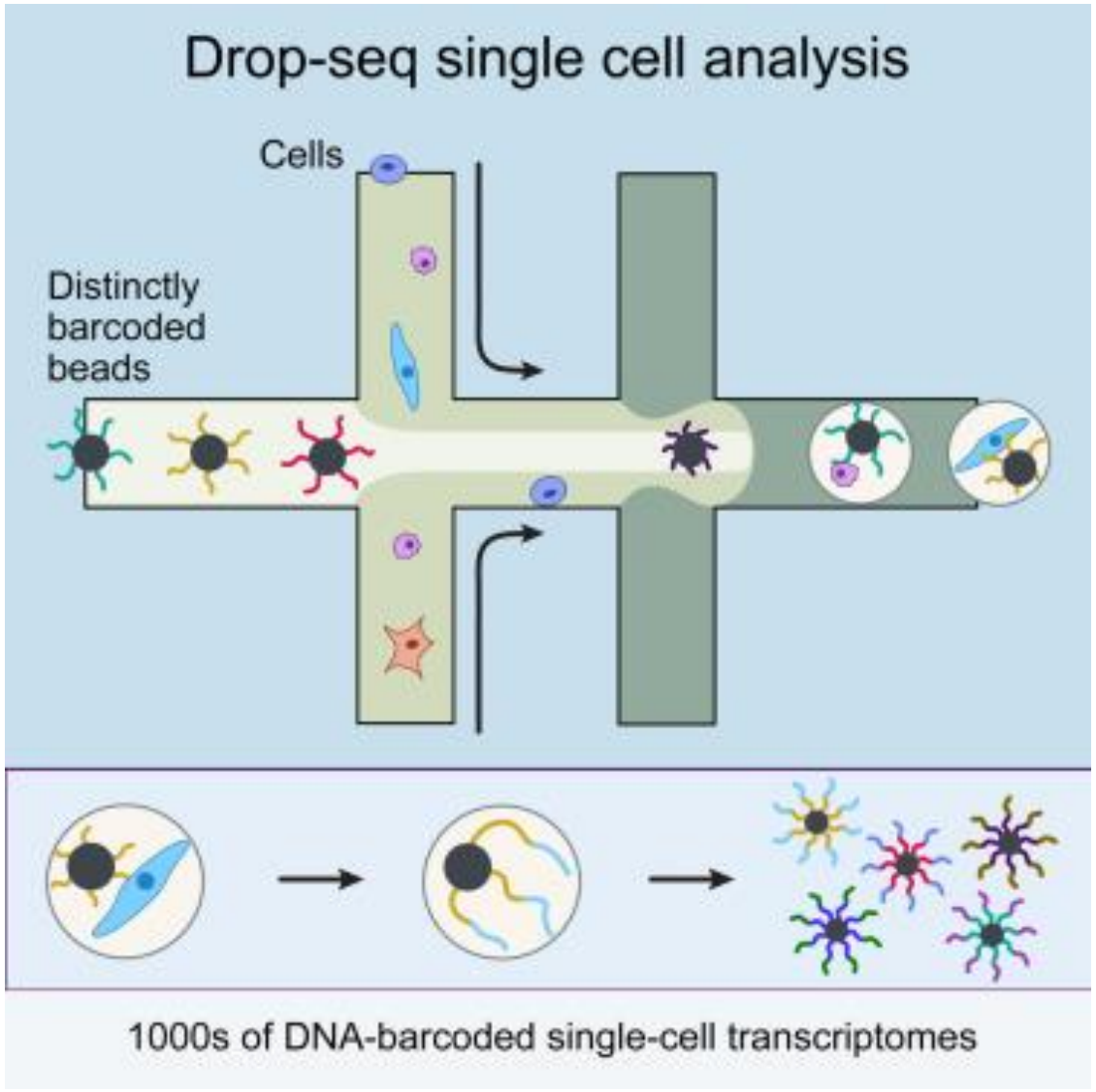
What will you learn

1. How does scRNA-seq work and what can go wrong
   - Empties, doublets and dropouts
   - What are UMIs and why do we use them
2. Why is scRNA-seq data challenging to analyze
3. What are the main analysis steps for clustering cells and finding cluster marker genes
4. What is Seurat

Single-cell RNA-seq

- Relatively new technology, data analysis methods are actively developed
- Gene expression profiling at single cell level has many applications
  - cell type detection, cellular differentiation processes, tumor heterogeneity and response to drugs, etc
- Many technologies for capturing single cell transcriptomes
  - Droplet-based (e.g. 10X Chromium, Drop-seq), plate-based and well-based
- Libraries are usually 3’ tagged: only a short sequence at the 3’ end of the mRNA is sequenced
- The size and scale of single-cell sequencing datasets is rapidly increasing

Bead: Cell barcode and unique molecular identifiers (UMIs)



- Cell barcode: which cell the read comes from
- UMI: which mRNA molecule the read comes from (helps to detect PCR duplicates)

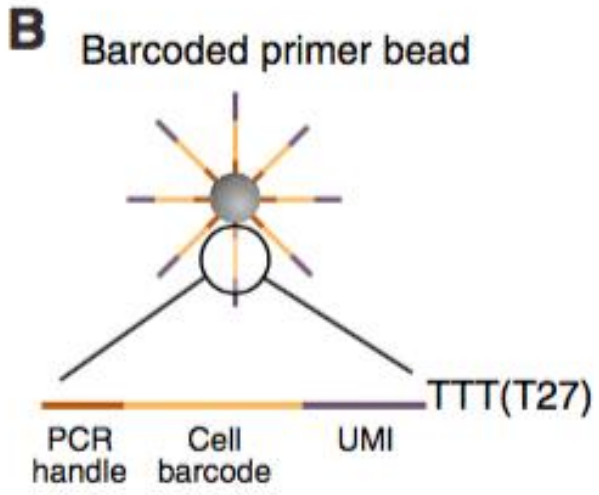


Figure by Macosko et al, Cell, 161:1202-1214, 2015

From reads to digital gene expression matrix (DGE)

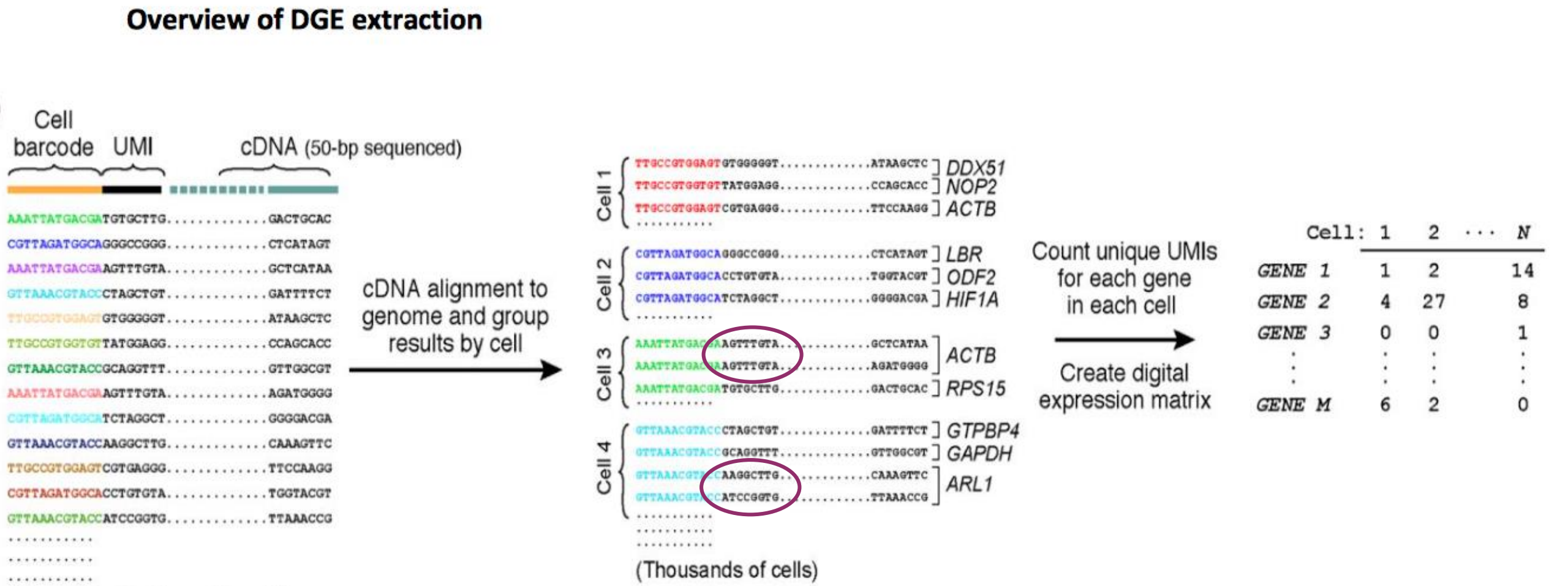


Figure by Macosko et al, Cell, 161:1202-1214, 2015

What can go wrong?

1. Ideally there is one healthy cell in the droplet. However, sometimes
   - There is no cell in the droplet, just ambient RNA

     →Remove "empties" based on the small number of genes expressed
   - There are two (or more) cells in a droplet

     → Remove doublets (and multiplets) based on the large number of genes expressed
   - The cell in the droplet is broken/dead

     → Remove dead cells based on high percentage of mitochondrial transcripts
2. Sometimes barcodes have synthesis errors in them, e.g. one base is missing

   → Check the distribution of bases at each position and fix the barcode or remove the cell

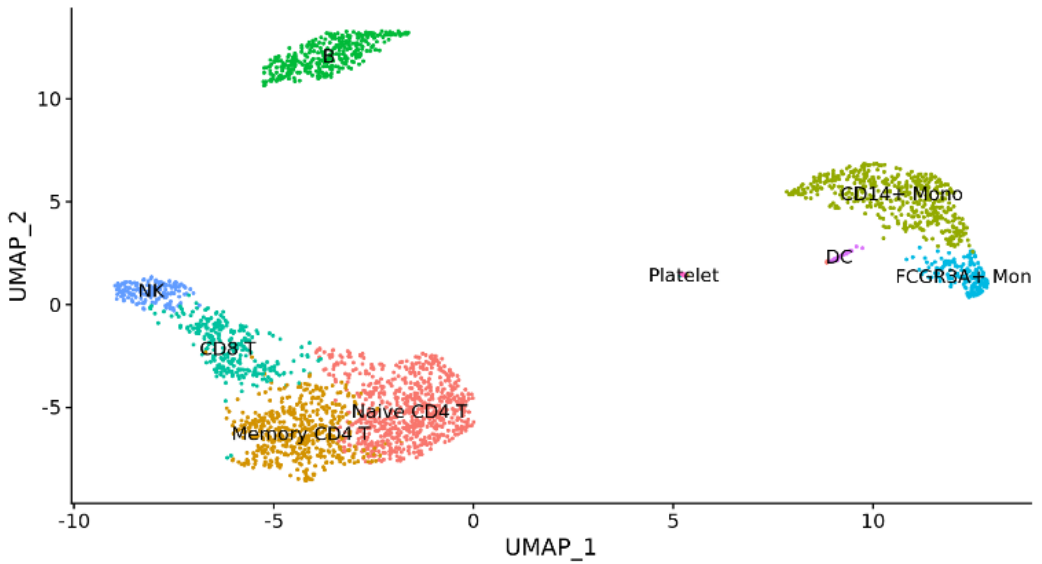
Single-cell RNA-seq data is challenging

- High number of dropouts
  - Dropout: a gene is expressed but the expression is not detected due to technical limitations → the detected expression level for many genes is zero
- Data is noisy. High level of variation between the cells due to
  - Capture efficiency (percentage of mRNAs captured)
  - Reverse transcription efficiency
  - Amplification bias (non-uniform amplification of transcripts)
  - Significant differences in sequencing depth (number of UMIs/cell)
  - Cell size and cell cycle stage
- Complex distribution of expression values
  - Cell heterogeneity and the abundance of zeros give rise to multimodal distributions

→ Analysis methods for bulk RNA-seq data don't work for single-cell RNA-seq

Analysis steps for clustering cells and finding cluster marker genes

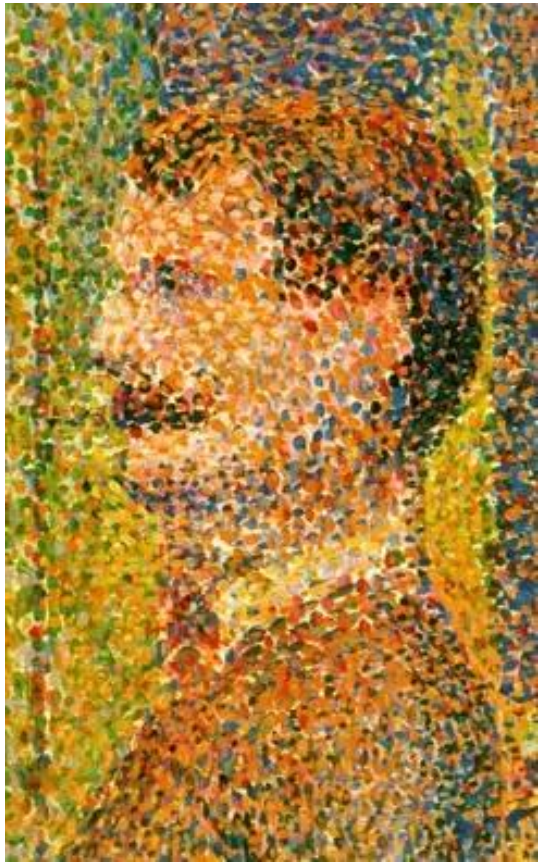
1. Check the quality of cells, filter genes
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation



6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (UMAP or tSNE) using the PCs
10. Detect and visualize marker genes for the clusters

Seurat

- One of the most popular R packages for scRNA-seq data analysis
- Provides tools for all the steps mentioned in the previous slide
  - Also tools for integrative analysis
- Stores data in Seurat object
  - Contains specific slots for different types of data like counts, PCA and clustering results, etc
- http://satijalab.org/seurat



Detail from La Parade (1889) by Georges Seurat

Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

What will you learn

1. What kind of input files can be used
2. What is the structure of 10X Genomics matrix file
3. How to filter out genes
4. How to check the quality of cells and filter out bad ones

Start the analysis with one of these 3 types of input files in Chipster:

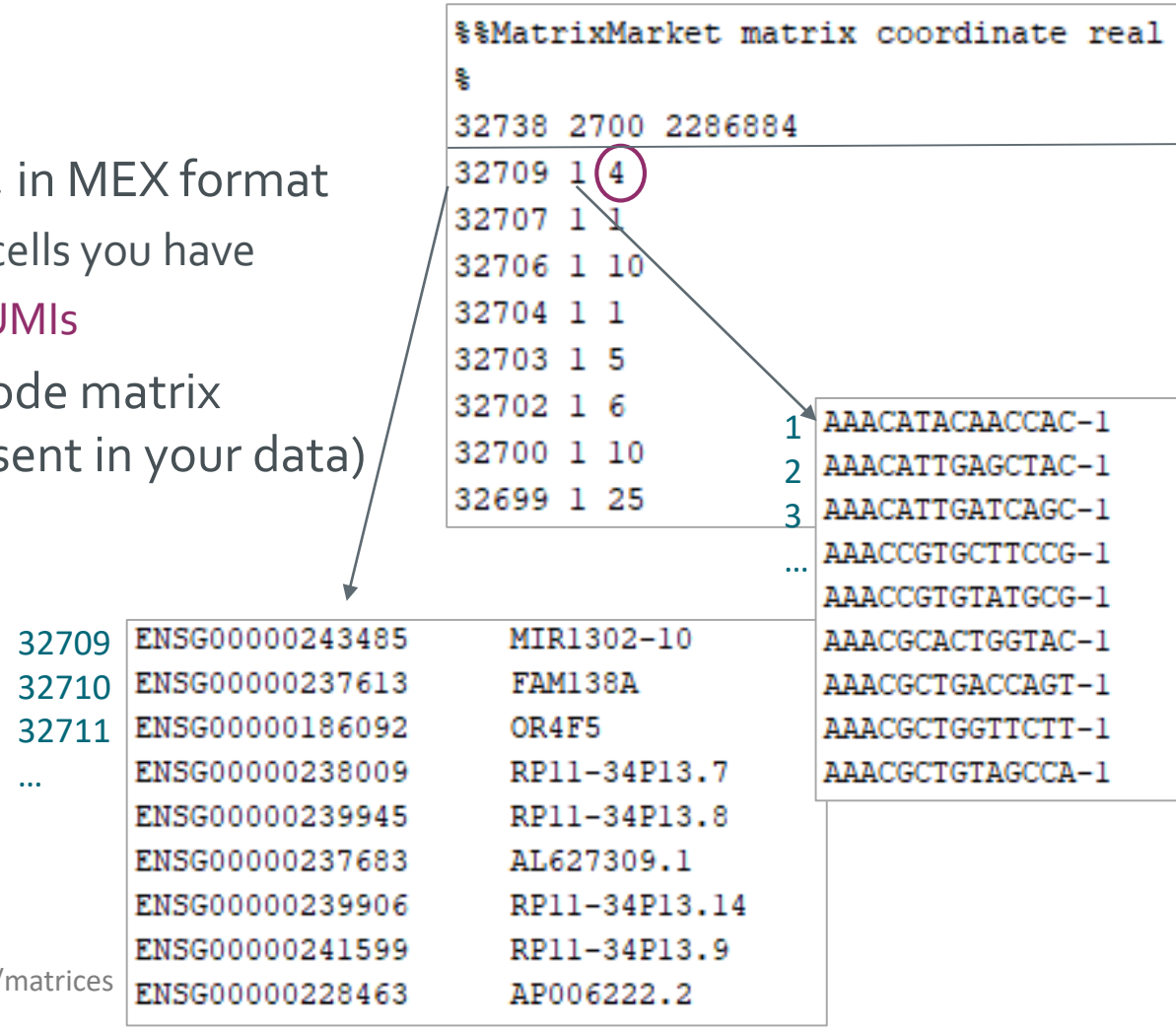
- 10X Genomics Market Exchange (MEX) format
  - Three files are needed: barcodes.tsv, features.tsv (genes.tsv) and matrix.mtx
    - the files need to be named exactly like this
  - You need to put the files in a tar package (use Chipster tool "Utilities / Make a Tar package")

1. 10X Genomics CellRanger or CellBender HDF5 output format (.h5 file)
   - Hierarchical Data Format (HDF5 or H5) is a binary format that can compress and access data much more efficiently than text formats such as MEX, so it is especially useful for large datasets.
2. DGE (=digital gene expression) matrix (.tsv file)
   - DGE matrix made with DropSeq tools in Chipster, or import a ready-made matrix (.tsv file)
   - Genes as rows, cells as columns

- Check that the input file is correctly assigned!

What do the 10X files contain?

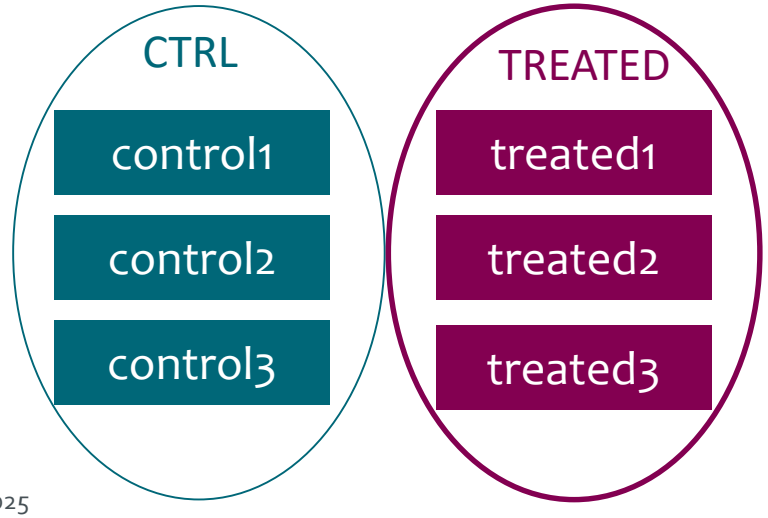
1. **matrix.mtx**
   - Number of UMIs for a given gene in a given cell
   - Sparse matrix (only non-zero entries are stored), in MEX format
     - Header: third line tells how many genes and cells you have
     - Each row: gene index, cell index, number of UMIs
   - Make sure that you use the filtered feature barcode matrix (contains only those cell barcodes which are present in your data)
2. **barcodes.tsv**
   - Cell barcodes present in your data
3. **features.tsv (genes.tsv)**
   - Identifier, name and type (gene expression)

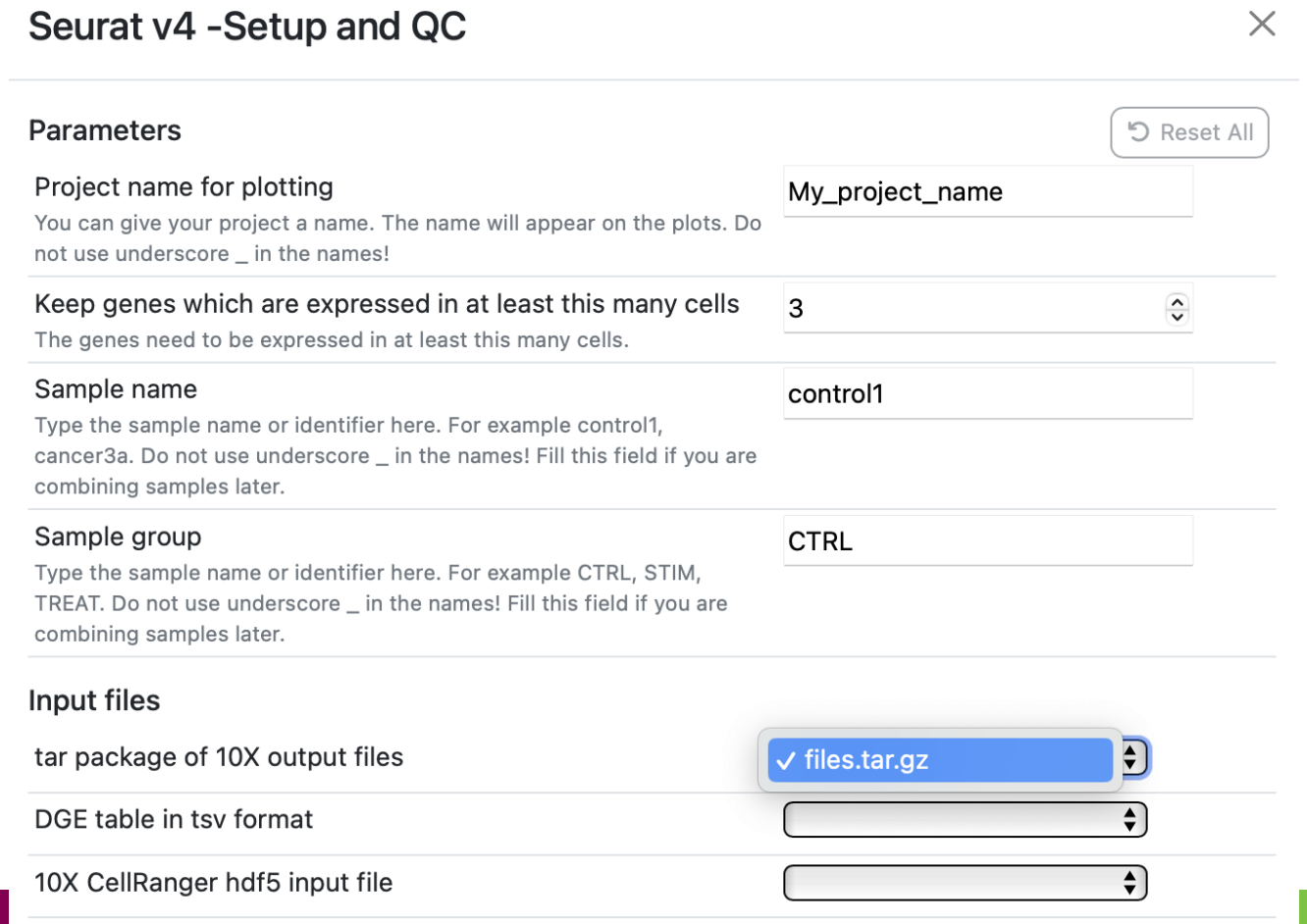


https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/output/matrices

Setting up a Seurat object, filtering genes

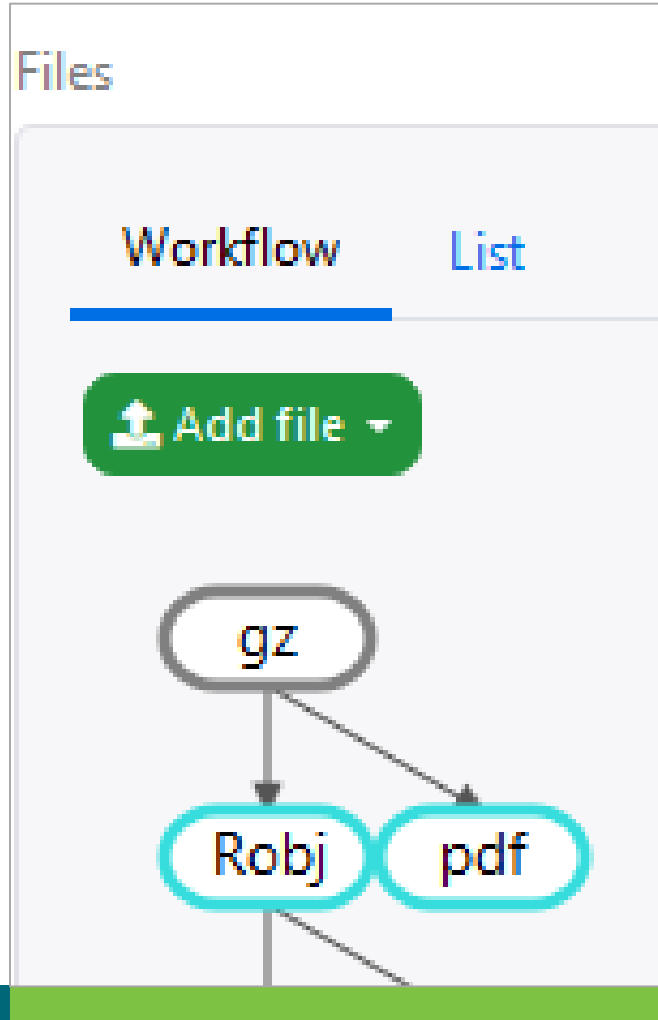
- Give a name for the project (used in some plots)
- Filtering genes
  - Keep genes which are expressed (= detected) in at least this number of cells
- Input files
  - Assign correctly!
- Sample name & sample group
  - Important if you have several samples





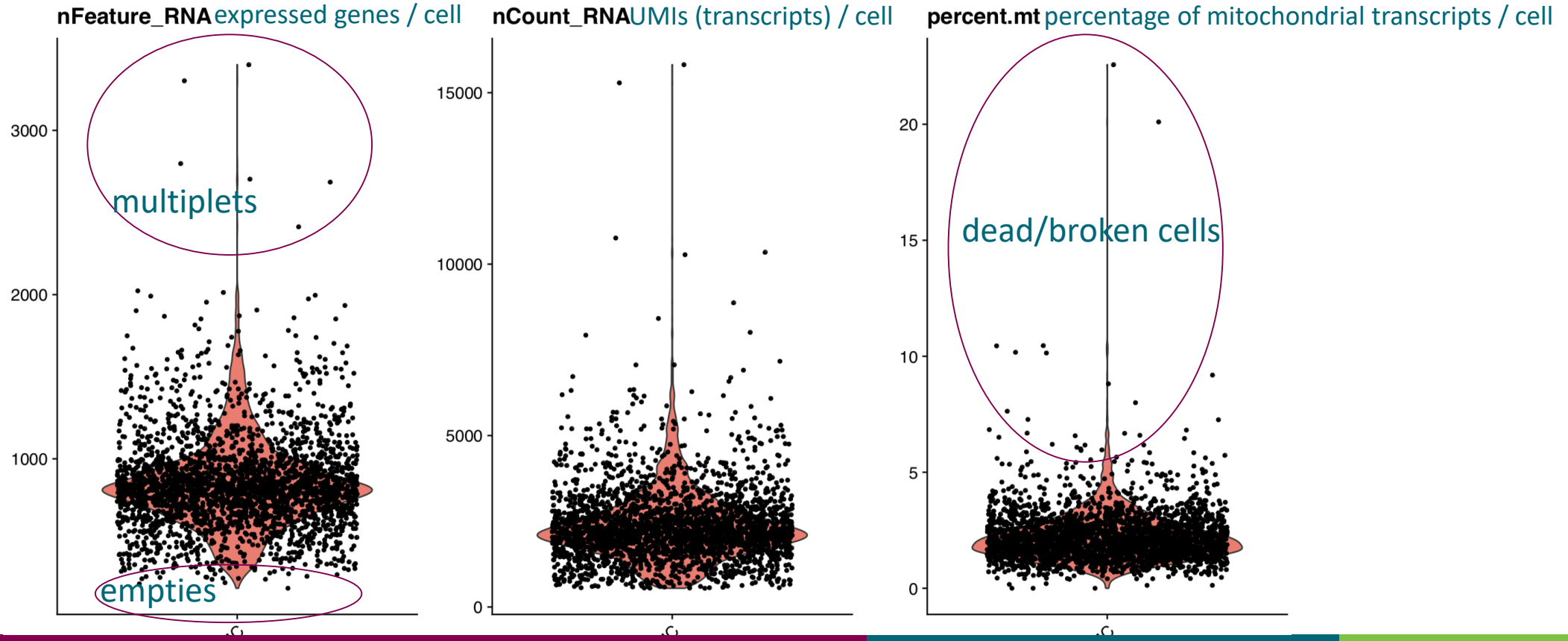
Output files

- Seurat object (Robj) that Seurat-based tools use to store data
  - Use this file as input for the next analysis tool
  - Contains specific slots for different types of data
    - View the contents using the tool Extract information from Seurat object
  - You cannot open the file in Chipster but you can import it to R
- Pdf file with quality control plots and cell number info
  - nFeature_RNA = number of expressed genes in a cell
  - nCount_RNA = number of transcripts in a cell
  - percent.mt = percentage of mitochondrial transcripts
  - percent.rb = percentage of ribosomal transcripts



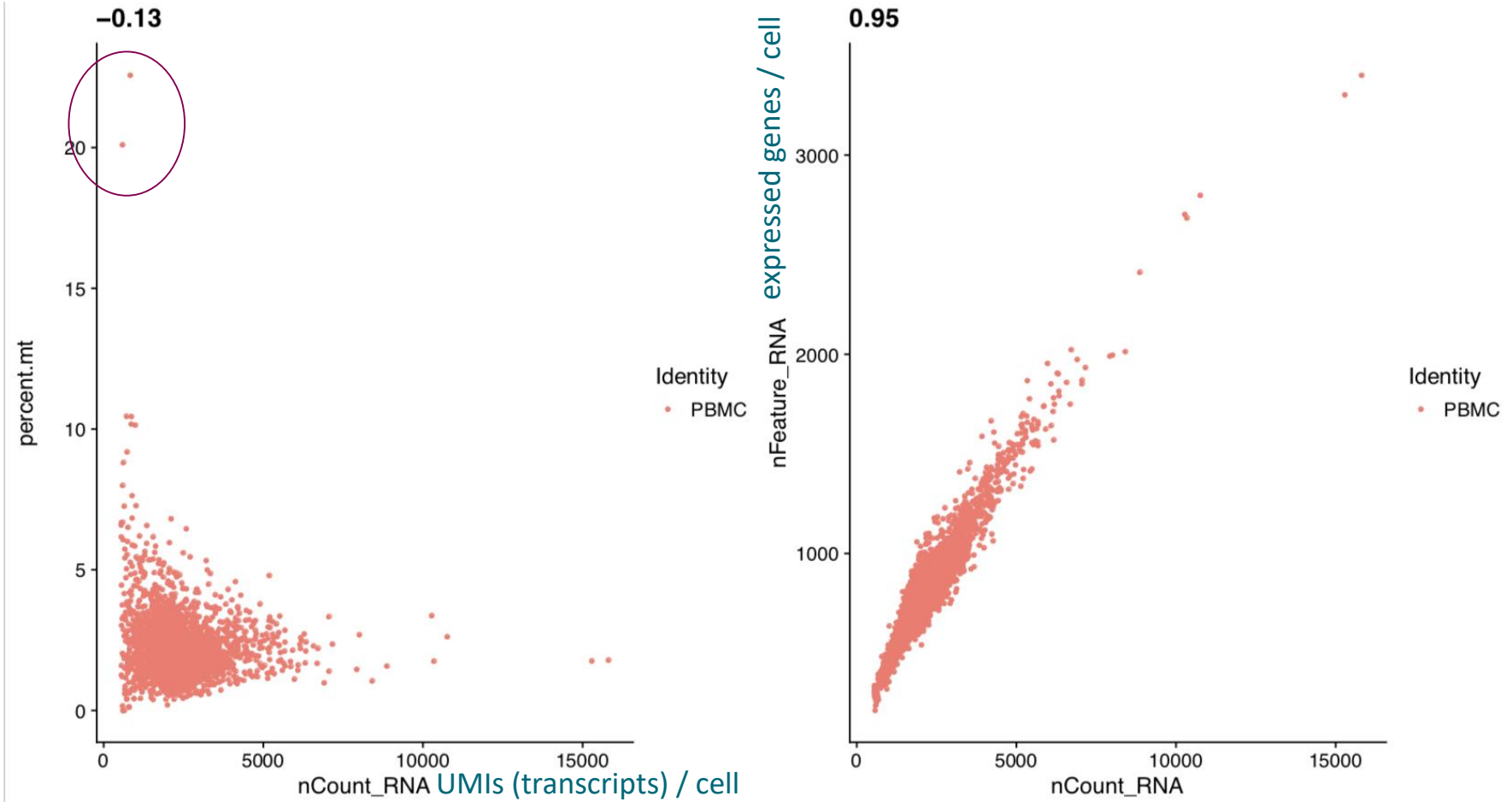
How to detect empties, multiplets and broken cells?

- Empty = no cell in droplet: low gene count (nFeature_RNA < 200)
- Doublet/multiplet = more than one cell in droplet: large gene count (nFeature_RNA > 2500)
- Broken/dead cell in droplet: lot of mitochondrial transcripts (percent.mt > 5%)



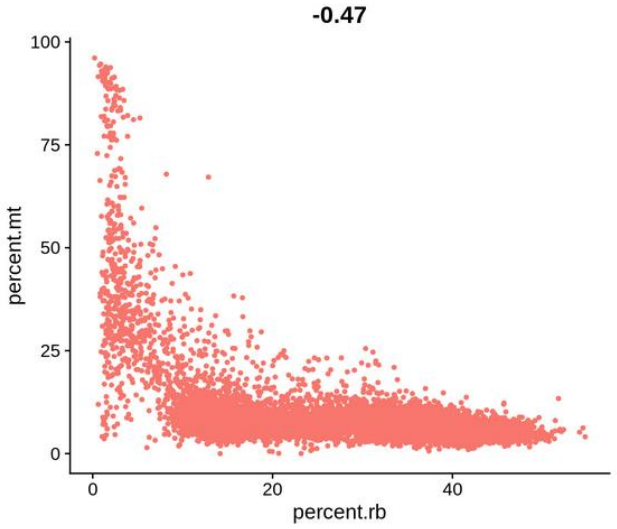
Scatter plots for quality control

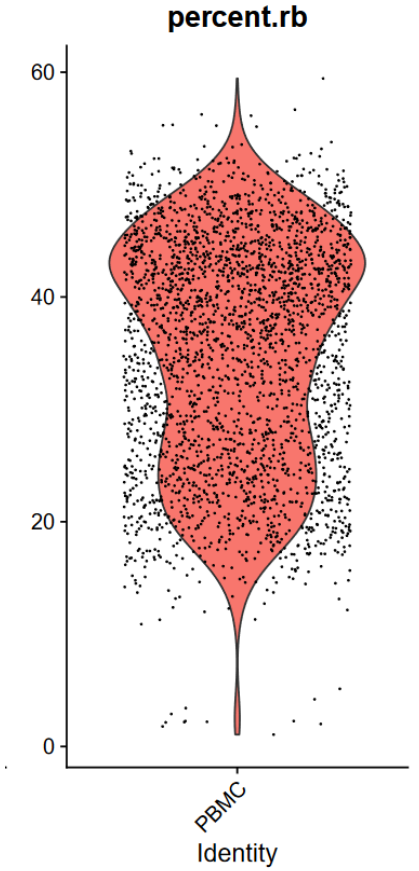
- nCount_RNA vs percent.mt: are there cells with low number of transcripts and high mito%
- nCount_RNA vs nFeature_RNA: these should correlate.



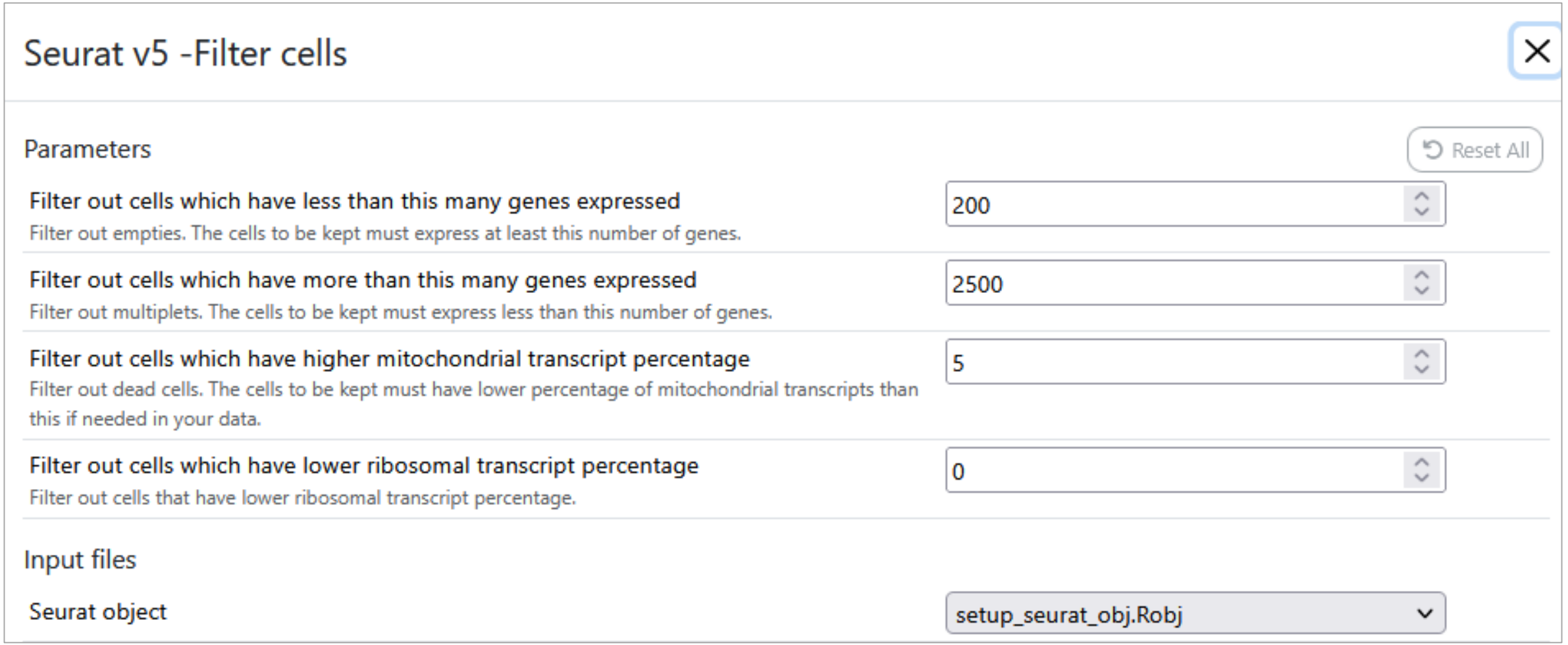
Percentage of UMIs mapping to ribosomal genes (percent.rb)

- Ribosomal transcripts don't have polyA, so they should not be captured
- However, a large proportion of UMIs can be from ribosomal transcripts
  - Percent.rb varies between cells for technical and biological reasons
- Percent.rb and percent.mito *anti*-correlate
  - You can filter out cells which have *lower* ribosomal percentage than x





Parameters for filtering out bad quality cells



Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

What will you learn

1. Why do we need to normalize gene expression values
2. What is a dropout
3. What does global scaling normalization do
4. When does it not work well

Normalizing scRNA-seq gene expression values

- We cluster cells based on differences in their gene expression profiles
- Variance of gene expression values should reflect biological variation across cells
  → We need to remove non-biological variation
- Single-cell gene expression values are noisy
  - Low mRNA content in a cell
  - Variable mRNA capture
  - Variable sequencing depth
- Normalization methods for bulk RNA-seq data don't work for single cell data
  - dropouts = genes whose expression is not detected → lot of zeros

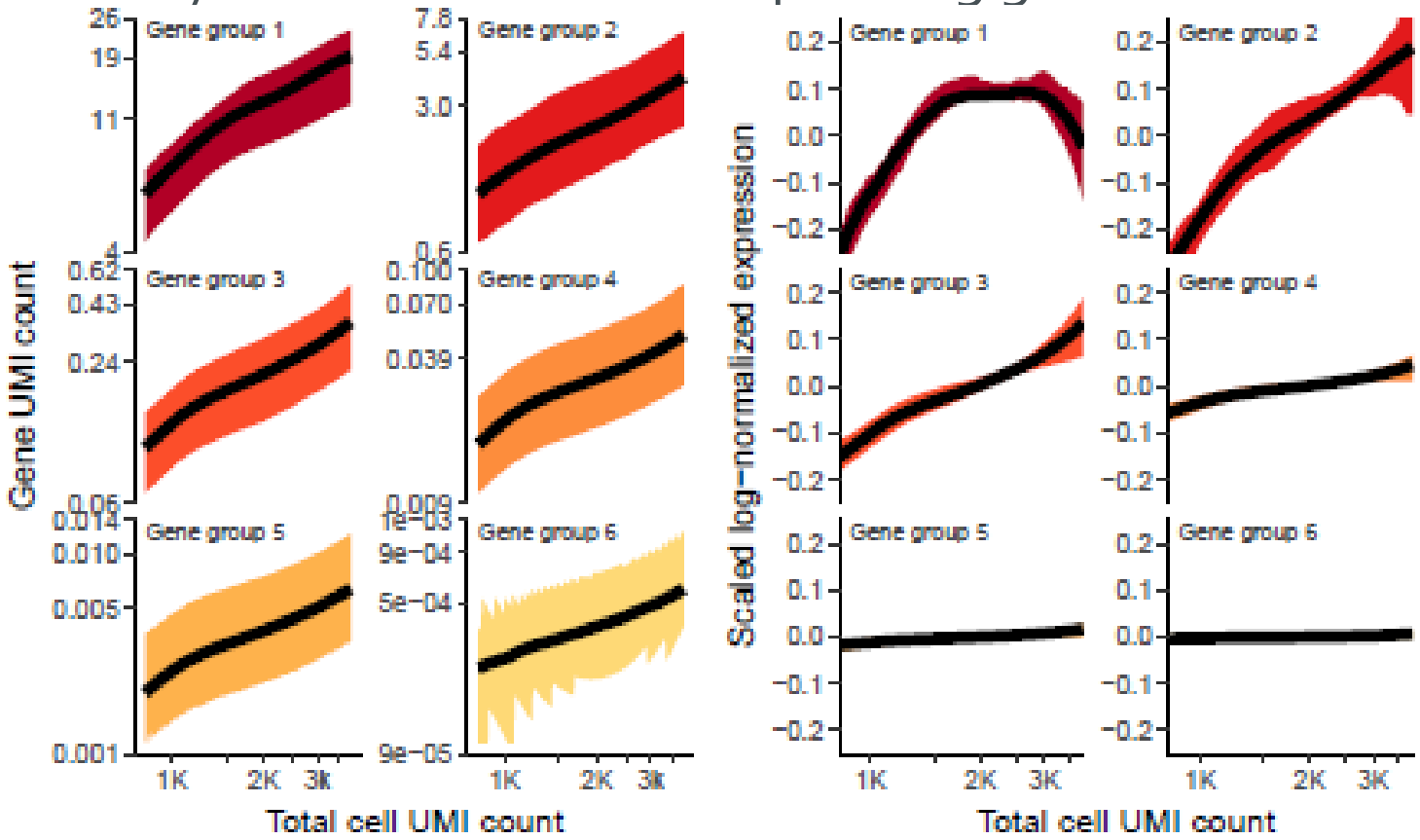
Global scaling normalization

- Divide gene's UMI count in a cell by the total number of UMIs in that cell
- Multiply the ratio by a scale factor (10,000 by default)
  - This scales each cell to this total number of transcripts
- Transform the result by taking natural log

Global scaling normalization: problem with high expressing genes

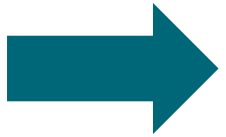
- Sequencing depth (number of UMIs per cell) varies significantly between cells
- Normalized expression values of a gene should be independent of sequencing depth

- The global scaling normalization works only for low to medium expressing genes
  - Expression values of high expressing genes correlate with sequencing depth
  - SCTransform can deal with this better
    - Hafemeister (2019): Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression



SCTransform – alternative approach to normalization etc

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. ~~Normalize expression values~~
4. ~~Identify highly variable genes~~
5. ~~Scale data, regress out unwanted variation~~



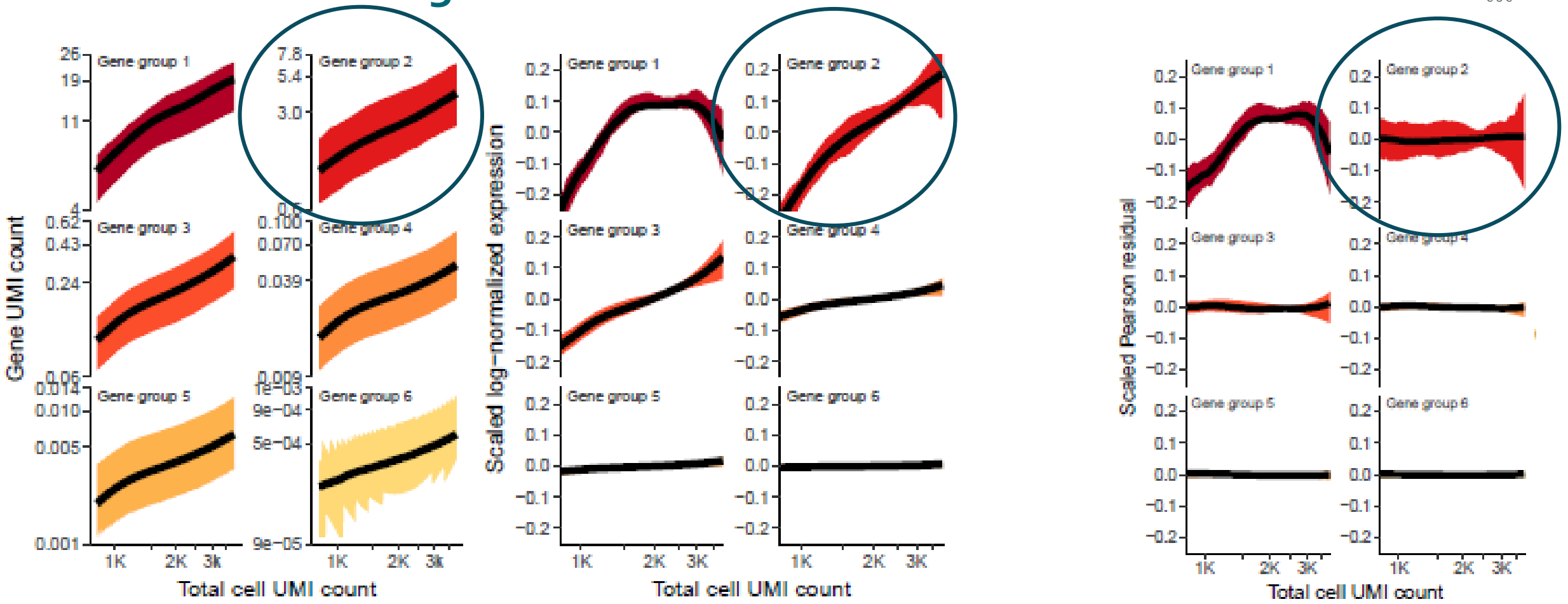
SCTransform

6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

SCTransform: modeling framework for normalization and variance stabilization

- Sequencing depth (number of UMIs per cell) varies significantly between cells
- Normalized expression values of a gene should be independent of sequencing depth
- The default log normalization works ok only for low to medium expressing genes
  - For high expressing genes the normalized expression values correlate with sequencing depth
  - High expressing genes show disproportionally high variance in cells with low sequencing depth
- SCTransform models gene expression as a function of sequencing depth using GLM
  - Constrains the model parameters through regularization, by pooling information across genes which are expressed at similar levels
  - Normalized expression values = Pearson residuals from regularized negative binomial regression
    - Pearson residual = response residual devided by the expected standard deviation (effectively VST)
    - Positive residual for a given gene in a given cell indicate that we observed more UMIs than expected given the gene's average expression in the population and the cellular sequencing depth

Normalization using Pearson residuals works best

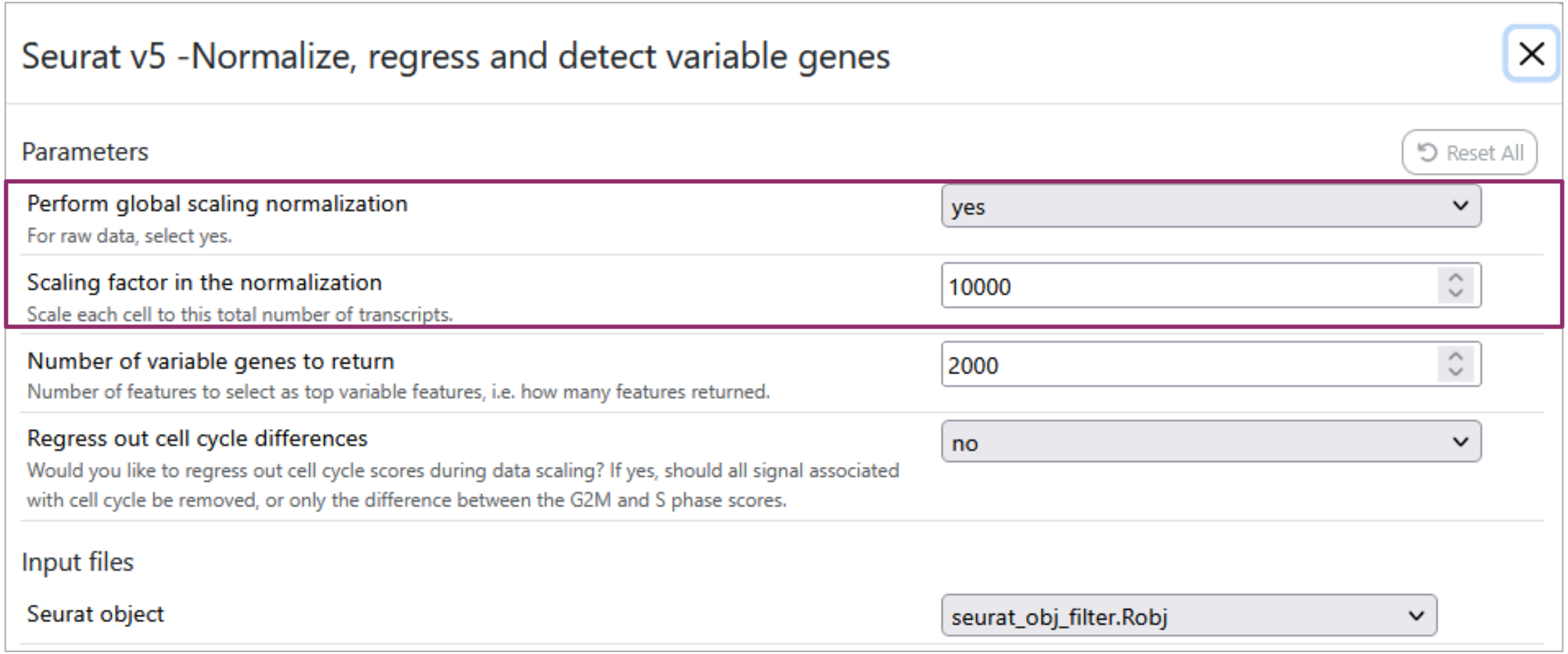


Hafemeister (2019): Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression

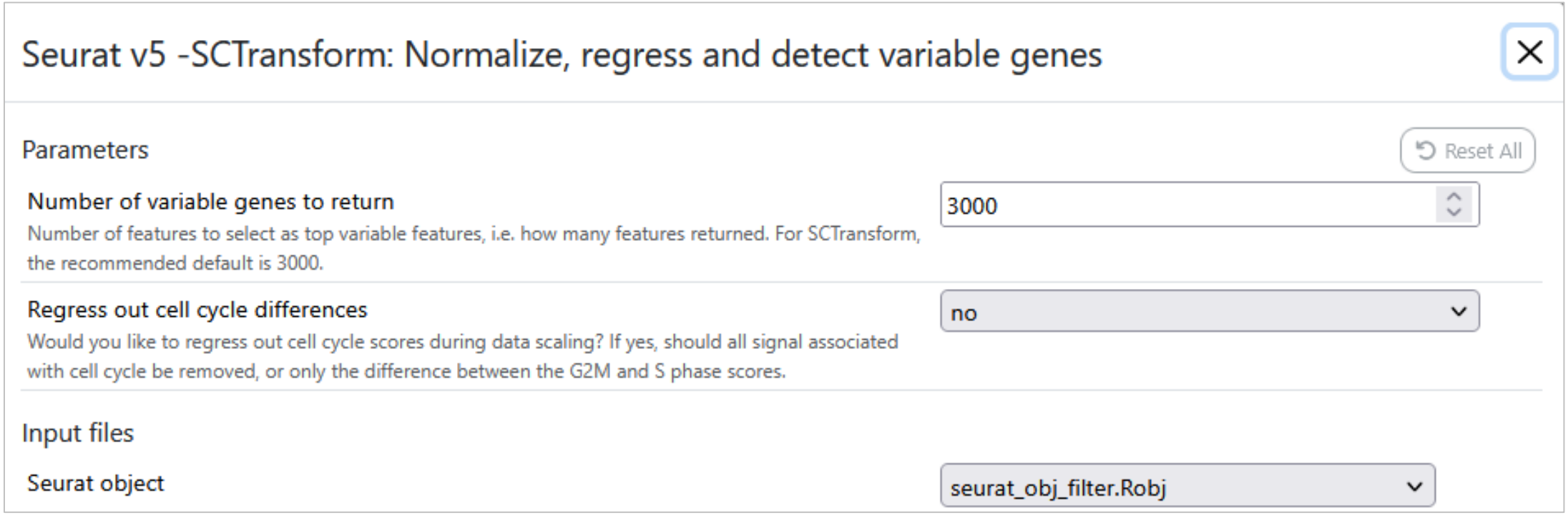
SCTransform: things to take into account in analysis

- When the data is normalized with SCTransform, it is recommended to set
  - In normalization: Number of highly variable genes = 3000 (instead of 2000)
  - In PCA: Number of PCs to compute = 50 (instead of 20)
  - In clustering: Number of principal components to use = 30 (instead of 10), resolution = 0.8 (instead of 0.5)
- Why do we use a different number of highly variable genes and PCs after SCTransform?
  - SCTransform does a better job in normalization (variation in sequencing depth is not a confounding factor any more) → additional variable features are less likely to be driven by technical differences across cells, and instead may represent more subtle biological variability

Parameters for global scaling normalization



Parameters for SCTransform



Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

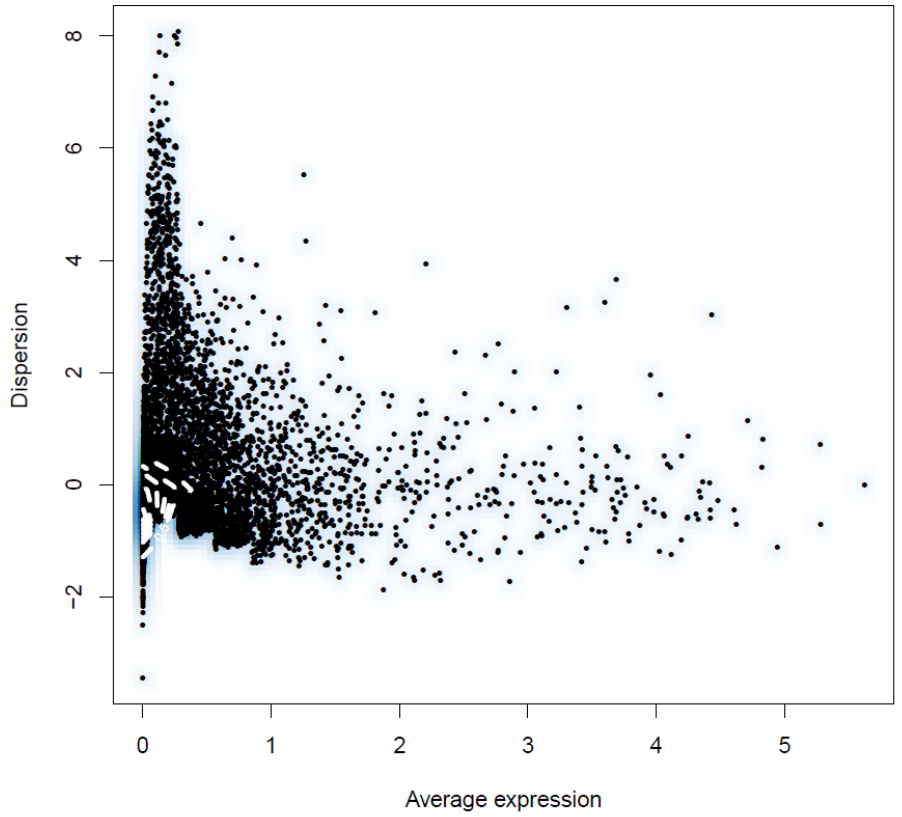
What will you learn

1. Why do we need to find highly variable genes
2. What kind of mean-variance relationship is there in scRNA-seq data
3. Why do we need to stabilize the variance of gene expression values

Selecting highly variable genes

- We want to cluster cells, so we need to find genes whose expression varies across the cells
  - Highly variable genes are used for PCA, and the PCs are used for clustering
- We cannot select genes based on their variance, because scRNA-seq data has strong mean-variance relationship
  - low expressing genes have higher variance

→ variance needs to be stabilized first

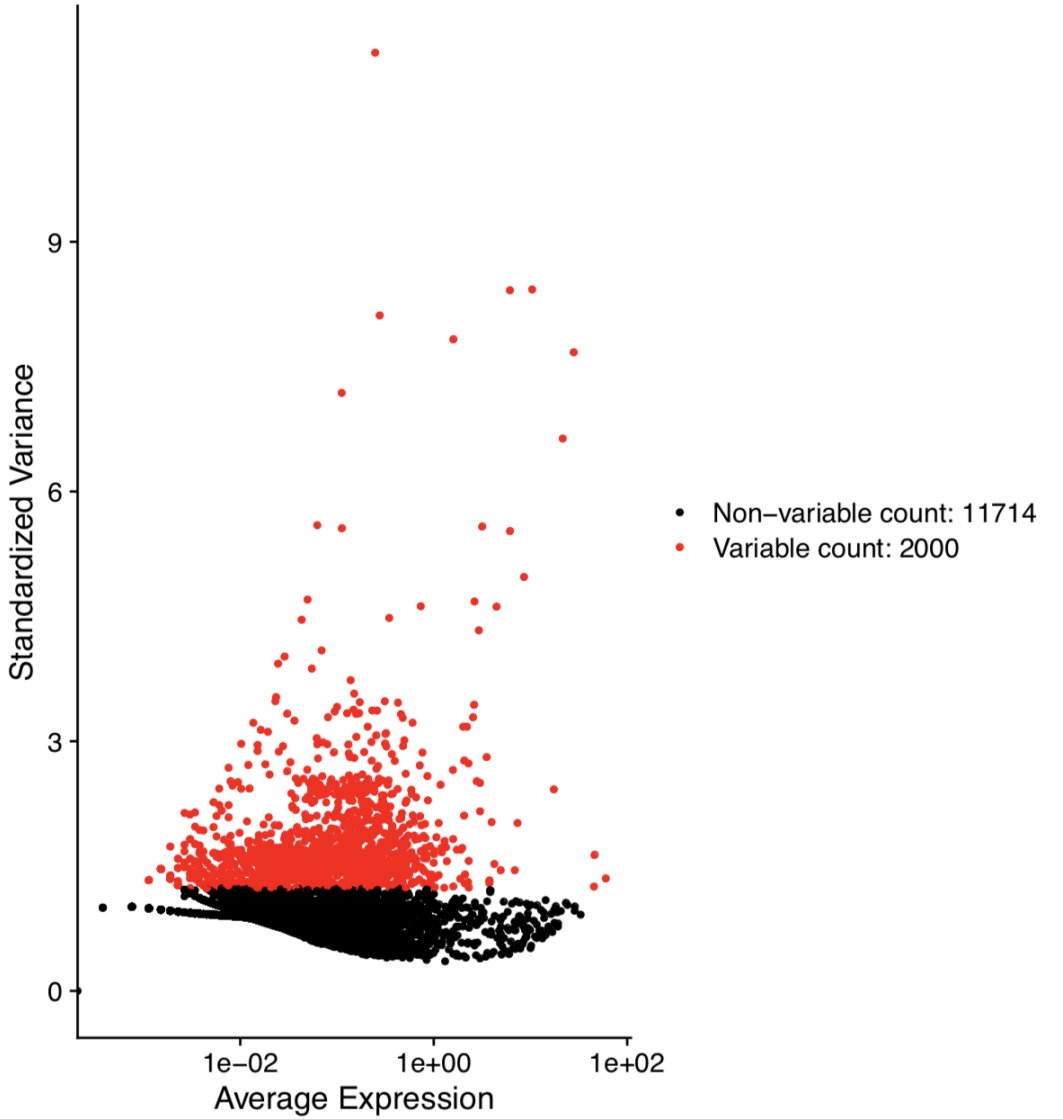


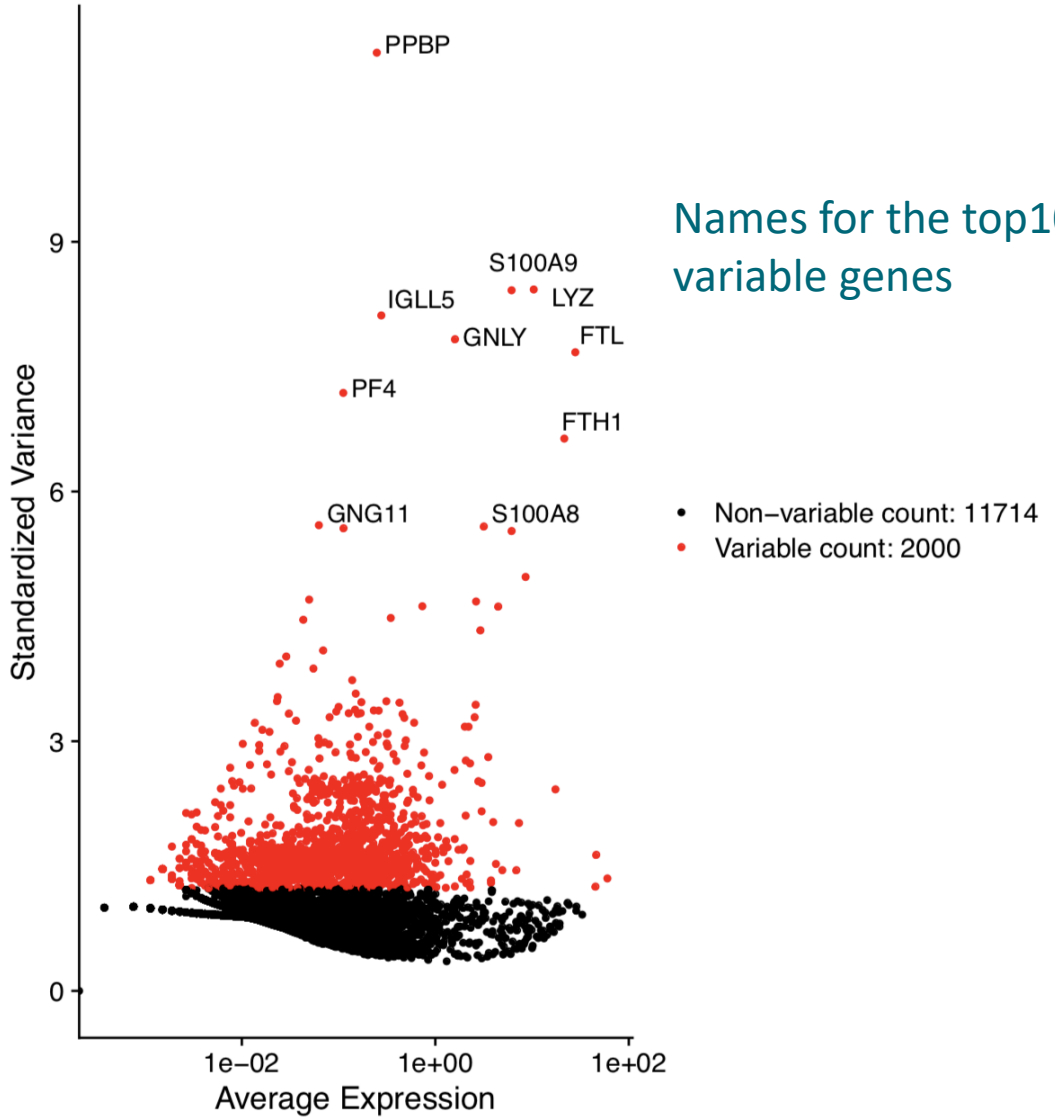
Variance stabilizing transformation (VST)

- Compute the mean and variance for each gene using the unnormalized UMI counts
- Take $\log_{10}$ of mean and variance
- Fit a curve to predict the variance of each gene as a function of its mean expression
- Standardized count = (expression$_{geneXcellY}$ – mean expression$_{geneX}$) / predicted SD$_{geneX}$
  - reduce the impact of technical outliers: set the max of standardized counts to the square root of number of cells
- For each gene, compute the variance of the standardized values across all cells

→ Rank the genes based on their standardized variance and use the top 2000 genes for PCA and clustering

Detection of highly variable genes: plots





Names for the top10 highly variable genes

Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

What will you learn

1. Why do we need to scale data prior to PCA
2. How is scaling done
3. How can we remove unwanted sources of variation

Scaling expression values prior to dimensional reduction

- Standardize expression values for each gene across all cells prior to PCA
  - This gives equal weight in downstream analyses, so that highly expressed genes do not dominate
- Z-score normalization in Seurat's ScaleData function
  - Shifts the expression of each gene, so that the mean expression across cells is 0
  - Scales the expression of each gene, so that the variance across cells is 1
- ScaleData has an option to regress out unwanted sources of variation
  - E.g. cells might cluster according to their cell cycle state rather than cell type

Regress out unwanted sources of variation

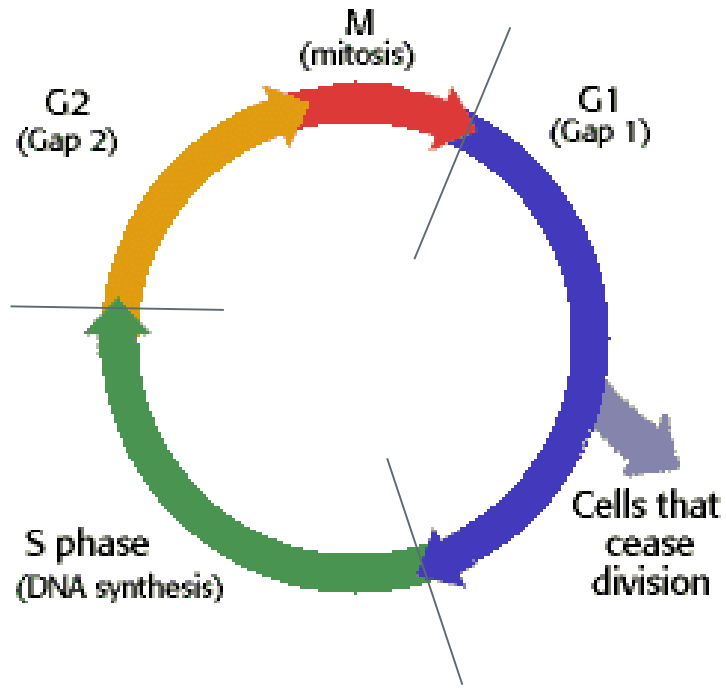
- Several sources of uninteresting variation
  - technical noise
  - batch effects
  - cell cycle stage, etc
- Removing this variation improves downstream analysis
- Seurat constructs linear models to predict gene expression based on user-defined variables
  - number of detected transcripts per cell, mitochondrial transcript percentage, batch,…
  - variables are regressed individually against each gene, and the resulting residuals are scaled and centered
  - scaled z-scored residuals of these models are used for dimensionality reduction and clustering
  - **In Chipster** the following effects are removed:
    - number of detected molecules per cell
    - mitochondrial transcript percentage
    - cell cycle stage (optional)

Mitigating the effects of cell cycle heterogeneity

1. Compute cell cycle phase scores for each cell based on its expression of G2/M and S phase marker genes
    - These markers* are well conserved across tissues and species
    - Cells which do not express markers are considered not cycling, G1

2. Model each gene's relationship between expression and the cell cycle score

3. Two options to regress out the variation caused by different cell cycle stages
    1. Remove ALL signals associated with cell cycle stage
    2. Remove the DIFFERENCE between the G2M and S phase scores.
        - This preserves signals for non-cycling vs cycling cells, only the difference in cell cycle phase amongst the dividing cells are removed. Recommended when studying differentiation processes

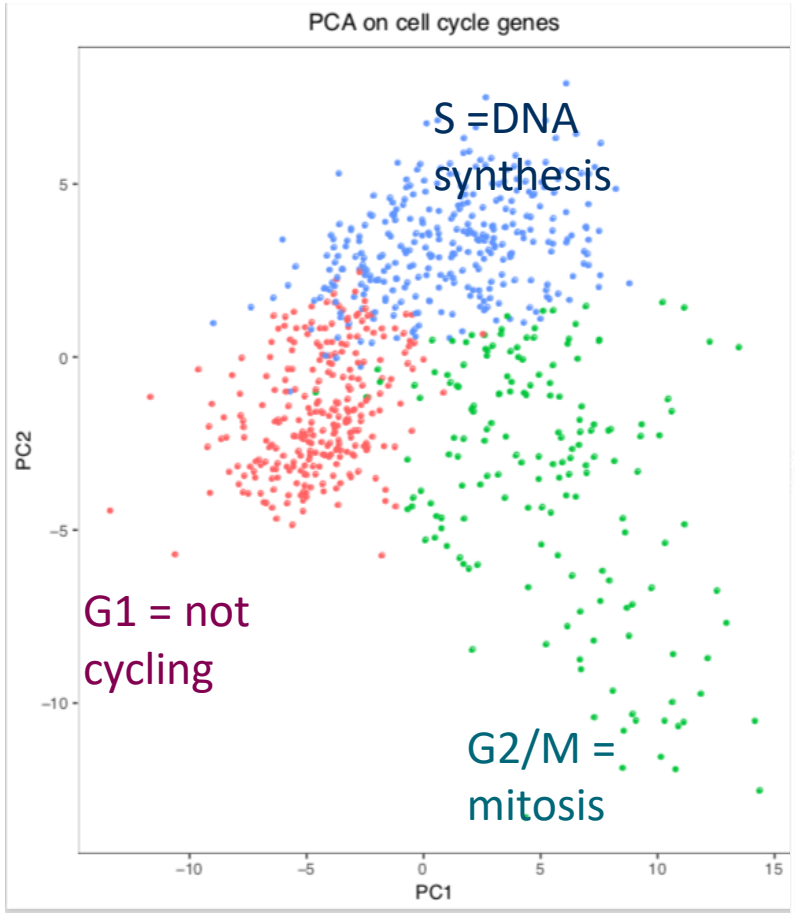


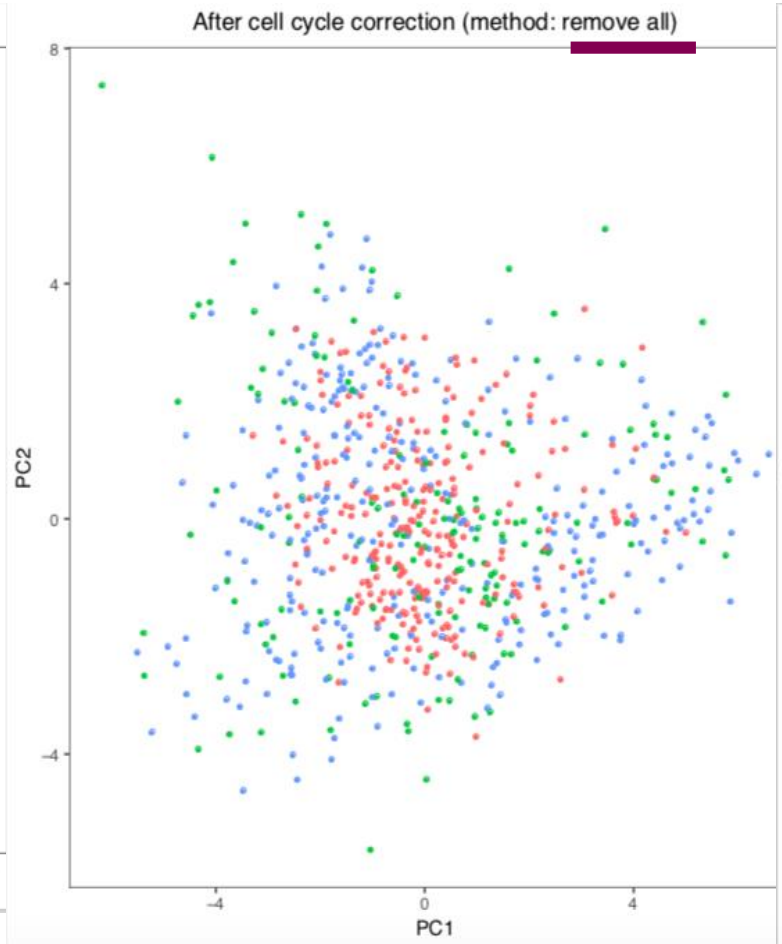
*list of cell cycle markers, from Tirosh et al, 2015
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4944528/

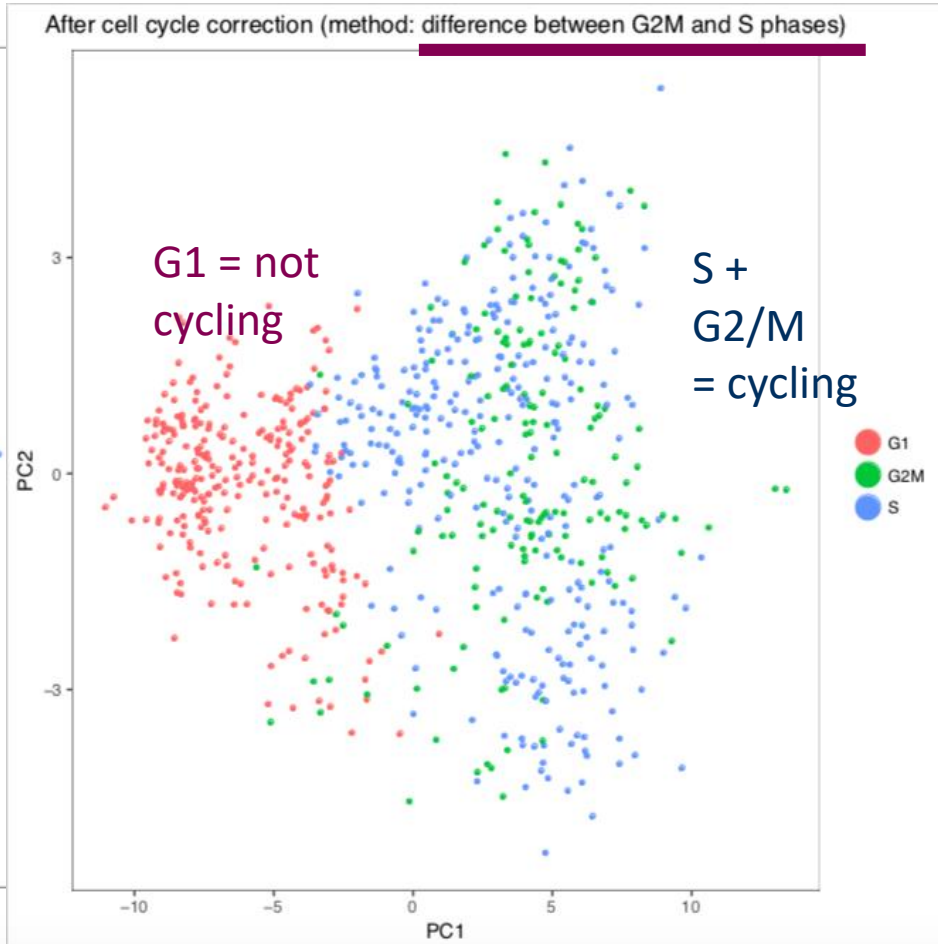
Regressing out the variation caused by different cell cycle stages

When? If we see clear distinction, and the cell differentiation process is not what we are interested in

- OR: perform the clustering without regression and see if we have clusters separated by cell cycle phase. If yes -> come back and perform the regression







PCA on cell cycle genes (dot = cell, colors = phases)

Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

What will you learn

1. Why do we need to do dimensional reduction?
2. How dimensional reduction methods (PCA, tSNE, UMAP) work on intuitive level
3. Why we use both PCA and tSNE/UMAP?
4. How to select the principal components for the clustering step

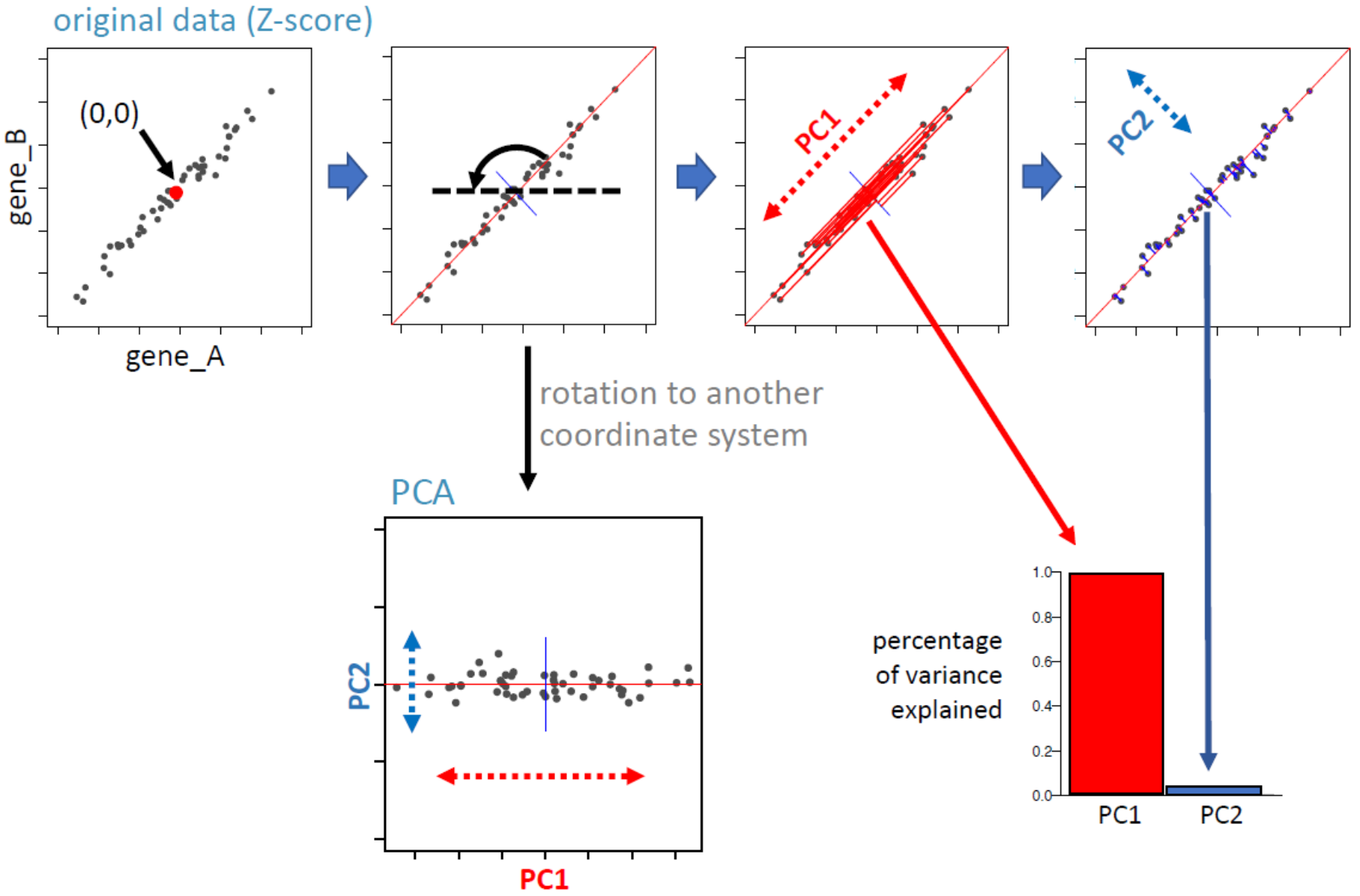
Dimensionality reduction

- What for?
  1. Making clustering step easier (PCA)
  2. Visualization (tSNE, UMAP)
- Simplifies complexity so that the data becomes easier to work with
  - Cells are characterized by the expression values of all the genes → thousands of dimensions
  - We have thousands of genes and cells
- Removes redundancies in the data
  - The expression of many genes is correlated, we don't need so many dimensions to distinguish cell types
- Identifies the most relevant information in order to cluster cells
  - Overcomes the extensive technical noise in scRNA-seq data
- Can be linear (e.g. **PCA**) or non-linear (e.g. **tSNE**, **UMAP**)

Principal Component Analysis (PCA)

- Finds principal components (PCs) of the data
  - Directions where the data is most spread out = where there is most variance
  - PC1 explains most of the variance in the data, then PC2, PC3, …
- We will select the most important PCs and use them for clustering cells
  - Instead of 20 000 genes we have now maybe 10 PCs
  - Essentially, each PC represents a robust 'metagene' that combines information across a correlated gene set
- Prior to PCA we scaled the data so that genes have equal weight in downstream analysis and highly expressed genes don't dominate
  - Shift the expression of every gene so that the mean expression across cells is 0 and the variance across cells is 1.

How PCA works



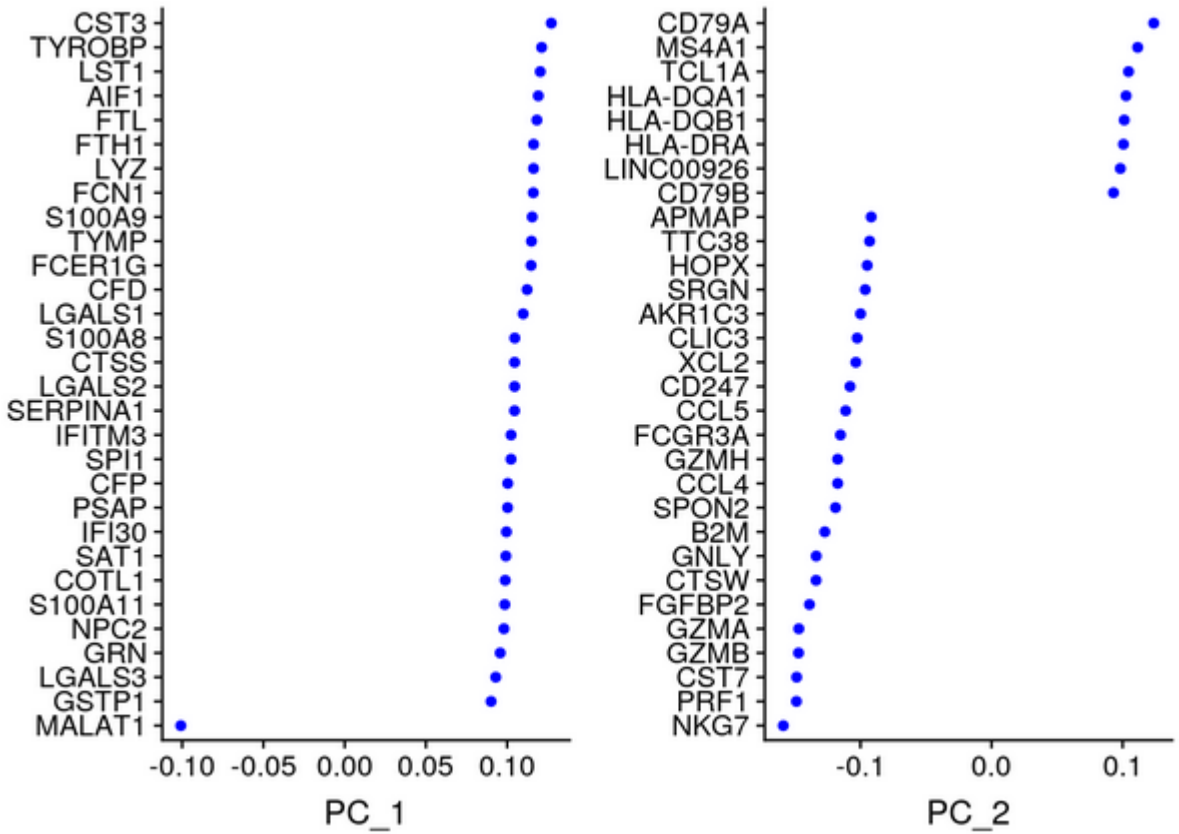
- PC1 explains 98% of the variance
- => PC1 represents these two genes very well
- PC2 is nearly insignificant, and could be disregarded

- In real life, thousands of genes, and maybe tens of PCs

Slide by Paulo Czarnewski

Visualizing PCA results: loadings

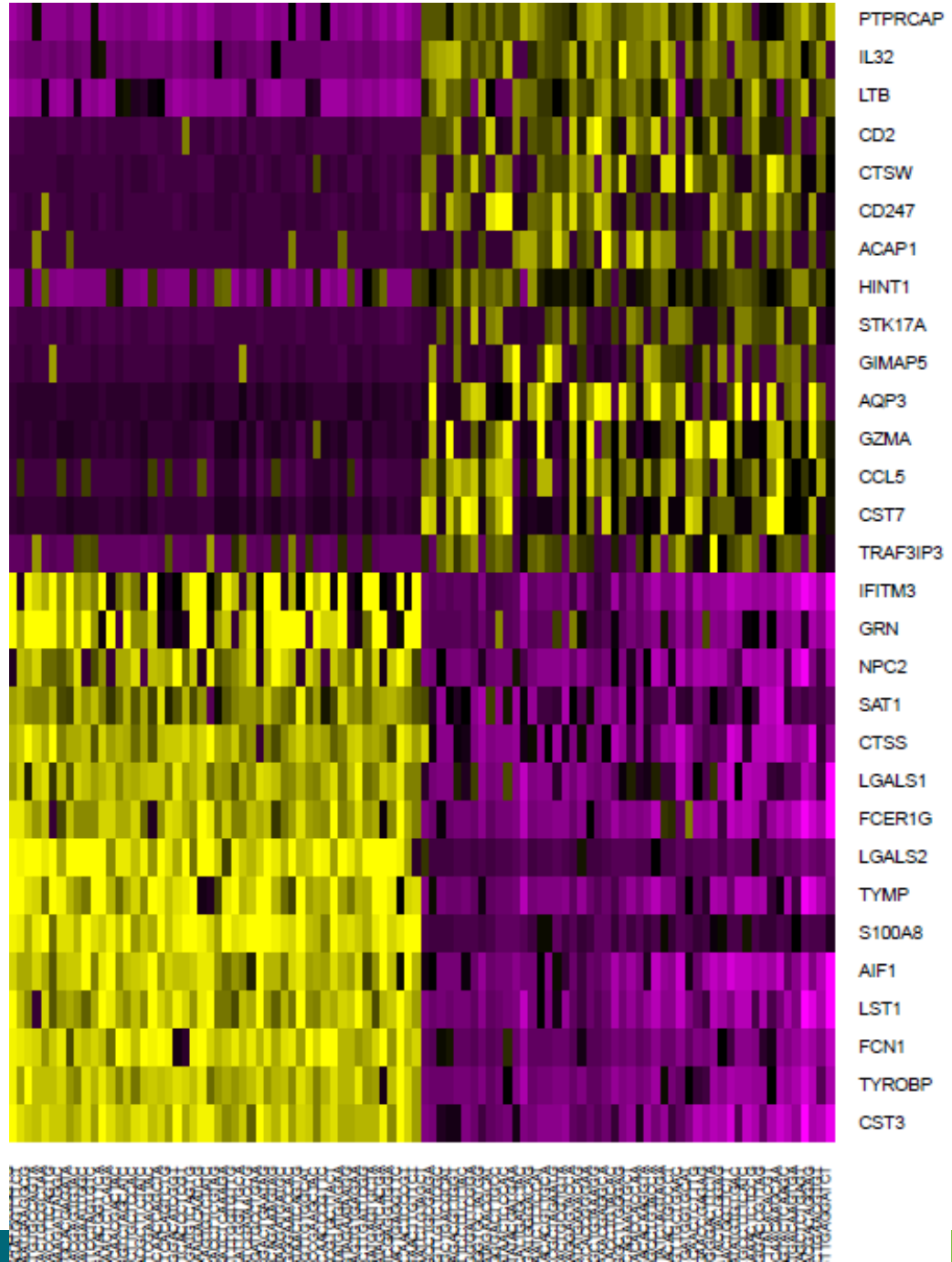
- Visualize top genes associated with principal components
  - = Which genes are important for PC1?)



- Is the correlation direct (positive) or reverse (negative)?
  - Note however, that the signs are arbitrary
  - = if all the variables in a component are positively correlated with each other, all the loadings will be positive
  - if there are some negative correlations among the variables, some of the loadings will be negative

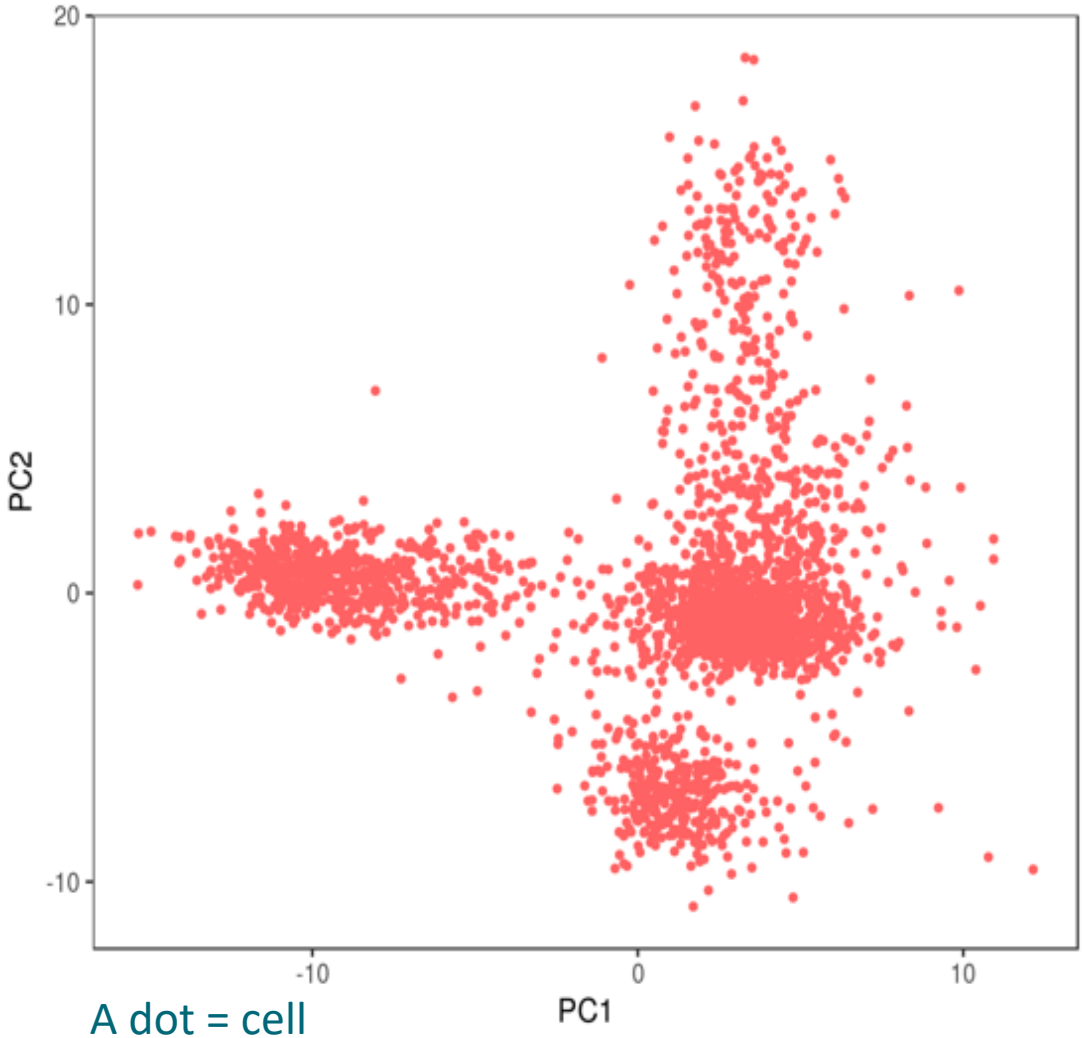
Visualizing PCA results: heatmap

- Which genes correspond to separating cells?
  - Check if there are cell cycle genes
  - Note: after clustering and DE gene analysis steps, we can also eyeball this from the data visualisations
- Both cells and genes are ordered according to their PCA scores. Plots the extreme cells on both ends of the spectrum



Visualizing PCA results: PCA plot

- Gene expression patterns will be captured by PCs → PCA *can* separate cell types
- Note however that PCA can also capture other things, like sequencing depth, cell size or cell heterogeneity/complexity!



A dot = cell

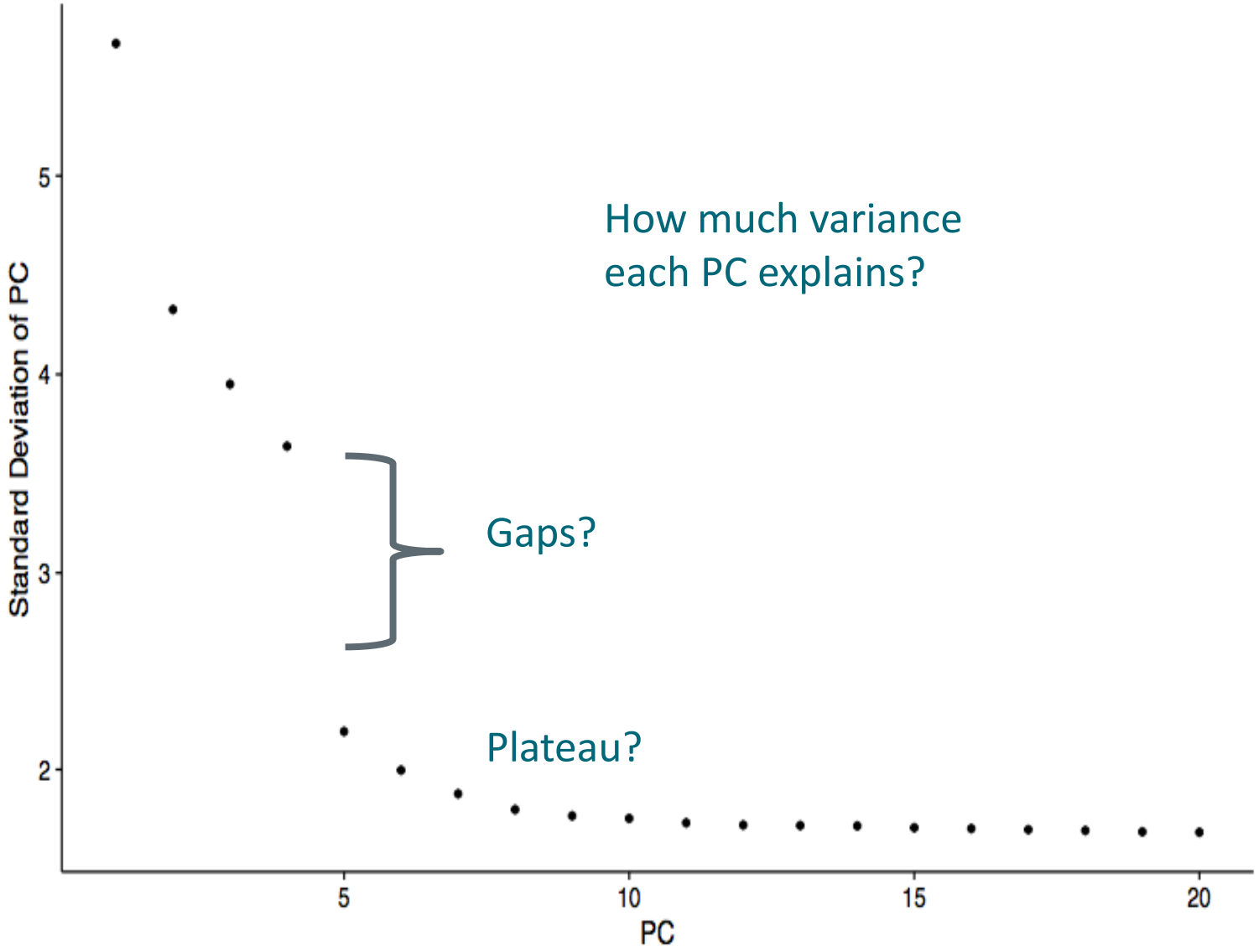
Expression of variable genes used as input data

Determine the significant principal components

- It is important to select the significant PCs for clustering analysis
- However, estimating the true dimensionality of a dataset is challenging
- Seurat developers:
  - Try repeating downstream analyses with a different number of PCs (10, 15, or even 50!).
    - The results often do not differ dramatically.
  - Rather choose higher number.
    - For example, choosing 5 PCs does significantly and adversely affect results
- Chipster provides the following plots to guide you selecting the significant PCs:
  - Elbow plot
  - PC heatmaps

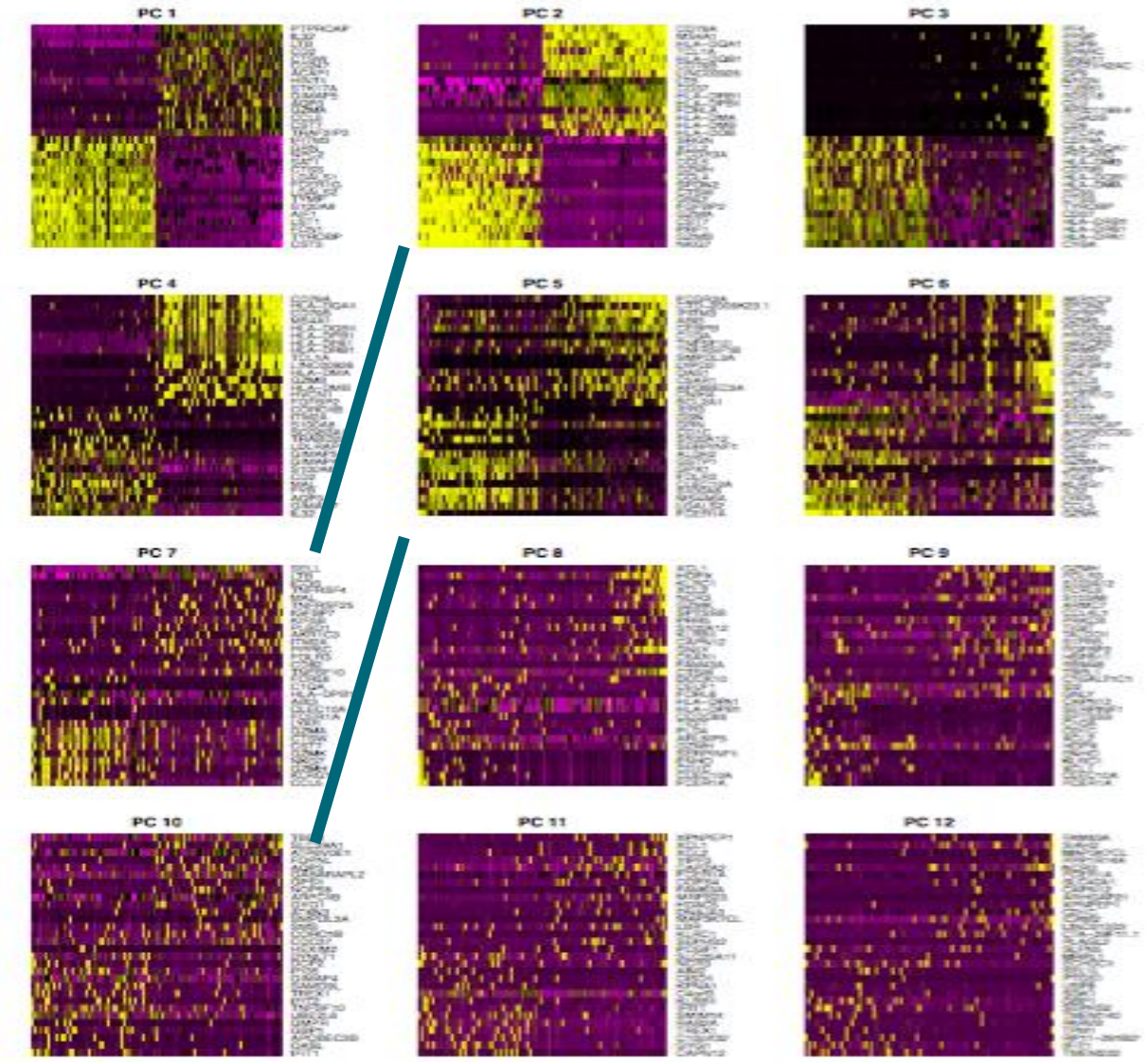
Elbow plot

- The elbow in the plot tends to reflect a transition from informative PCs to those that explain comparatively little variance.



Principal component heatmaps

- Check if there is still a difference between the extremes
- Exclude also PCs that are driven primarily by uninteresting genes (cell cycle, ribosomal or mitochondrial)



Other dimension reduction methods: used later for visualisation

- Graph-based, non-linear methods like tSNE and UMAP
- PCA, tSNE and UMAP available as options in most tools
- We use PCA for dimension reduction before clustering, and tSNE or UMAP for visualisation

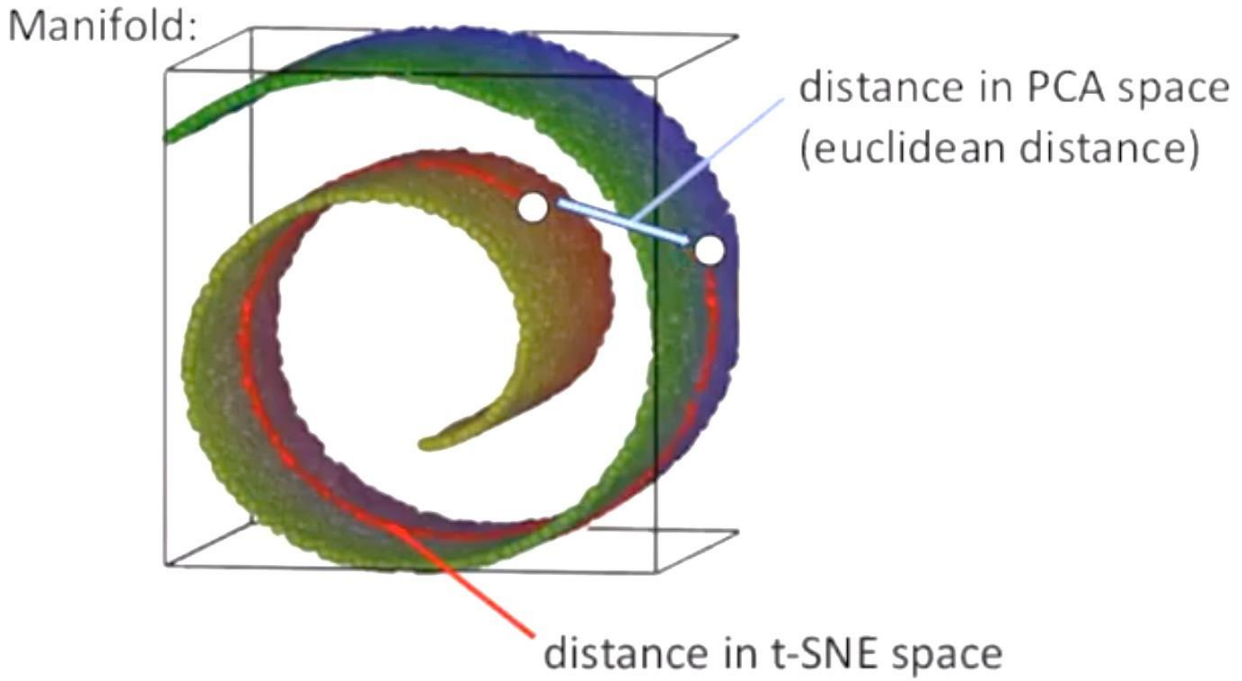
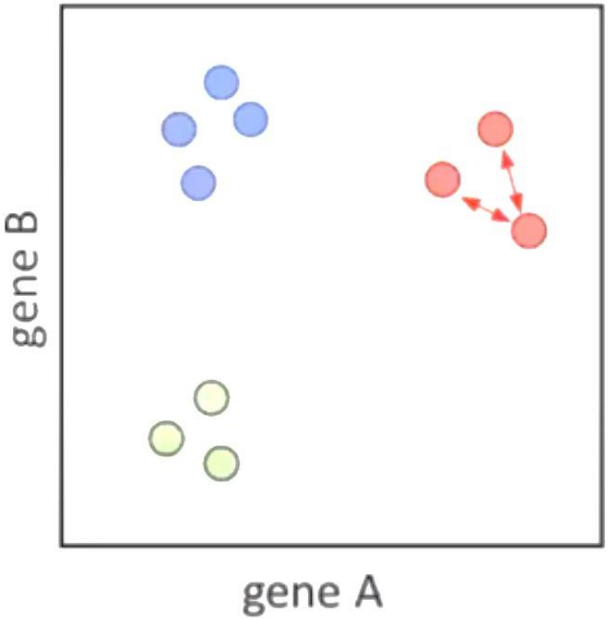


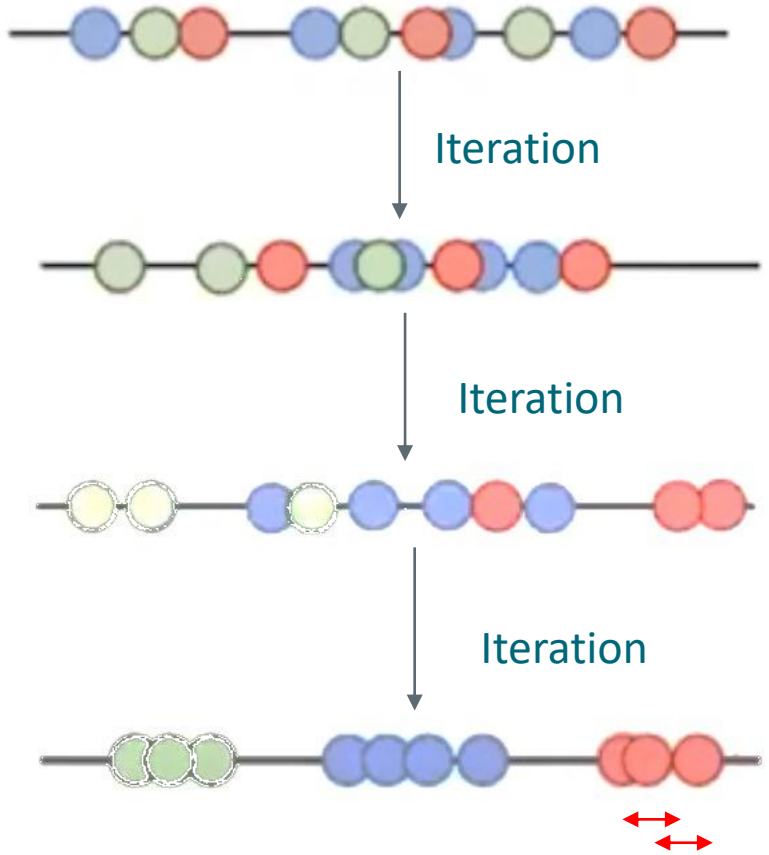
Image by Shigeo Takahashi et al, http://web-ext.u-aizu.ac.jp/~shigeo/research/manifold/

tSNE simplified

- Graph-based
- Non-linear
- Stochastic
- (Only) local distances preserved: distance between groups are not meaningful
- Gold standard
- Can be run on top of PCs
- Many parameters to optimize

Example: From 2D to 1D





Slide modified from Paulo Czarnewski's slides, image based on StatQuest

UMAP

- Non-linear graph-based dimension reduction method like tSNE
- Newer & efficient = fast
- Runs on top of PCs
- Based on topological structures in multidimensional space
- Unlike tSNE, you can compute the structure once (no randomization)
    - => faster
    - => you could add data points without starting over
- **Preserves the global structure** better than tSNE
- More info: video 6 at bit.ly/scRNA-seq
    - Dimensionality reduction explained by Paulo Czarnewski

Analysis steps for clustering cells and finding marker genes

1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

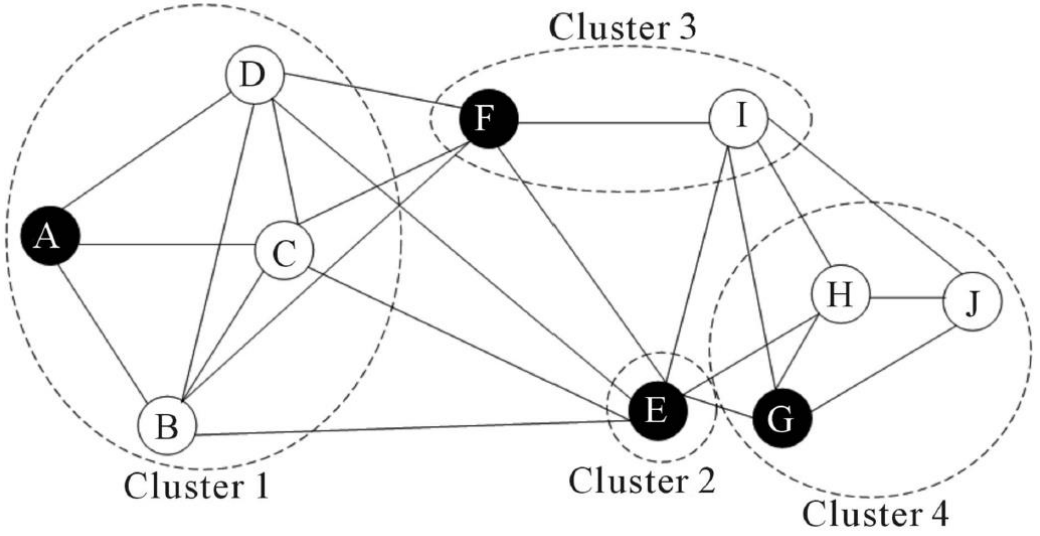
What will you learn

1. Why is clustering a bit complex step?
2. What happens in the clustering step?
3. How to visualise the clusters

Clustering

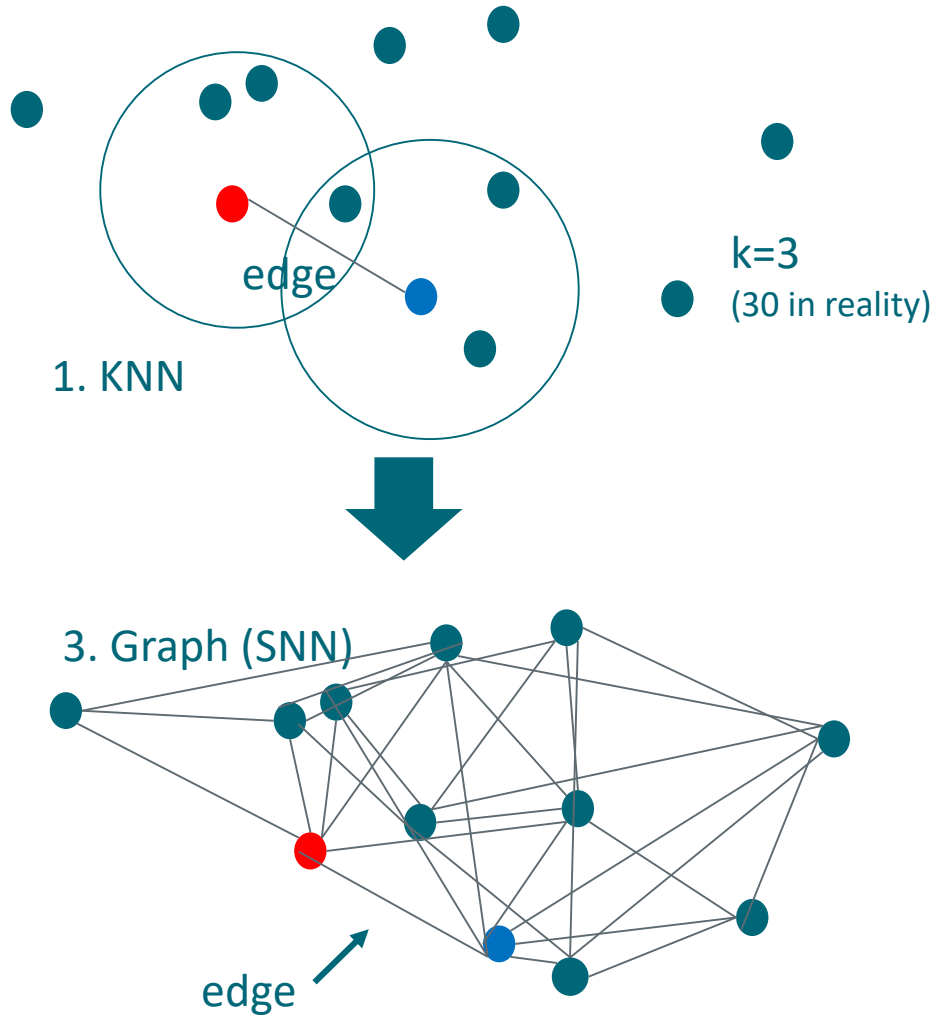
- Divides cells into distinct groups based on gene expression
- Our data is big and complex (lot of cells, genes and noise), so we use principal components instead of genes. We also need a clustering method that can cope with this.
  - →Graph-based clustering
  - →Shared nearest neighbor approach
  - →Graph cuts by Louvain method

Nodes -> cells
Edges -> similarity

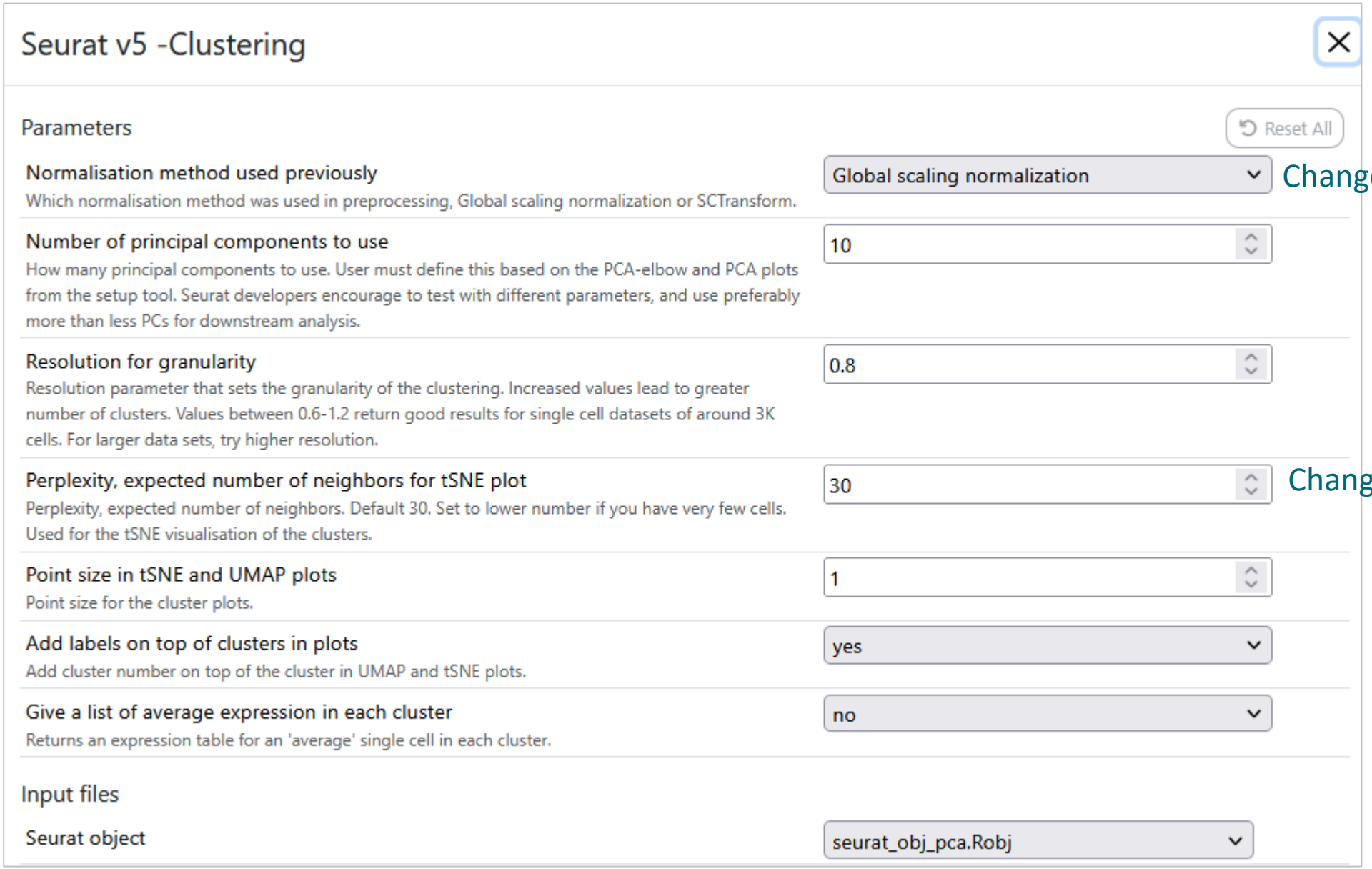


Graph based clustering in Seurat

1. Identify *k nearest neighbours* of each cell
   - Euclidean distance in PC space
2. Rank the neighbours based on distance
3. Build the graph: add an edge between cells if they have a shared nearest neigbour (SNN)
   - Give edge weights based on ranking
4. Cut the graph to subgraphs (clusters) by optimizing modularity
   - *Louvain algorithm* by default



Clustering parameters



Change if you used SCTransform

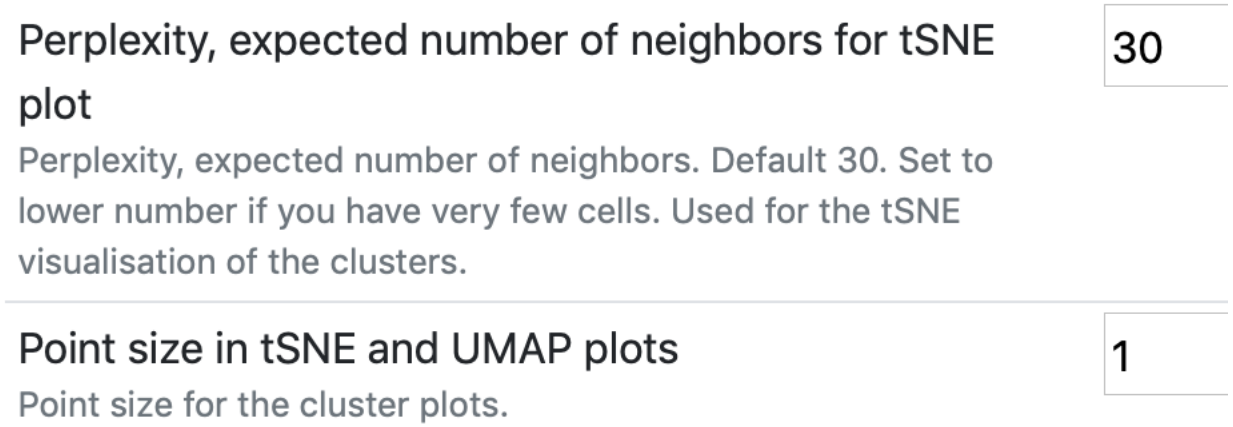
Change if very few cells

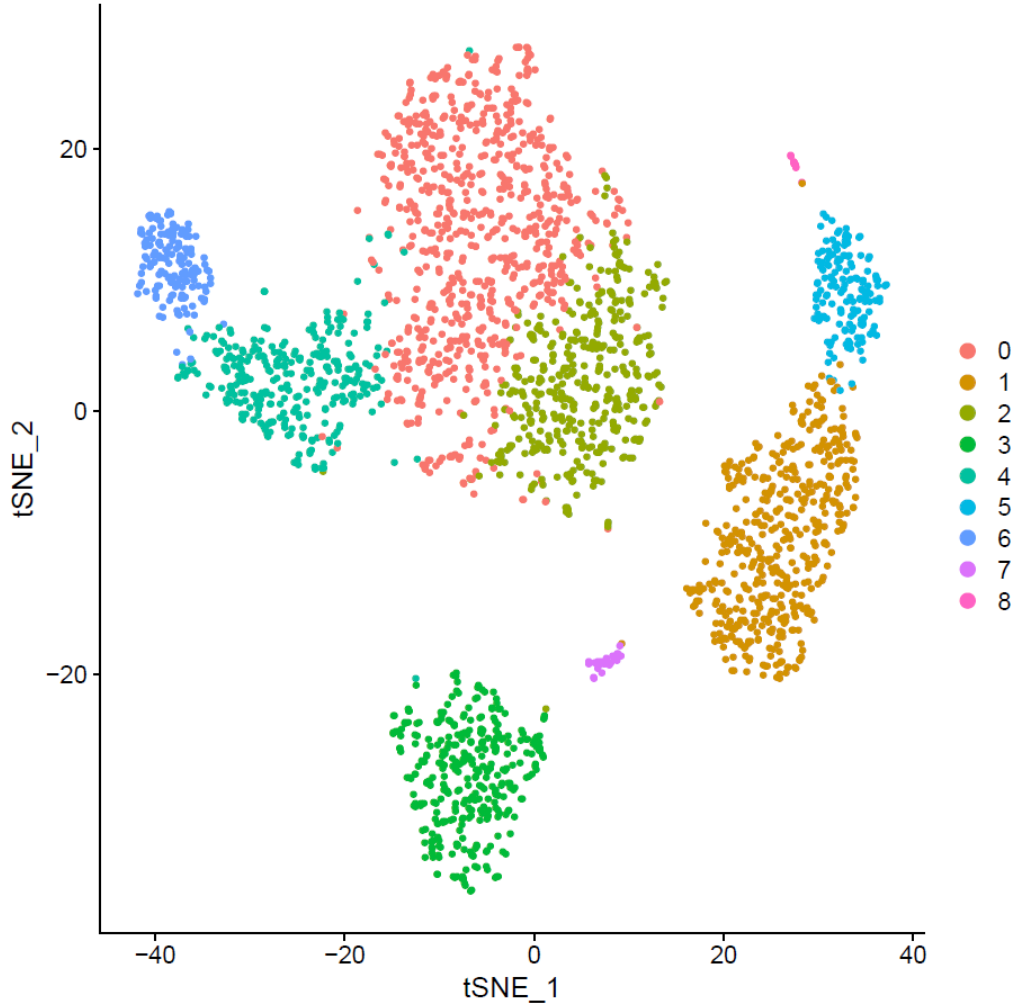
Clustering parameters

- **Number of principal components to use**
    - Experiment with different values
    - If you are not sure, use a higher number
- **Resolution for granularity**
    - Increasing leads to more clusters
    - Values 0.4 - 1.2 can return good results for datasets of around 3000 cells
    - Higher resolution is often optimal for larger datasets

Visualization of clusters: tSNE or UMAP

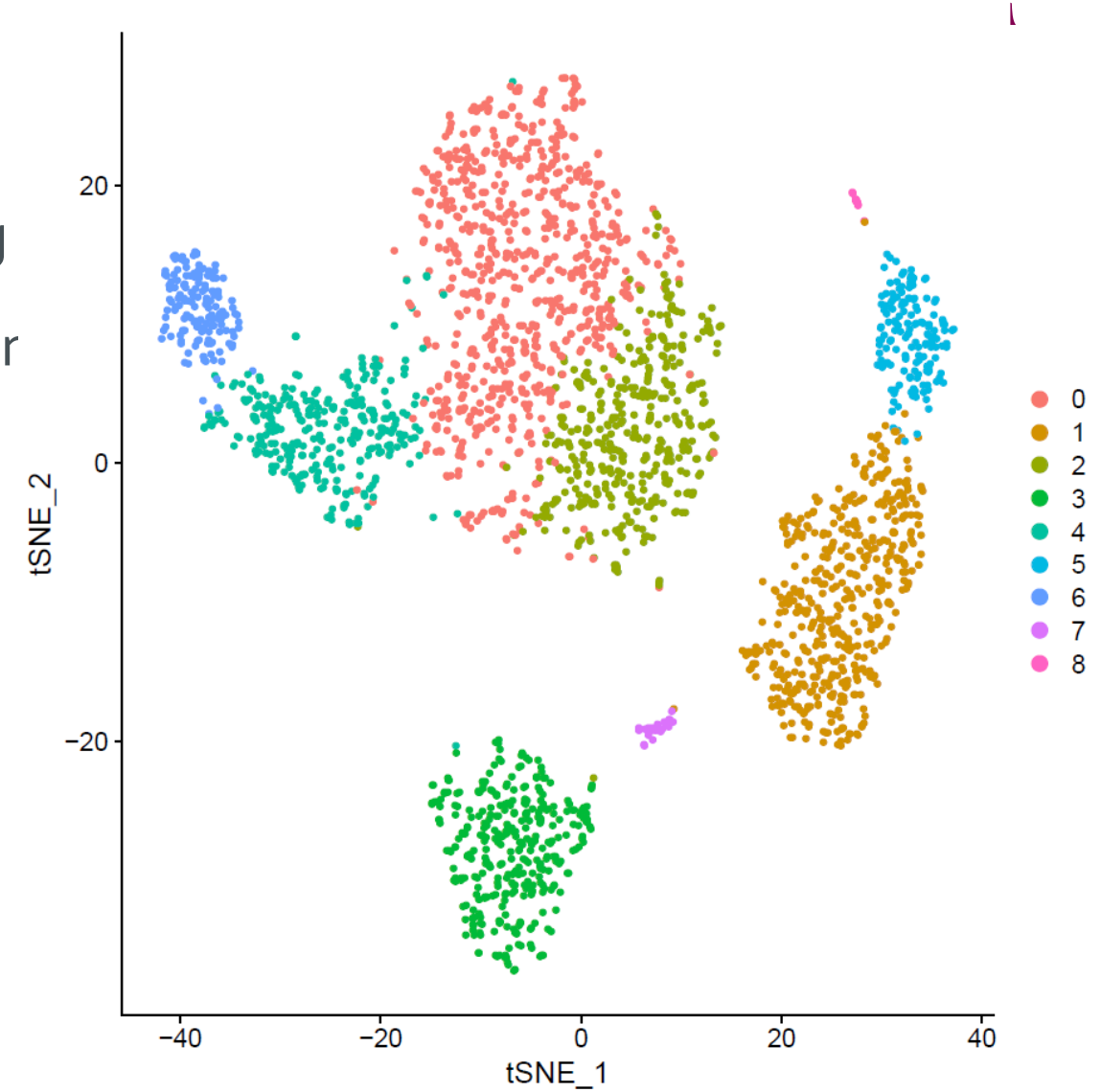
- tSNE/UMAP plot is gray by default, we color it by clustering results from the previous step
  - Check how well the groupings found by tSNE/UMAP match with cluster colors
- Input data: same PCs as for the clustering
- 2 parameters:





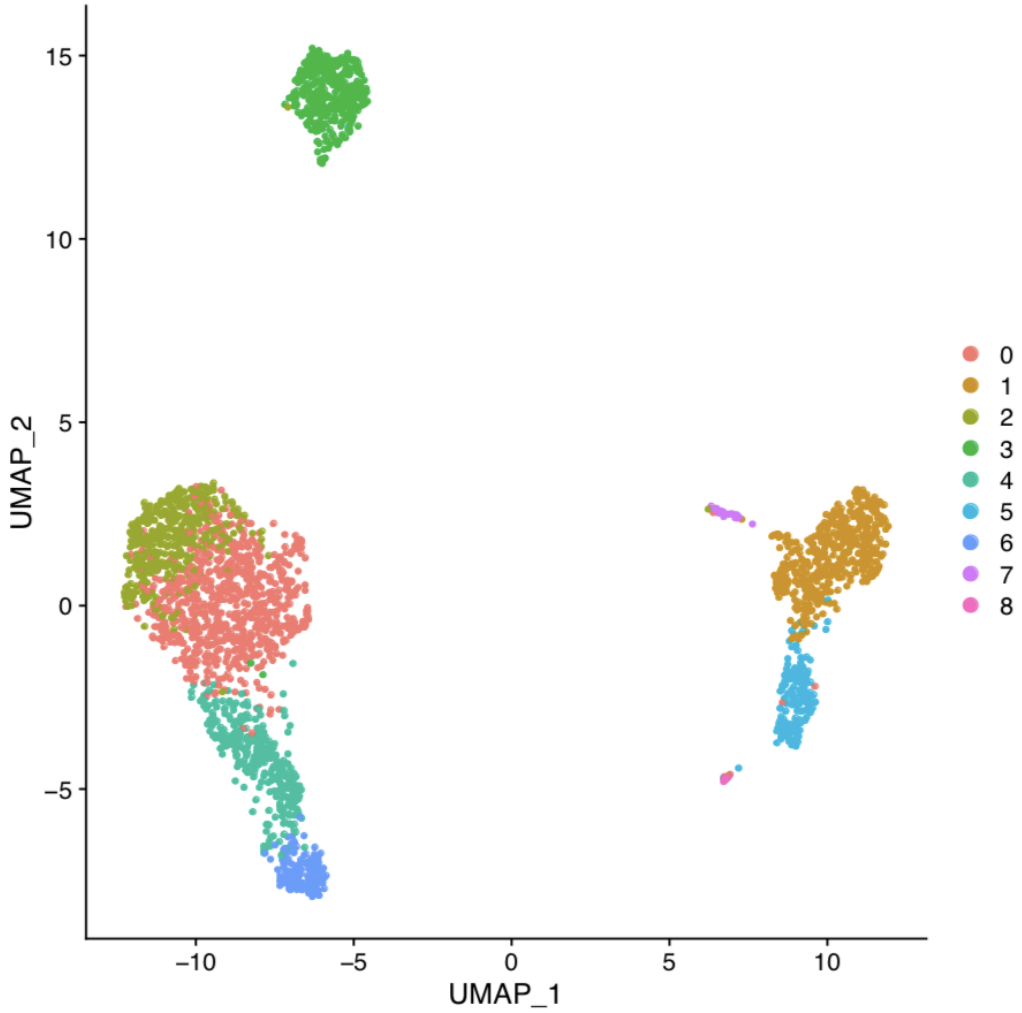
tSNE plot for cluster visualization

- t-distributed Stochastic Neighbor Embedding
- Graph-based non-linear dimensional reductior
  - Different transformations to different regions
- Specialized in local embedding
  - Distance between clusters is not meaningful
  - https://distill.pub/2016/misread-tsne/
- Perplexity = number of neighbors to consider
  - Default 30, lower for small datasets



UMAP plot for cluster visualization

- UMAP = Uniform Manifold Approximation and Projection
- Non-linear graph-based dimension reduction method like tSNE
  - Preserves more of the global structure than tSNE



Analysis steps for clustering cells and finding marker genes

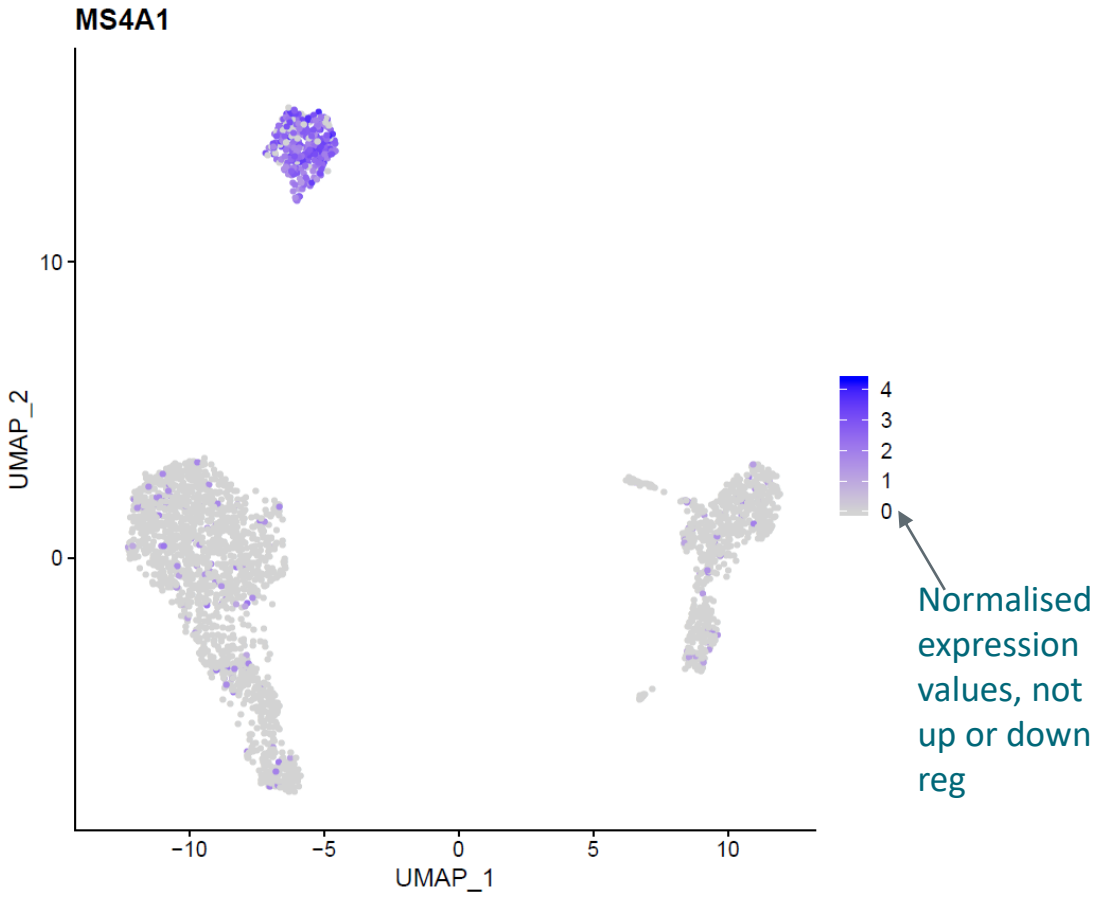
1. Create Seurat object, filter genes, check the quality of cells
2. Filter out low quality cells
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, regress out unwanted variation
6. Reduce dimensions using principal component analysis (PCA) on the variable genes
7. Determine significant principal components (PCs)
8. Use the PCs to cluster cells with graph based clustering
9. Visualize clusters with non-linear dimensional reduction (tSNE or UMAP) using the PCs
10. Detect and visualize marker genes for the clusters

What will you learn

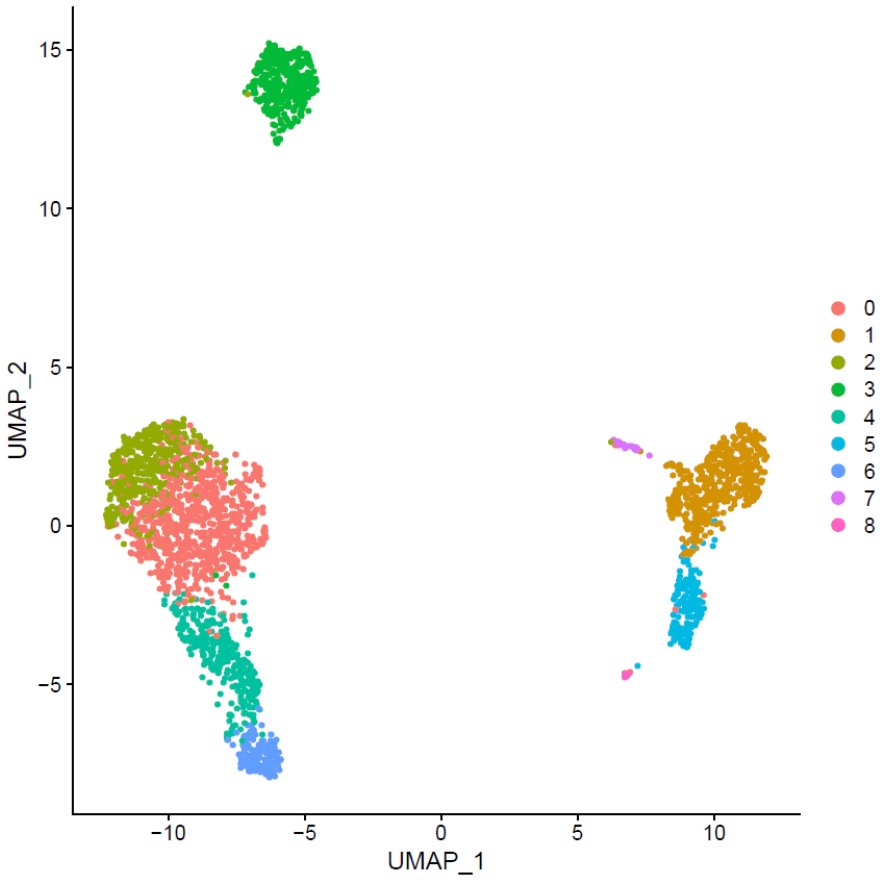
1. What is a marker gene
2. What aspects of scRNA-seq data complicate differential expression analysis
3. Why do we want to filter out genes prior to statistical testing

Marker gene for a cluster

- Differentially expressed between the cluster and all the other cells

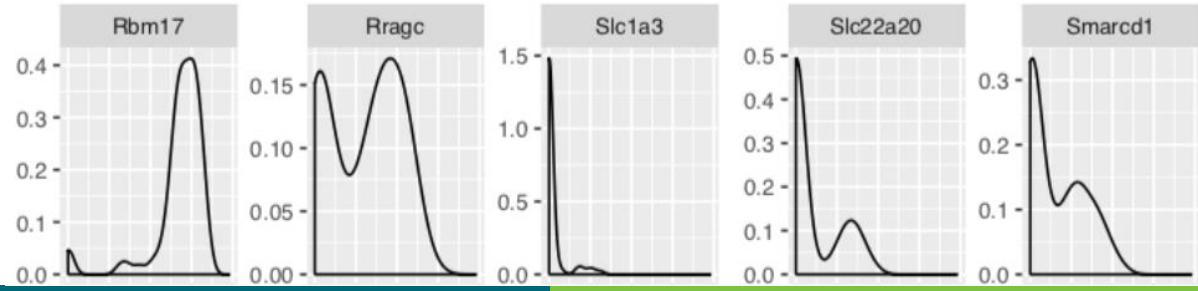


Normalised expression values, not up or down reg



Differential expression analysis of scRNA-seq data

- Challenging because the data is noisy
  - low amount of mRNA → low counts, high dropout rate, amplification biases
  - uneven sequencing depth
- Non-parametric tests, e.g. Wilcoxon rank sum test (Mann-Whitney U test)
  - Can fail in the presence of many tied values, such as the case for dropouts (zeros) in scRNA-seq
- Methods specific for scRNA-seq, e.g. MAST
  - Take advantage of the large number of samples (cells) for each group
  - MAST accounts for stochastic dropouts and bimodal expression distribution
- Methods for bulk RNA-seq, e.g. DESeq2
  - Based on negative binomial distribution, works ok for UMI data.
  - Note: you should *not* filter genes, because DESeq2 models dispersion by borrowing information from other genes with similar expression level
  - Very slow! Use only for comparing 2 clusters



Wilcoxon rank-sum test

Gene A

Cluster1	Rank1	Cluster2	Rank2
21	7	5	1.5
5	1.5	13	6
29	8	6	3
10	5	8	4
	21.5		14.5

values	Rank
5	1.5
5	1.5
6	3
8	4
10	5
13	6
21	7
29	8

$$\text{U-stat} = \text{Rank sum} - \frac{n(n+1)}{2}$$

UA = 21.5 – 4(4+1)/2 = 21.5 – 10 = 11.5

UB = 17.5 – 4(4+1)/2 = 14.5 – 10 = 4.5

U-stat = 4.5 (use the smallest from above)

U-critical = 0 (for alpha=0.05)

U-stat > U-critical (no significant difference)

P-value = 0.342857

Slide by Bishwa Ghimire

Filtering out genes prior to statistical testing – why?

- We test thousands of genes, so it is possible that we get good p-values just by chance (false positives)

→ Multiple testing correction of p-values is needed

  - The amount of correction depends on the number of tests (= genes)
  - Bonferroni correction: adjusted p-value = raw p-value * number of genes tested
  - If we test less genes, the correction is less harsh → better p-values

- Filtering also speeds up testing

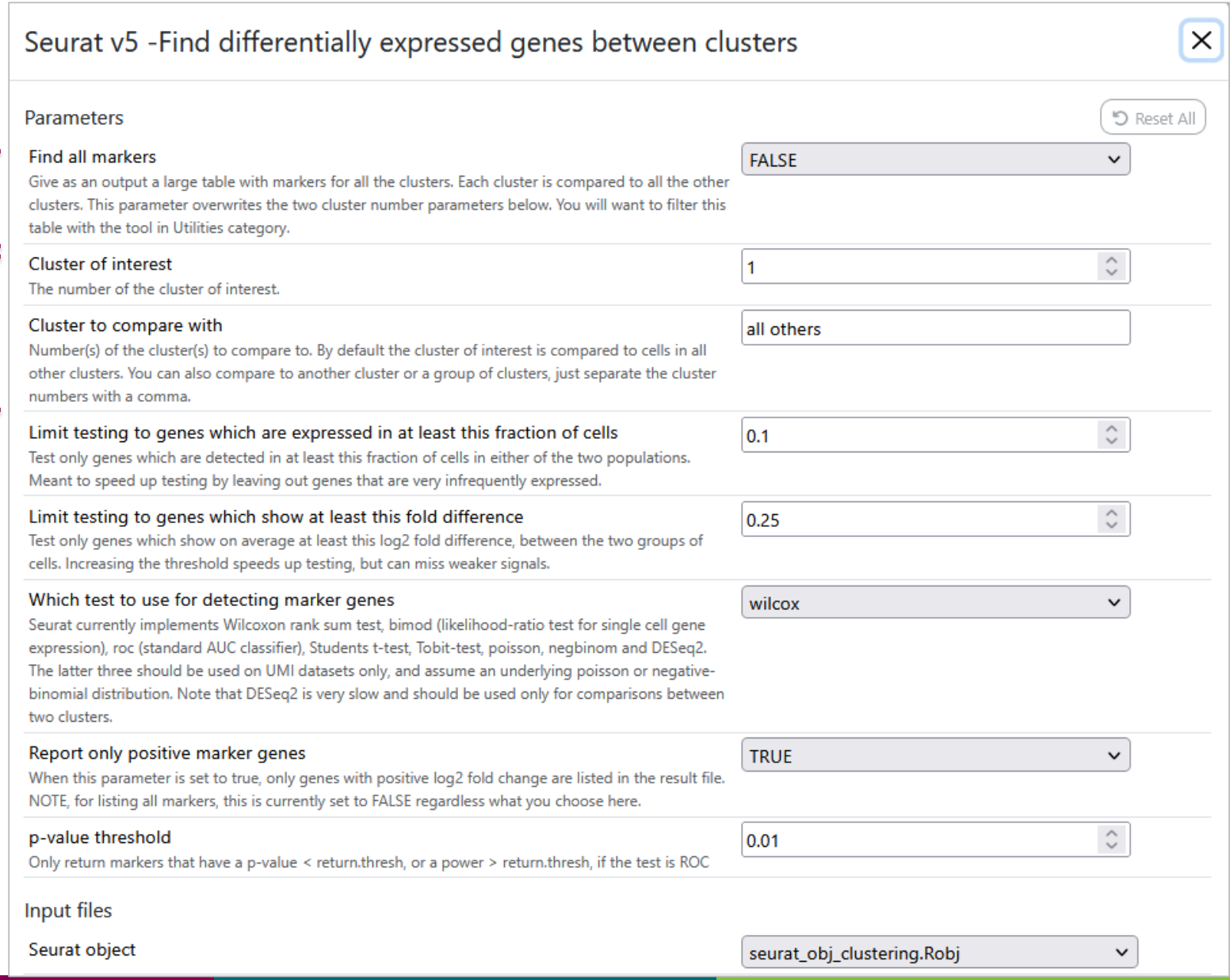
Detect cluster marker genes

Find all markers = every cluster is compared to all the other cells

OR

Compare the cluster of interest to all other cells or to another cluster

- Limit testing to genes which
  - are expressed in at least this fraction of cells in either of the two groups (default 10%)
  - show at least this log2 fold change between the two groups (default 0.25)



Cluster marker gene result table

- p_val = p-value
- p_val_adj = p-value adjusted using the Bonferroni method
- avg_logFC = $\log_2$ fold change between the groups
- cluster = cluster number
- pct1 = percentage of cells where the gene is detected in the first group

markers.tsv •••

Spreadsheet | Text | Details

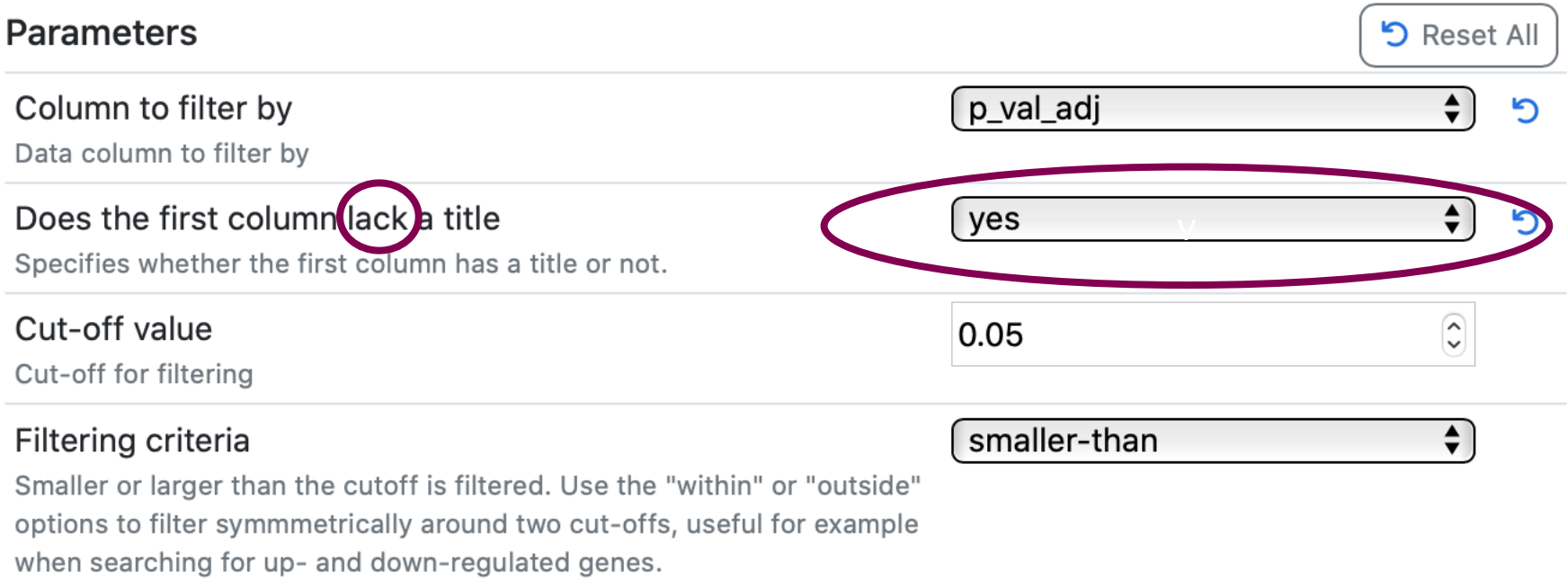
Showing the first 100 of 477 rows. View in full screen to see all rows.

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj	cluster	gene
LTB	6.219516e-123	1.348424	0.958	0.600	8.529445e-119	0	LTB
IL32	3.455676e-113	1.186582	0.893	0.413	4.739115e-109	0	IL32
LDHB	1.155309e-111	1.059929	0.913	0.578	1.584391e-107	0	LDHB
CD3D	1.235473e-109	1.113704	0.872	0.376	1.694328e-105	0	CD3D
IL7R	2.138245e-94	1.278283	0.699	0.281	2.932389e-90	0	IL7R
CD2	1.398161e-60	1.141928	0.551	0.223	1.917438e-56	0	CD2
S100A9	0.000000e+00	5.563093	0.996	0.216	0.000000e+00	1	S100A9
S100A8	0.000000e+00	5.482122	0.973	0.122	0.000000e+00	1	S100A8
LGALS2	0.000000e+00	3.804741	0.908	0.060	0.000000e+00	1	LGALS2
FCN1	0.000000e+00	3.390813	0.952	0.151	0.000000e+00	1	FCN1

You can export these tables and handle them in Excel. Note that due to how R deals with tables (the empty first column name), the column names will be shifted one step to the left. Luckily it is easy to move them to right spot ☺

How to filter the gene list?

- You can filter the result table for example based on the adjusted p-value column using the tool **Utilities / Filter table by column value** using the following parameters:

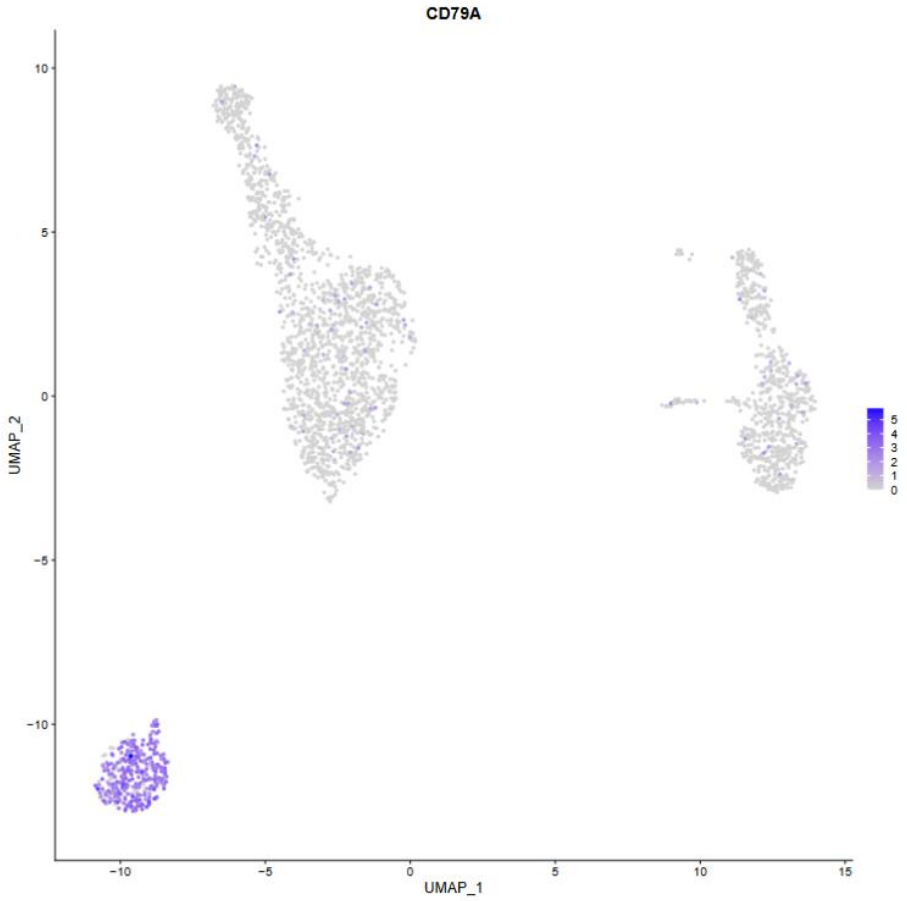


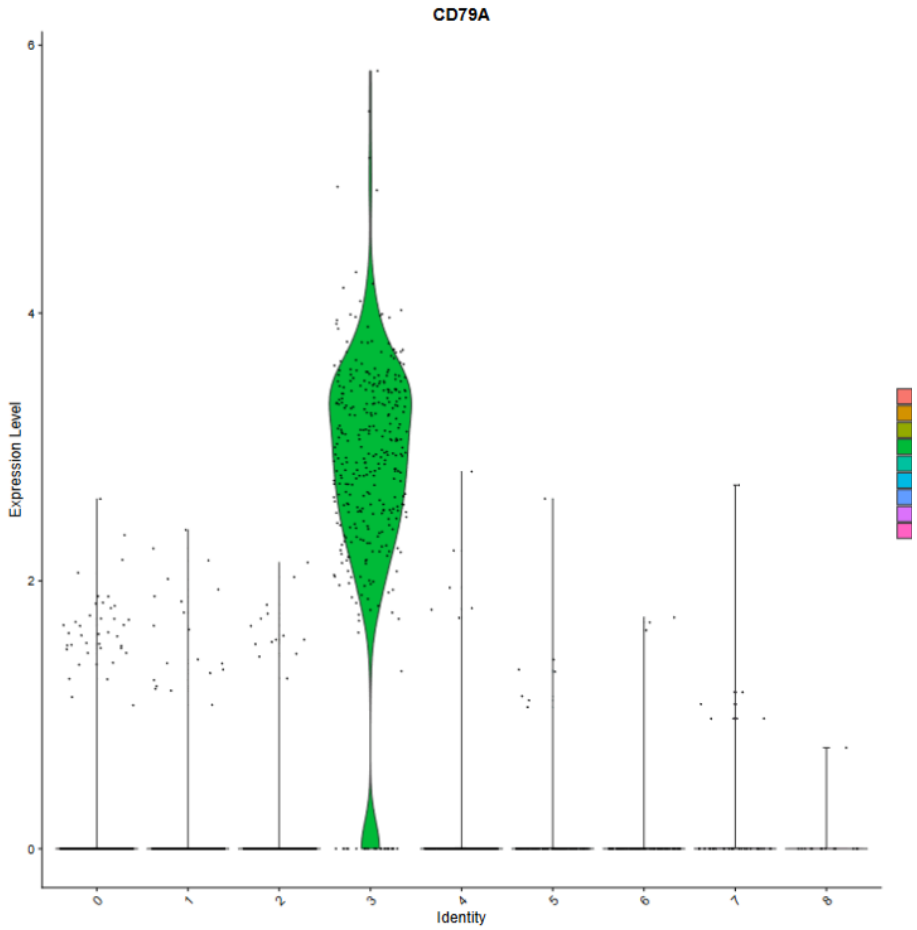
How to retrieve marker genes for a particular cluster?

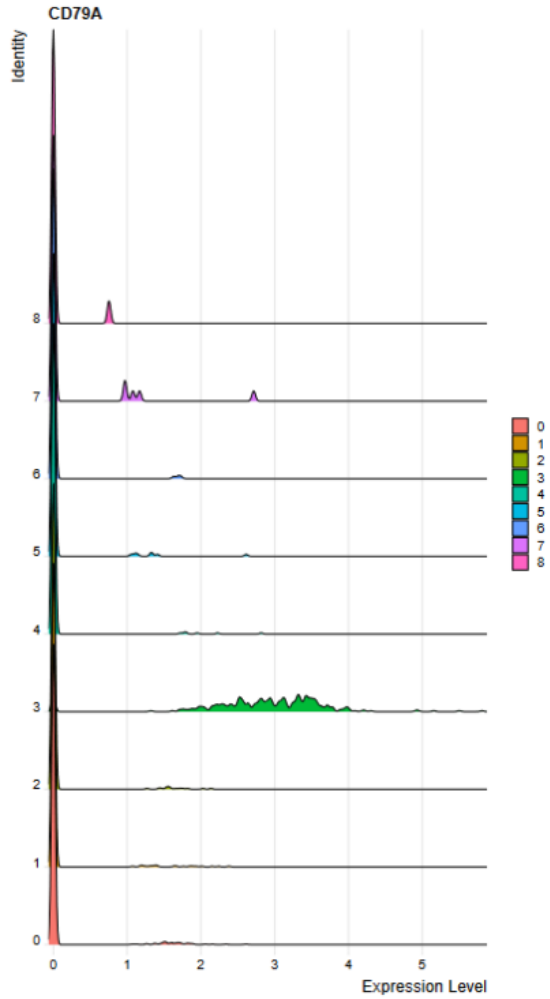
- If you had set Find all markers = TRUE, the result table contains marker genes for all the clusters
- You can filter the result table based on the cluster column using the tool **Utilities / Filter table by column value** using the following parameters

Visualize cluster marker genes

- UMAP, tSNE or PCA plot colored with marker gene expression
- Violin plot
- Ridge plot







Parameters for visualizing genes

Seurat v5 -Visualize genes

Parameters

Reset All

Gene name(s)
Name(s) of the biomarker gene to plot. If you list multiple gene names, use comma (,) as separator.
MS4A1, LYZ

Normalisation method used previously
Which normalisation method was used in preprocessing, Global scaling normalization (default, NormalizeData function used) or SCTransform.
Global scaling normalization

Feature plot visualisation with tSNE, UMAP or PCA
Which dimensionality reduction plot to use.
UMAP

Point size in feature plot
Point size in the UMAP, tSNE or PCA feature plot.
1

Add labels on top of clusters in feature plot
Add cluster numbers on top of clusters in the feature plot.
no

Plotting order of cells based on expression in feature plot
Plot cells in the the order of expression in the feature plot. Can be useful to turn this on if cells expressing given feature are getting buried.
no

Determine color scale based on all features in feature plot
Determine whether the color scale in the feature plot is based on all genes or individual genes. By default, the color scale is determined for each gene individually and may differ between genes. If you wish to compare gene expression between different genes, it is useful to set this parameter to "yes" so that the color scale is the same for all genes.
no

For each gene, list the average expression and percentage of cells expressing it in each cluster
Returns two tables: average expression and percentage of cells expressing the user defined genes in each cluster.
no

Result tables

- Gene's average expression level in each cluster
- Percentage of cells expressing the gene in each cluster

percentage_of_cells_expressing.tsv •••

Spreadsheet | Text | Open in New Tab | Details

Showing all 3 rows.

	0	1	2	3	4	5	6	7	8
MS4A1	4.56	5.64	5.29	86.01	4.61	8.18	5.56	3.23	7.14
LYZ	50.98	100	49.37	42.57	41.78	98.74	43.06	96.77	50
PF4	0.13	1.67	0.5	1.46	0.99	3.77	0	6.45	100

average_expressions.tsv •••

Spreadsheet | Text | Open in New Tab | Details

Showing all 3 rows.

	0	1	2	3	4	5	6	7	8
MS4A1	0.192	0.217	0.221	11.749	0.265	0.256	0.255	0.063	0.469
LYZ	3.211	183.343	2.874	3.223	2.688	30.414	2.892	127.022	11.558
PF4	0.006	0.099	0.022	0.059	0.111	0.152	0	0.216	158.976

Extract information from Seurat R-object

- Seurat R-object consists of specific data slots which contain more slots
- Chipster tool "Extract information from Seurat object" allows you to check
  - what the scRNA-seq data set includes
    - How many cells and genes
    - What genes
    - What are the highly variable genes
  - whether the data has already been normalised and how (SCTransform or global scaling)
  - which Seurat functions were used
- If you get a Seurat object from somebody and import it to Chipster, you can see what has been done

Result tables

- Text file including the different slots in the object such as the counts and assays
- Meta data table containing additional information associated with the cells or features of the object

slots.txt

File size 1.0 kB.

```
[1] "Assays in the seurat object: "
$RNA
Assay data with 13714 features for 2700 cells
First 10 features:
 AL627309.1, AP006222.2, RP11-206L10.2, RP11-206L10.9, L
KLHL17, PLEKHN1, RP11-5407.17, HES4

[1] "Active assay in the object: "
[1] "RNA"
[1] "Active cluster identity in the cluster: "
AAACATACAACCAC-1 AAACATTGAGCTAC-1 AAACATTGATCAGC-1 AAACC
            PBMC             PBMC             PBMC
AAACCGTGTATGCG-1 AAACGCACTGGTAC-1
            PBMC             PBMC
Levels: PBMC
[1] "List of graph objects in the seurat object:"
list()
[1] "List of neighbor objects in the seurat object:"
list()
[1] "List of dimensional reductions for this object:"
list()
[1] "List of spatial image objects in this object:"
list()
[1] "Name of the project:"
[1] "PBMC"
[1] "A list of miscellaneous information in the Seurat o
list()
[1] "Version of Seurat this object was built under:"
[1] '4.1.1'
[1] "A list of logged commands run on this Seurat object
list()
[1] "A list of miscellaneous data generated by other too
list()
```

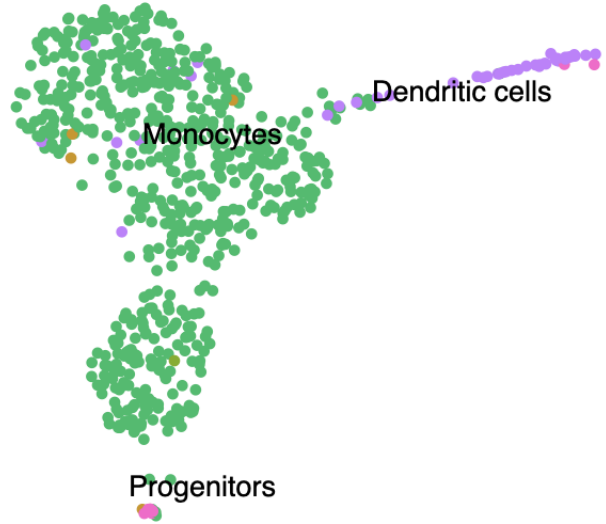
meta_data.tsv

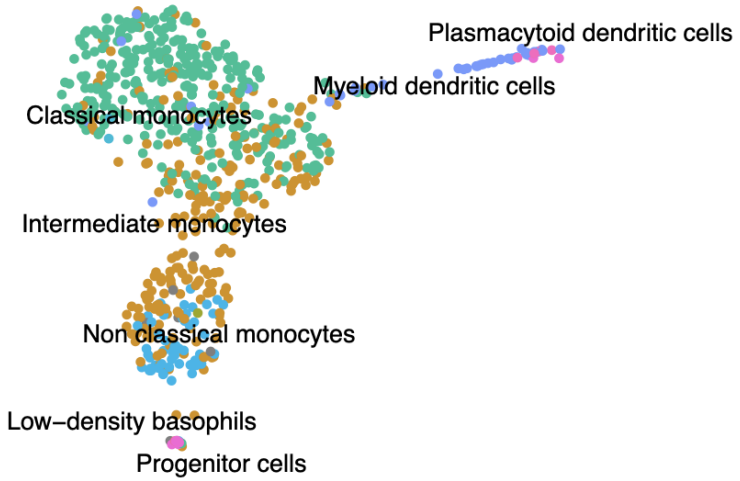
Showing all 2700 rows.

	orig.ident	nCount_RNA	nFeature_RNA	percent.mt
AAACATACAACCAC-1	PBMC	2419	779	3.0177759
AAACATTGAGCTAC-1	PBMC	4903	1352	3.7935958
AAACATTGATCAGC-1	PBMC	3147	1129	0.8897363
AAACCGTGCTTCCG-1	PBMC	2639	960	1.7430845
AAACCGTGTATGCG-1	PBMC	980	521	1.2244898
AAACGCACTGGTAC-1	PBMC	2163	781	1.6643551
AAACGCTGACCAGT-1	PBMC	2175	782	3.8160920
AAACGCTGGTTCTT-1	PBMC	2260	790	3.0973451
AAACGCTGTAGCCA-1	PBMC	1275	532	1.1764706
AAACGCTGTTTCTG-1	PBMC	1103	550	2.9011786
AAACTTGAAAAACG-1	PBMC	3914	1112	2.6315789
AAACTTGATCCAGA-1	PBMC	2388	747	1.0887772
AAAGAGACGAGATA-1	PBMC	2410	864	1.0788382
AAAGAGACGCGAGA-1	PBMC	3033	1058	1.4177382
AAAGAGACGGACTT-1	PBMC	1151	457	2.3457863
AAAGAGACGGCATT-1	PBMC	792	335	2.3989899
AAAGATCTGGGCAA-1	PBMC	1347	551	5.9391240
AAAGCAGAAGCCAT-1	PBMC	1158	567	5.0949914
AAAGCAGATATCGG-1	PBMC	4584	1422	1.3961606
AAAGCCTGTATGCG-1	PBMC	2928	1013	1.7076503
AAAGGCCTGTCTAG-1	PBMC	4973	1445	1.5282526
AAAGTTTGATCACG-1	PBMC	1268	444	3.4700315
AAAGTTTGGGGTGA-1	PBMC	3281	1015	2.5906736
AAAGTTTGTAGAGA-1	PBMC	1102	417	1.5426497
AAAGTTTGTAGCGT-1	PBMC	2683	877	2.4972046
AAATCAACAATGCC-1	PBMC	2319	787	1.1642950
AAATCAACACCAGT-1	PBMC	1412	508	1.9830028
AAATCAACCAGGAG-1	PBMC	2800	823	2.2500000
AAATCAACCCTATT-1	PBMC	5676	1541	2.4312896
AAATCAACGGAAGC-1	PBMC	3473	996	1.7564066

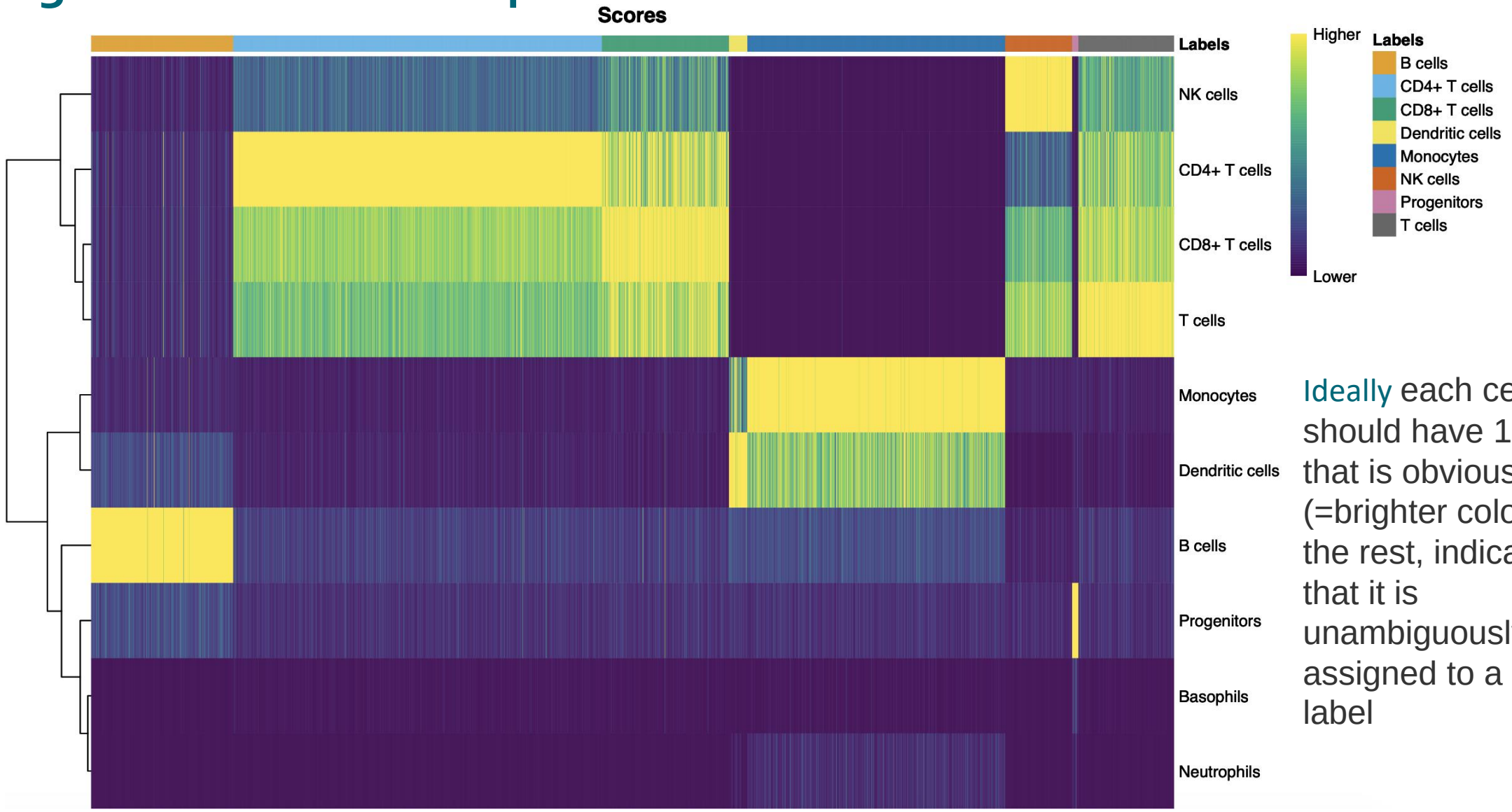
SingleR annotations to clusters

- **SingleR** is an automatic annotation method for scRNA-seq data
- Labels cells from the query dataset based on similarity to the reference dataset with known labels
- The **CellDex reference package** provides access to several reference datasets (mostly derived from bulk RNA-seq or microarray data) through dedicated retrieval functions -> sometimes, connection issues
- User can select the CellDex package to be used as reference
- Main level & fine level annotations





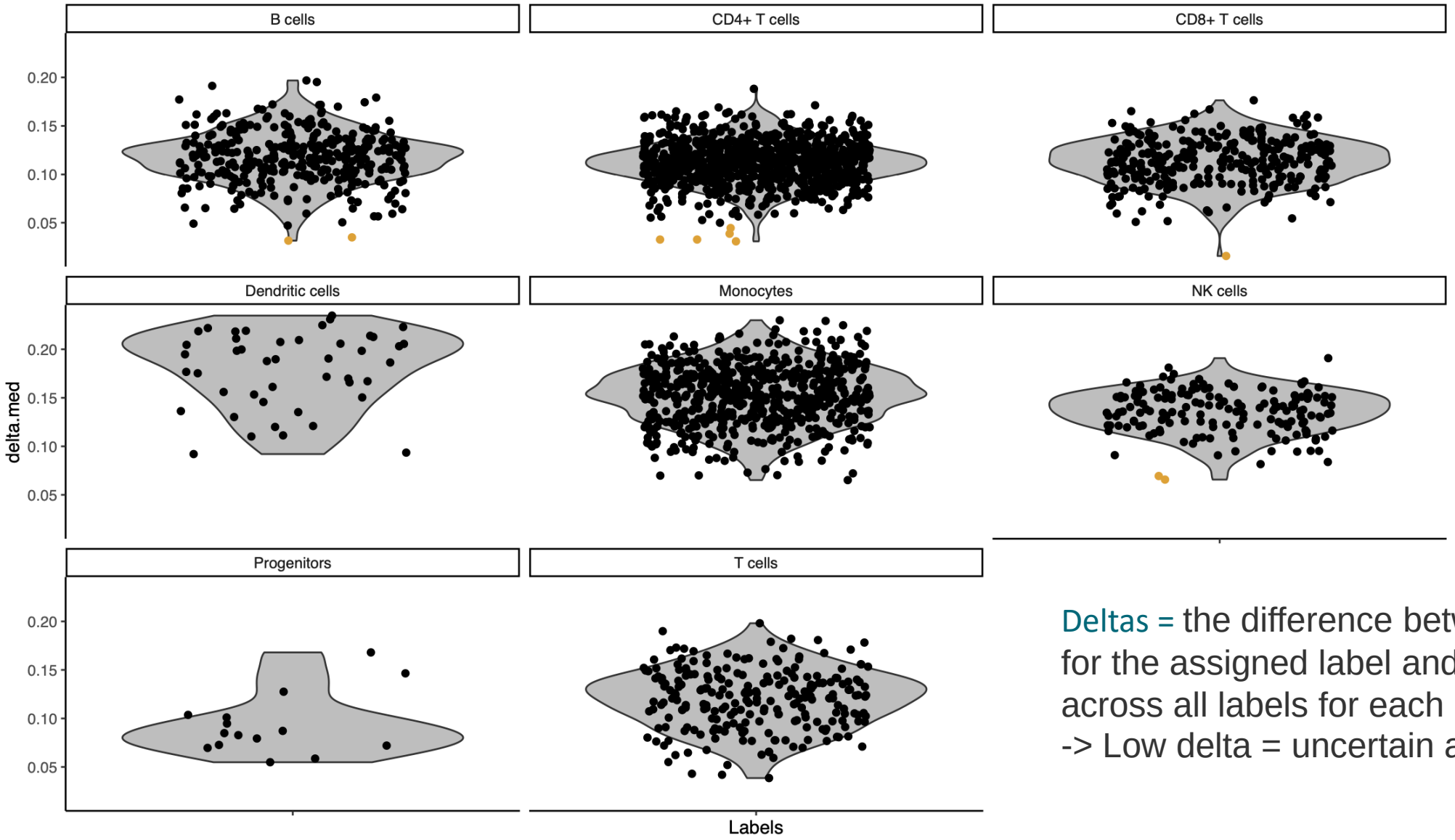
SingleR annotation: QC plots



Ideally each cell should have 1 score that is obviously larger (=brighter color) than the rest, indicating that it is unambiguously assigned to a single label

Columns = cells

SingleR annotation: QC plots



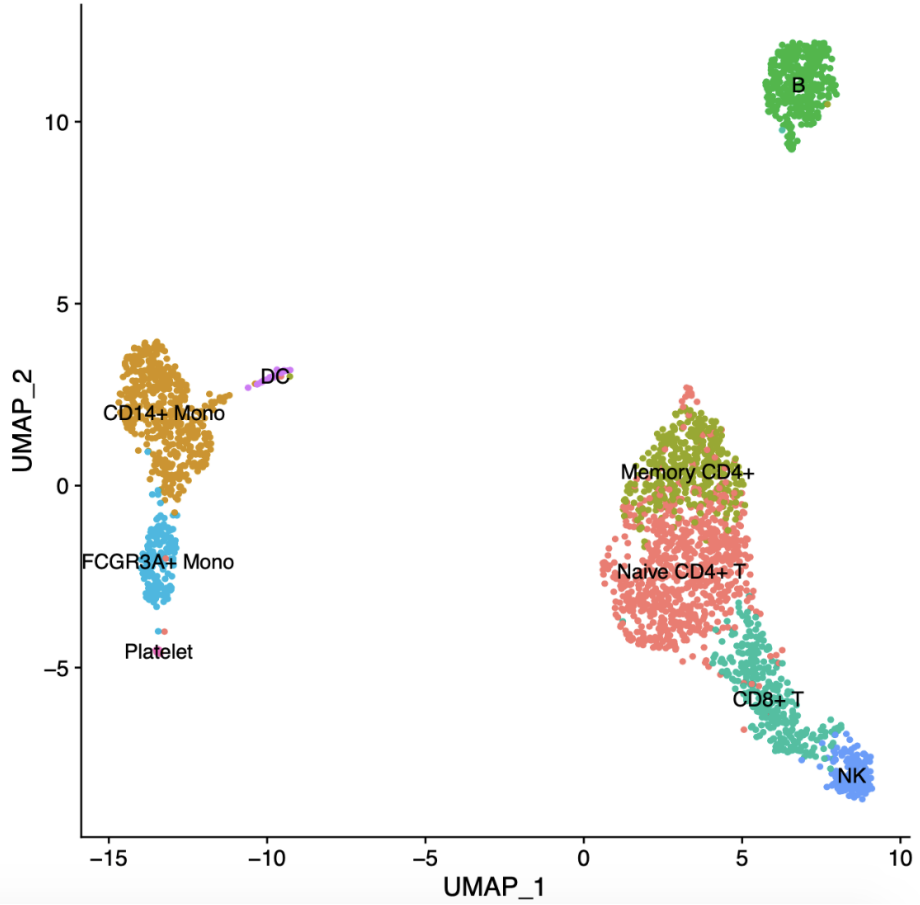
Pruned = potentially poor-quality or ambiguous assignments are removed based on the deltas

Deltas = the difference between the score for the assigned label and the median across all labels for each cell
-> Low delta = uncertain assignment

Rename clusters

- Based on previous knowledge and/or the SingleR results
- Import a table like this:

Cluster ID	Cluster name
0	Naive CD4+ T
1	CD14+ Mono
2	Memory CD4+
3	B
4	CD8+ T
5	FCGR3A+ Mono
6	NK
7	DC
8	Platelet



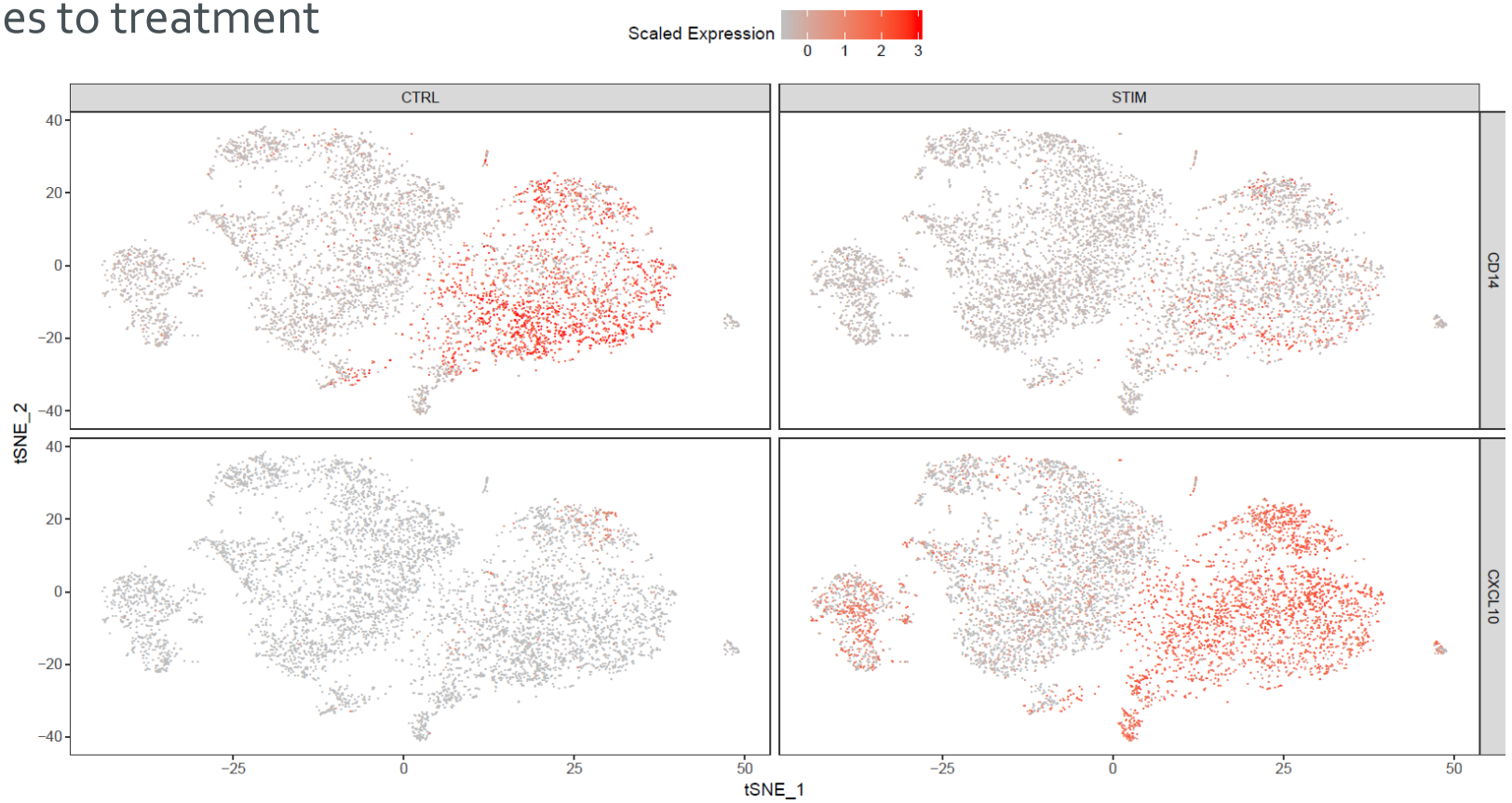
Integrated analysis of multiple samples

What will you learn

1. What we need to consider when comparing samples
2. How to integrate samples
3. How to find conserved cluster marker genes
4. How to find differentially expressed genes between samples, within clusters
5. How to visualize interesting genes

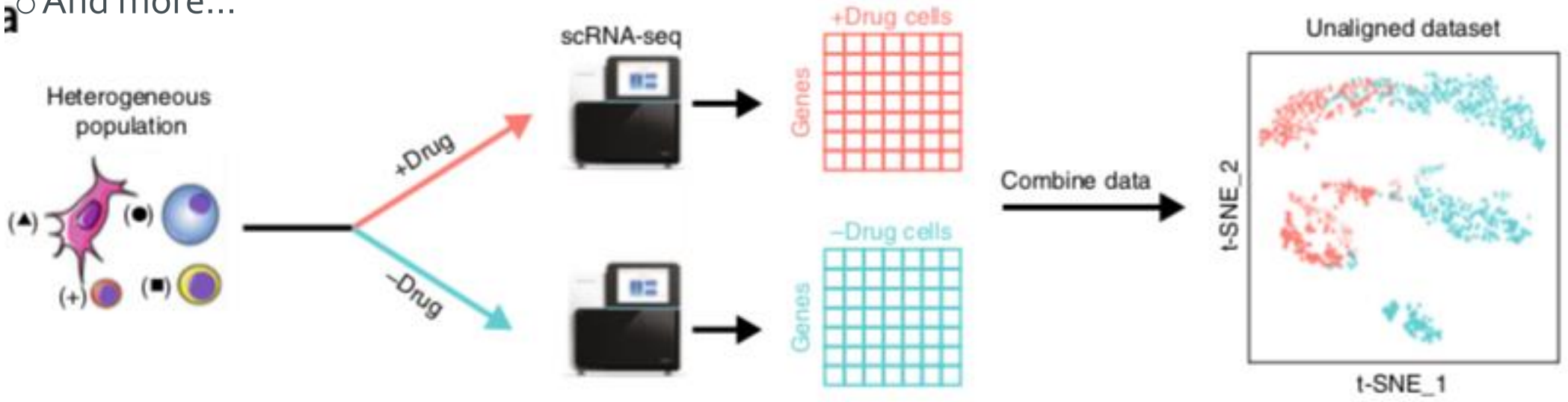
Goals of integrated analysis

- When comparing two samples, e.g. control and treatment, we want to
  - Identify cell types that are present in both samples
  - Obtain cell type markers that are conserved in both control and treated cells
  - Find cell-type specific responses to treatment



When comparing samples we need to correct for batch effects

- We need to find corresponding cells in the samples
  - Technical and biological variability can cause batch effects which make this difficult
- Several batch effect correction methods for single cell RNA-seq data available, e.g.
  - Seurat v2: Canonical correlation analysis (CCA) + dynamic time warping
  - Seurat v3-v4: CCA + anchors
  - Mutual nearest neigbors (MNN)
  - And more…

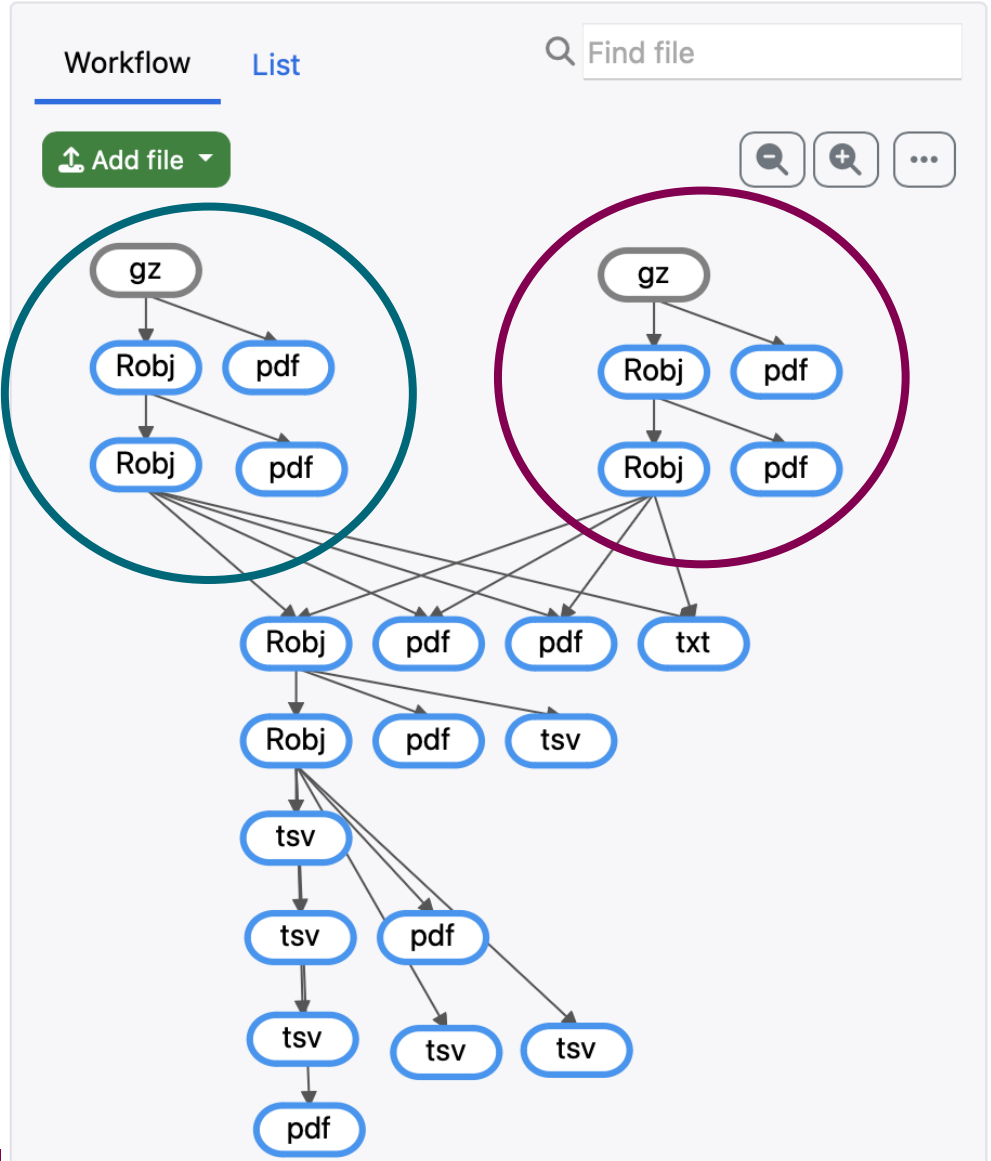


Analysis steps for integrated analysis

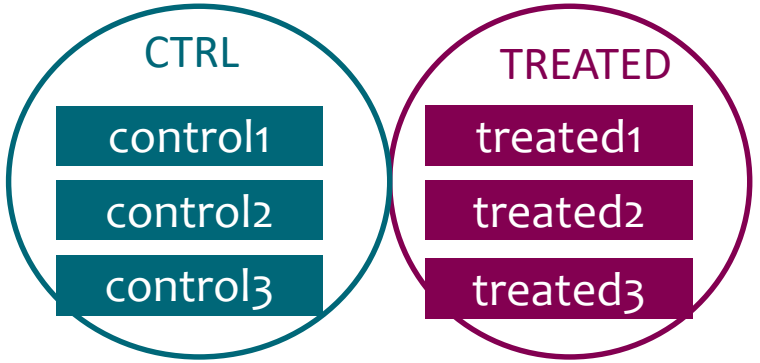
1. Create Seurat objects, check the quality & filter cells
2. Merge samples together
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, perform PCA
6. Integrate samples and perform CCA, align samples
7. Cluster cells, visualize clusters with tSNE or UMAP
8. Find conserved biomarkers for clusters
9. Find differentially expressed genes between samples, within clusters
10. Visualize interesting genes

Integrated analysis: Setup, QC, filtering

Files



- Perform the Seurat object setup, QC and filtering steps separately for the samples
  - Same as before, just remember to name:
  - the samples: e.g. control_1, control_2…
  - **and** sample groups: e.g. CTRL and STIM



Parameters

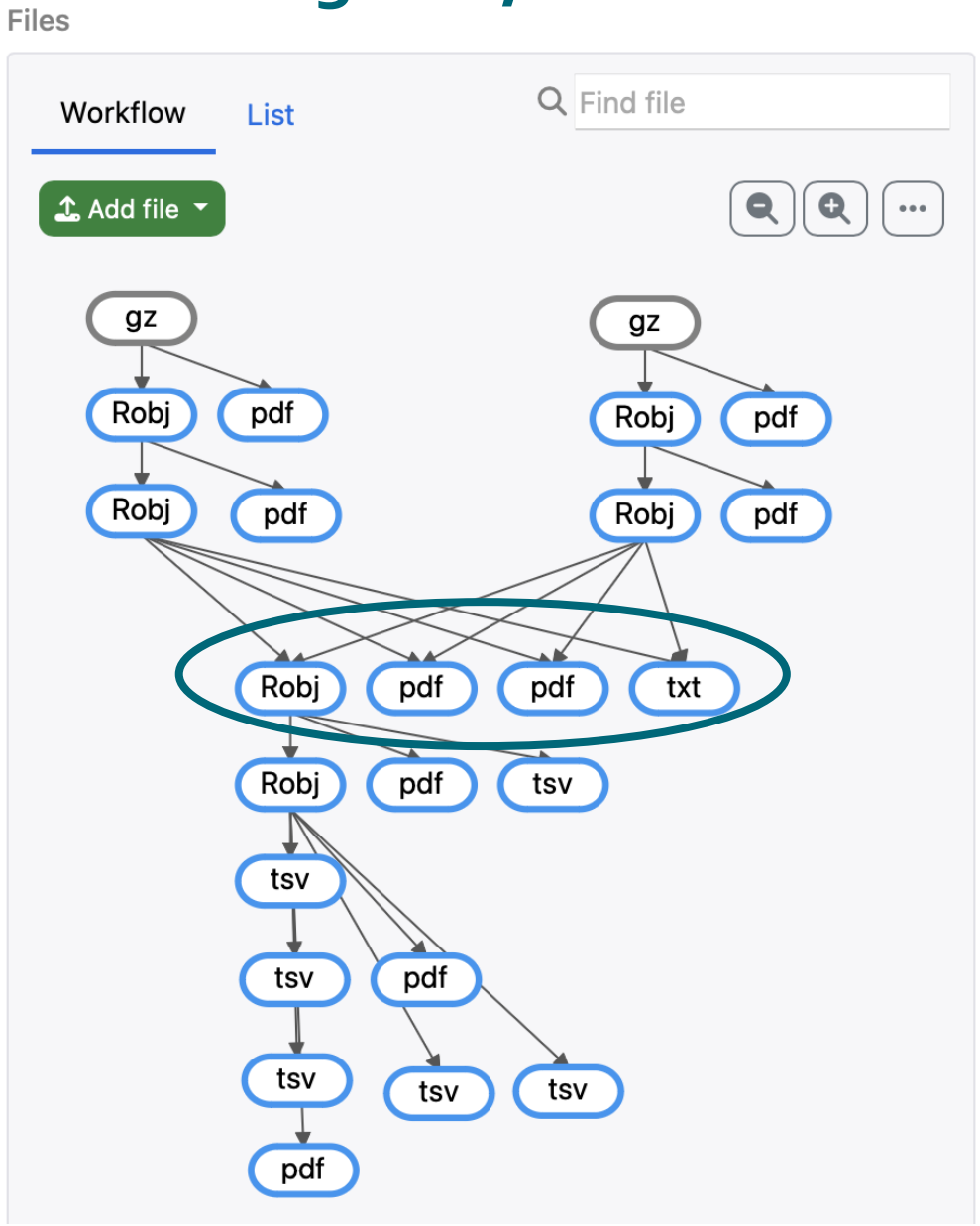
Project name for plotting You can give your project a name. The name will appear on the plots. Do not use underscore _ in the names!	My_project_name
Sample name Type the sample name or identifier here. For example control1, cancer3a. Do not use underscore _ in the names! Fill this field if you are combining samples later.	control1
Sample group Type the sample name or identifier here. For example CTRL, STIM, TREAT. Do not use underscore _ in the names! Fill this field if you are combining samples later.	CTRL

Analysis steps for integrated analysis

1. Create Seurat objects, check the quality & filter cells
2. Merge samples together
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, perform PCA
6. Integrate samples and perform CCA, align samples
7. Cluster cells, visualize clusters with tSNE or UMAP
8. Find conserved biomarkers for clusters
9. Find differentially expressed genes between samples, within clusters
10. Visualize interesting genes

Merge samples together, normalise, detect variable genes, regress and PCA

- These steps happen in one tool
- Parameters are same as in 1 sample case
- The resulting larger Seurat R object contains all the information about all the merged samples
  - Seurat v5 introduced sample "Layers", which allows this merging of samples in the early stage of the analysis

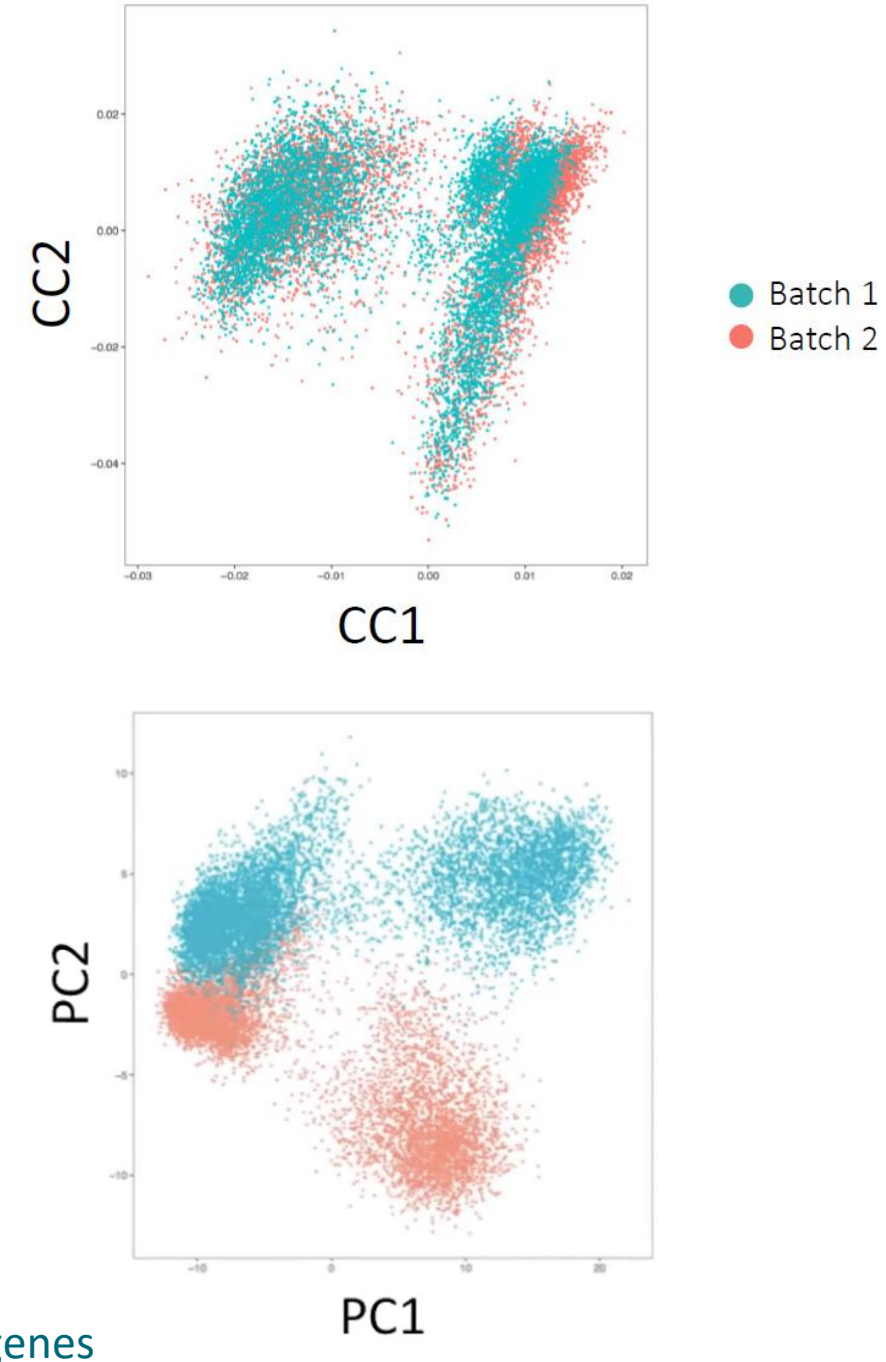


Analysis steps for integrated analysis

1. Create Seurat objects, check the quality & filter cells
2. Merge samples together
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, perform PCA
6. Integrate samples and perform CCA, align samples
7. Cluster cells, visualize clusters with tSNE or UMAP
8. Find conserved biomarkers for clusters
9. Find differentially expressed genes between samples, within clusters
10. Visualize interesting genes

Canonical correlation analysis (CCA)

- Dimension reduction, like PCA
- Captures common sources of variation between two datasets
  - Aim: place datasets in a shared, low-dimensional space
- Produces canonical correlation vectors, CCs
  - Effectively capture correlated gene modules that are present in both datasets
  - Represent genes that define a shared biological space
- Why not PCA?
  - It identifies the sources of variation, even if present only in 1 sample (e.g. technical variation)
  - We want to integrate, so we want to find the *similarities*

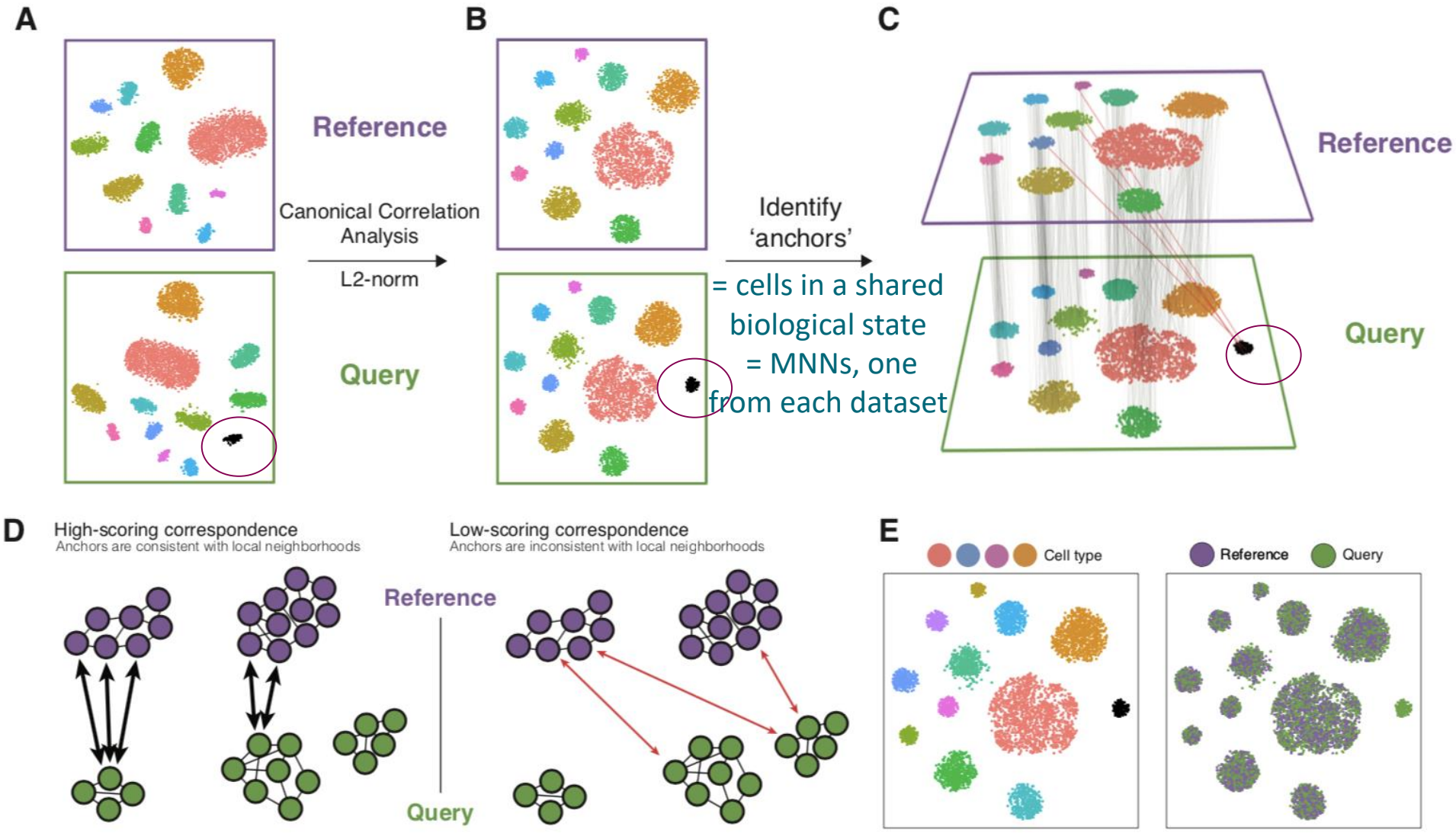


Input:
Highly variable genes

Aligning two samples (Seurat v3/v4)

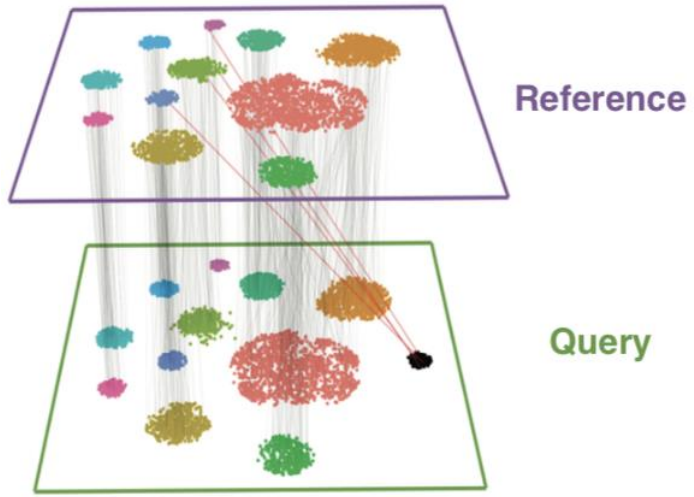
See the Seurat paper:
https://www.cell.com/cell/fulltext/S0092-8674(19)30559-8

1. Canonical correlation analysis
   + L2-normalisation of CCVs for scaling
   → shared space
2. Identify pairs of mutual nearest neighbors (MNN) → "anchors"
3. Filter & score anchors (based on neighborhood, in PC space)
4. Anchors + scores → correction vectors



Integrate multiple samples tool

1. Identify "anchors" for data integration
   - Parameter: how many CCs to use in the neighbor search [30]
2. Integrate datasets together
   - Parameter: how many PCs to use in the anchor weighting procedure [30]



Number of CCs to use in the neighbor search 30

Which dimensions to use from the CCA to specify the neighbor search space. The neighbors are used to determine the anchors for the alignment.

Number of PCs to use in the integration 30

Number of PCs to use in the anchor weighting procedure. The anchors and their weights are used to compute the correction vectors, which allow the datasets to be integrated.

Same question as before: What is the dimensionality of the data?

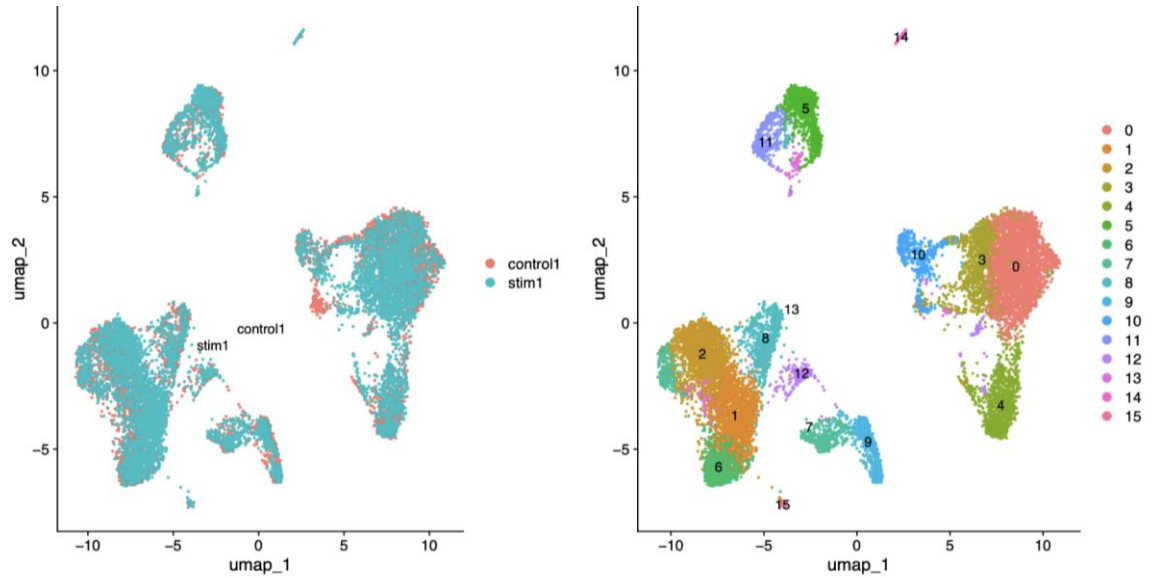
Dimensionality –how many CCs / PCs to choose for downstream analysis?

- In the article* by Seurat developers, they "neglect to finely tune this parameter for each dataset, but still observe robust performance over diverse use cases".
  - For all neuronal, bipolar, and pancreatic analyses: dimensionality of 30.
  - For scATAC-seq analyses: 20.
  - For analyses of human bone marrow: 50
  - The integration of mouse cell atlases: 100
- Higher numbers: for significantly larger dataset and increased heterogeneity

* https://www.biorxiv.org/content/biorxiv/early/2018/11/02/460147.full.pdf

Integrated analysis of two samples –tools (v5)

1. Cluster cells
   - As before
2. Visualize clustering
   - tSNE or UMAP, as a parameter
   - Number of PCs to use for UMAP or TSNE
   - Before and after integration

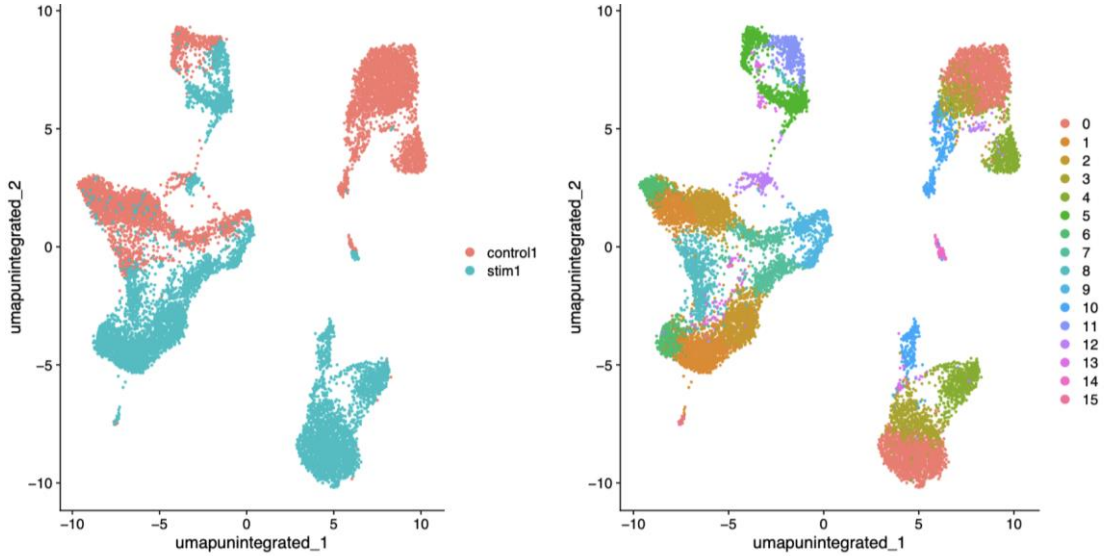


Total number of cells: 14022
Number of cells in each cluster:

	control1	stim1
0	1625	1486
1	671	1139
2	876	905
3	512	618
4	488	541
5	391	556
6	354	405
7	314	421
8	321	340
9	304	311
10	294	197
11	186	201
12	117	117
13	47	127
14	40	66
15	22	30
sums	6562	7460

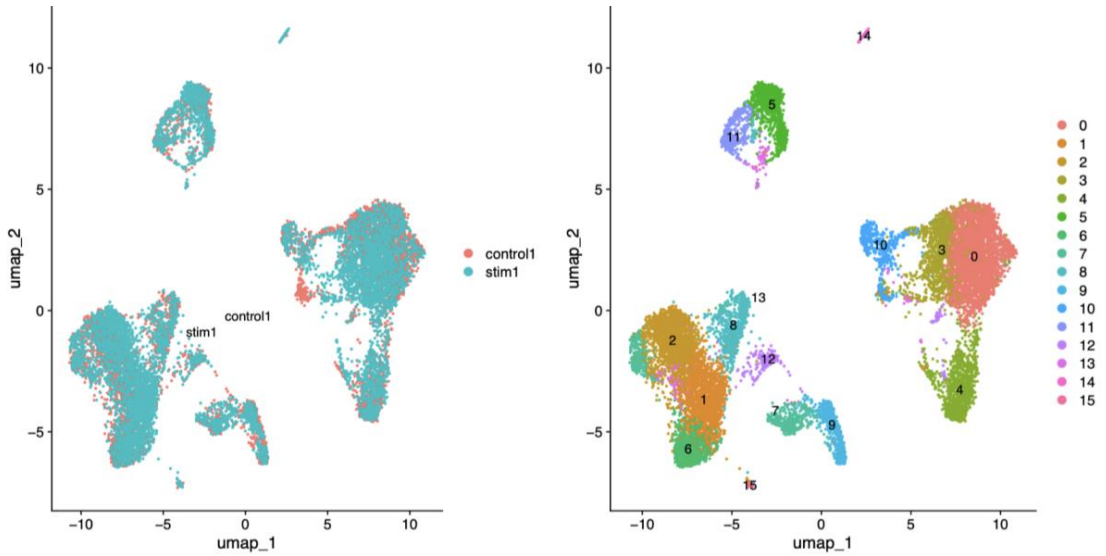
Integrated analysis of two samples –tools (v5)

unintegrated

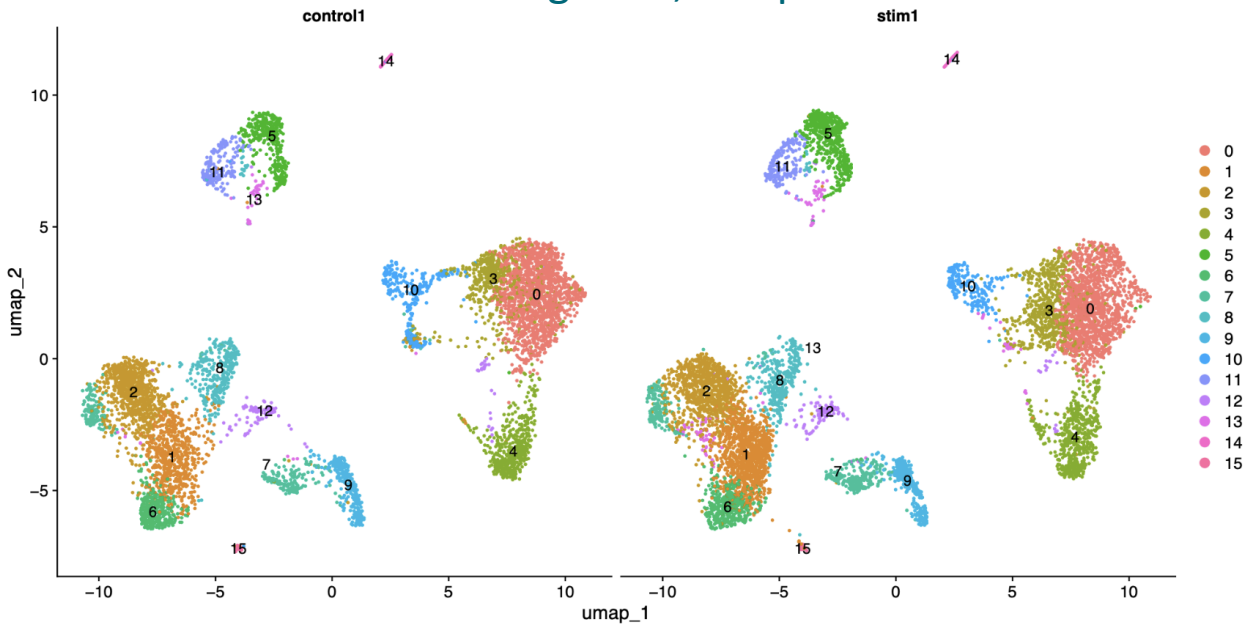


- Visualisations before and after integration
- Samples after integration

integrated



Integrated, samples:



Large datasets: Anchor identification method (CCA → rPCA)

- CCA = default
  - Might lead to overcorrection, especially when large proportion of cells are non-overlapping
  - Recommended when:
    - When cell types are conserved, but there's still big difference between the samples/experiments → experimental condition/disease causes very strong expression shift
    - Cross-modality mapping
    - Cross-species mapping
- rPCA = reciprocal PCA
  - **Faster**, more conservative: cells in different biological states are less likely to "align"
  - Each dataset is projected into the others PCA space and the anchors are constrained by the same mutual neighbourhood requirement
  - Recommended when:
    - A substantial fraction of cells in one dataset have no matching type in the other
    - Datasets originate from the same platform (i.e. multiple lanes of 10x Genomics)
    - There are a large number of datasets or cells to integrate

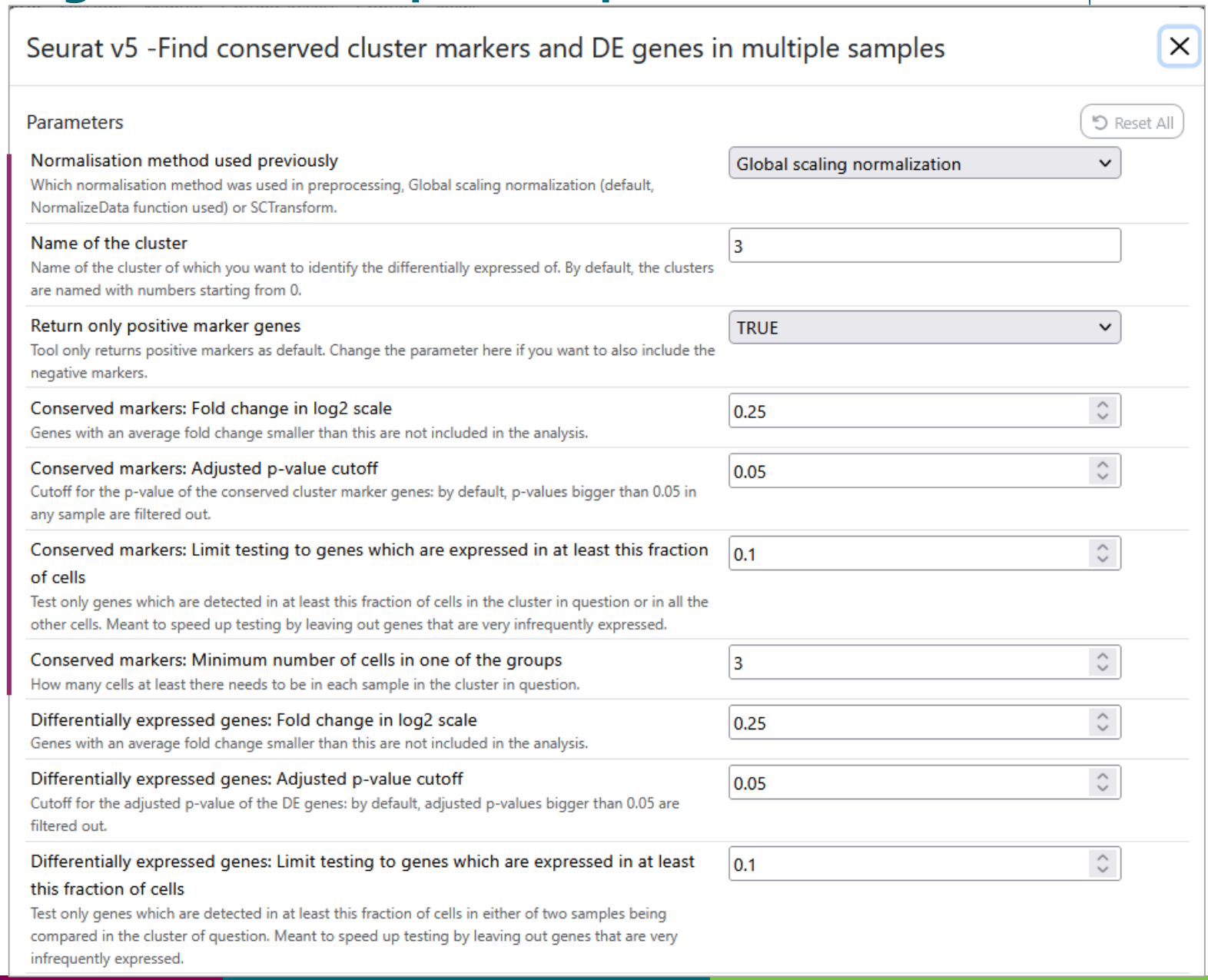
https://satijalab.org/seurat/articles/integration_rpca.html
https://satijalab.org/seurat/articles/integration_large_datasets.html

Analysis steps for integrated analysis

1. Create Seurat objects, check the quality & filter cells
2. Merge samples together
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, perform PCA
6. Integrate samples and perform CCA, align samples
7. Cluster cells, visualize clusters with tSNE or UMAP
8. Find conserved biomarkers for clusters
9. Find differentially expressed genes between samples, within clusters
10. Visualize interesting genes

Find conserved cluster marker genes in multiple samples

- Conserved marker gene = marker for a given cluster in *all* samples
  - Give a cluster as a parameter
  - Compares gene expression in cluster X vs all other cells
  - This is done in each sample, and then the p-values are combined using Wilkinson's method
- Uses Wilcoxon rank sum test



conserved_markers.tsv

Showing all 510 rows.

	control1_p_val	control1_avg_log2FC	control1_pct.1	control1_pct.2	control1_p_val_adj	stim1_p_val	stim1_avg_log2FC	stim1_pct.1	stim1_pct.2
SDS	4.53150278523327e-122	3.6517585874248	0.229	0.022	6.9060102446955e-118	3.12962456215534e-303	3.17123475537841	0.629	0.095
CCL2	3.6627405017609e-188	2.95933277524106	0.686	0.167	5.58201652468361e-184	1.49961677356546e-281	2.42180087627377	0.969	0.343
CTSB	1.18556159693945e-190	2.00538635679143	0.938	0.365	1.80679587373572e-186	3.92827371024726e-280	2.22831787437411	0.966	0.344
CTSL	5.23160940291684e-224	2.5319477894919	0.875	0.261	7.97297273004526e-220	2.44752278724089e-275	2.4204889736656	0.932	0.308
PLA2G7	2.1515083524576e-168	1.97200575718488	0.789	0.226	3.27889872914538e-164	3.03135007978236e-267	2.2423192743904	0.809	0.186
LYZ	8.13962661004904e-137	1.39926157545676	0.883	0.335	1.24047909537147e-132	1.2365176770847e-260	2.10455997059062	0.927	0.281
LGALS3	6.63508558858831e-183	1.97914569780486	0.982	0.471	1.01118704370086e-178	5.80542354973039e-255	2.11930528291019	0.969	0.363
HSPA1A	9.37058724889599e-66	0.958553509785575	0.463	0.15	1.42807749673175e-61	2.69100055084202e-233	1.89359934903763	0.934	0.327
CD63	1.46378081609245e-173	1.53923933029892	0.998	0.538	2.2308019637249e-169	1.73029976510931e-231	1.69101677472477	0.995	0.448

stim1_p_val_adj	max_pval	minimump_p_val	max.adj.p.val	minimum.adj.p.val
4.76954783272474e-299	4.53150278523327e-122	6.2592491243106e-303	6.9060102446955e-118	4.76954783272474e-299
2.28541596291376e-277	3.6627405017609e-188	2.99923354713062e-281	5.58201652468361e-184	2.28541596291376e-277
5.98668913441683e-276	1.18556159693945e-190	7.85654742049405e-280	1.80679587373572e-186	5.98668913441683e-276
3.73002472775511e-271	5.23160940291684e-224	4.89504557448163e-275	7.97297273004526e-220	3.73002472775511e-271
4.61977752158832e-263	2.1515083524576e-168	6.0627001595642 5e-267	3.27889872914538e-164	4.61977752158832e-263
1.88445293987708e-256	8.13962661004904e-137	2.47303535416926e-260	1.24047909537147e-132	1.88445293987708e-256
8.84746548978912e-251	6.63508558858831e-183	1.16108470994599e-254	1.01118704370086e-178	8.84746548978912e-251
4.10108483948323e-229	9.37058724889599e-66	5.38200110168377e-233	1.42807749673175e-61	4.10108483948323e-229
2.6369768420266e-227	1.46378081609245e-173	3.46059953021834e-231	2.2308019637249e-169	2.6369768420266e-227

Find cell-type specific differentially expressed genes between samples

- We are now looking for differential expression between *samples* in one cluster
- Uses Wilcoxon rank sum test
- Parameters for filtering the table:
  - Adjusted p-value cutoff for differentially expressed genes (default = 0.05)
  - Fold change threshold for differentially expressed genes in log2 scale (default = 0.25)
- If there are >2 samples, a table for each sample is given as output
  - named: de-list_sample1VsAllOthers.tsv, de-list_sample2VsAllOthers.tsv…

de-list_PBMC_control_vs_PBMC_stim.tsv •••

Spreadsheet | Text | Open in New Tab | Details

Showing the first 100 of 778 rows. View in full screen to see all rows.

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj	aver_expr_ident1	aver_expr_ident2
ISG15	1.7296497545789e-178	-4.653740532325	0.231	0.998	2.43067680010973e-174	2.5918	89.4134
IFIT3	7.75442106128566e-172	-4.51511176245437	0.046	0.959	1.08972879174247e-167	0.4133	31.3164
ISG20	8.79249690050336e-169	-2.99108560051549	0.651	1	1.23560958942774e-164	9.8323	85.1244
IFI6	6.95069852044584e-167	-4.19106395111127	0.076	0.953	9.76781663078254e-163	0.6552	29.2335
IFIT1	1.0058042691094e-150	-4.11242648599536	0.028	0.886	1.41345673937944e-146	0.2613	20.8171

Find cell-type specific differentially expressed genes between samples

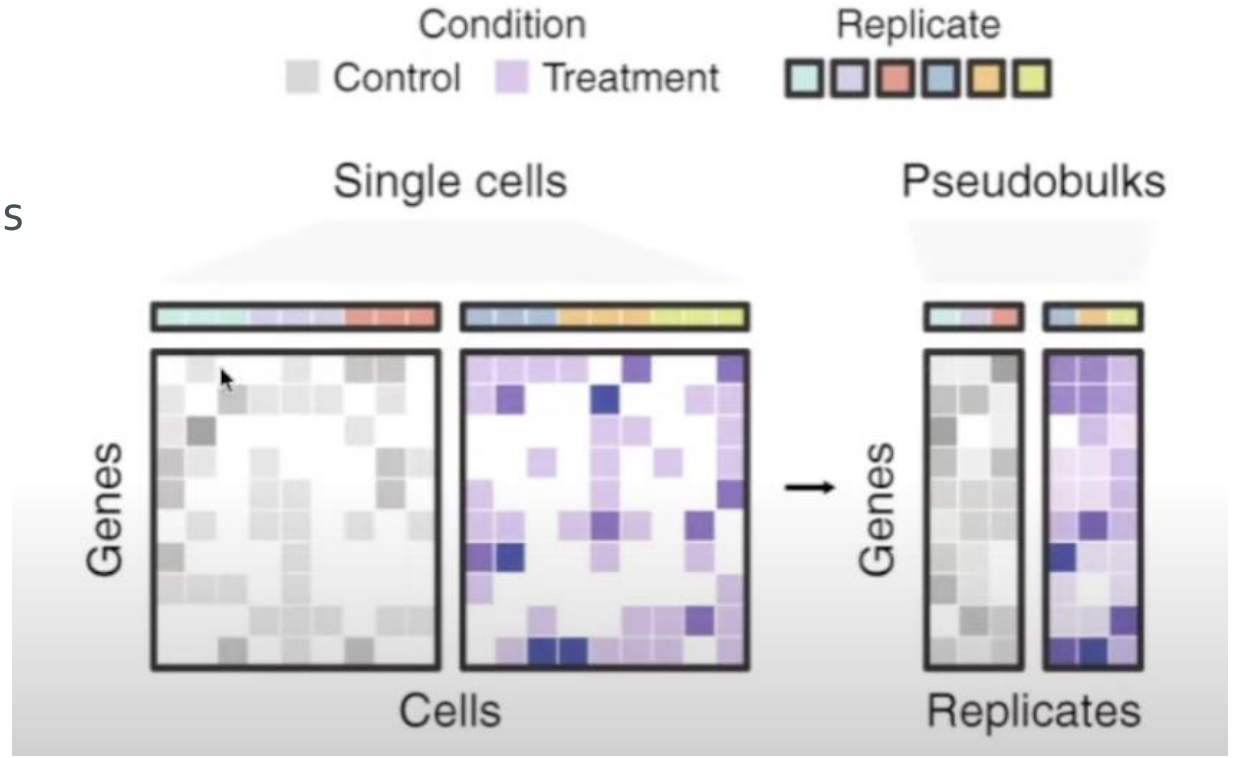
- We are now looking for differential expression between *samples* in one cluster
  - Actually, these tests are looking at the *cells* in that sample & in that cluster
  - Each cell is thus treated as an independent replicate, and the inherent correlations between cells originating from the same sample are ignored → Large number of false positives*
  - Use caution when interpreting!
- "Solution": **pseudobulk** analysis in Seurat v5
  - Sum together gene counts of all the cells from the sample & cluster → one expression profile, samples treated as individual observations. Then compare to the cell-wise results.
- 3 tools:

Seurat v4 -Find conserved cluster markers and DE genes in multiple samples	All sample-wise comparisons at once
Seurat v4 -Find DE genes between chosen samples	Choose the samples you want to compare
Seurat v4 -Find DE genes between sample groups	Choose to compare sample groups (from the setup tool)

*Squair et al., 2021, Zimmerman et al., 2021, Junttila et al., 2022

What is pseudobulk analysis?

- Why is it better than scRNAseq or bulk RNAseq?
  - Each cell treated as sample → inflated p-values → false positives
  - Tendency to identify high expressors as DE genes
  - Ignoring the variation across population / unmodelled correlation between samples
  - Get the benefit of single-cell resolution (= recognition of cell types) AND the statistical rigor of the existing bulk-RNAseq methods



https://www.nature.com/articles/s41467-021-25960-2/figures/2

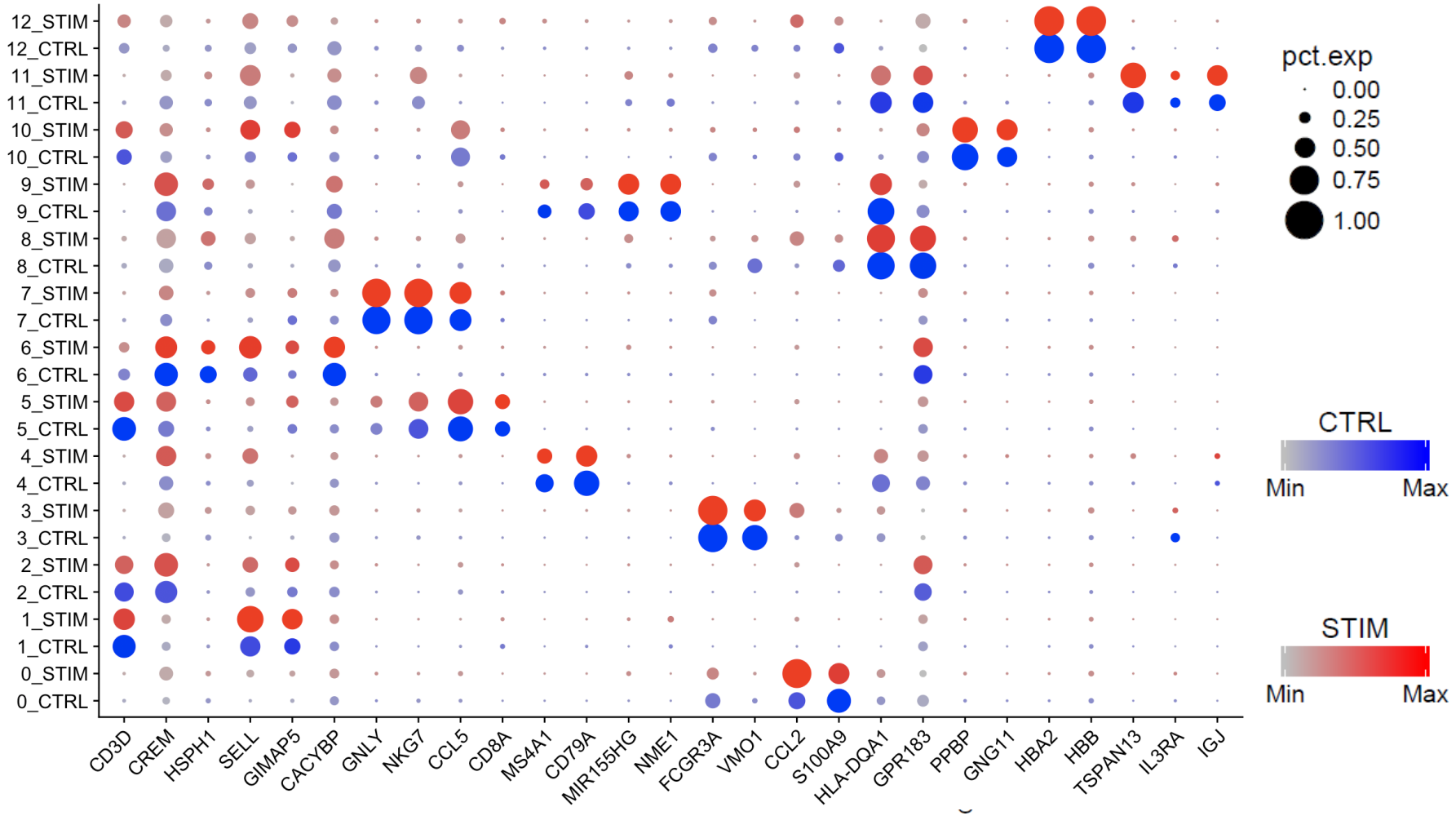
Bioinformagician in YouTube: https://youtu.be/04gB2owLKus?si=2O7qDWlTKOkSz7J8

Analysis steps for integrated analysis

1. Create Seurat objects, check the quality & filter cells
2. Merge samples together
3. Normalize expression values
4. Identify highly variable genes
5. Scale data, perform PCA
6. Integrate samples and perform CCA, align samples
7. Cluster cells, visualize clusters with tSNE or UMAP
8. Find conserved biomarkers for clusters
9. Find differentially expressed genes between samples, within clusters
10. Visualize interesting genes

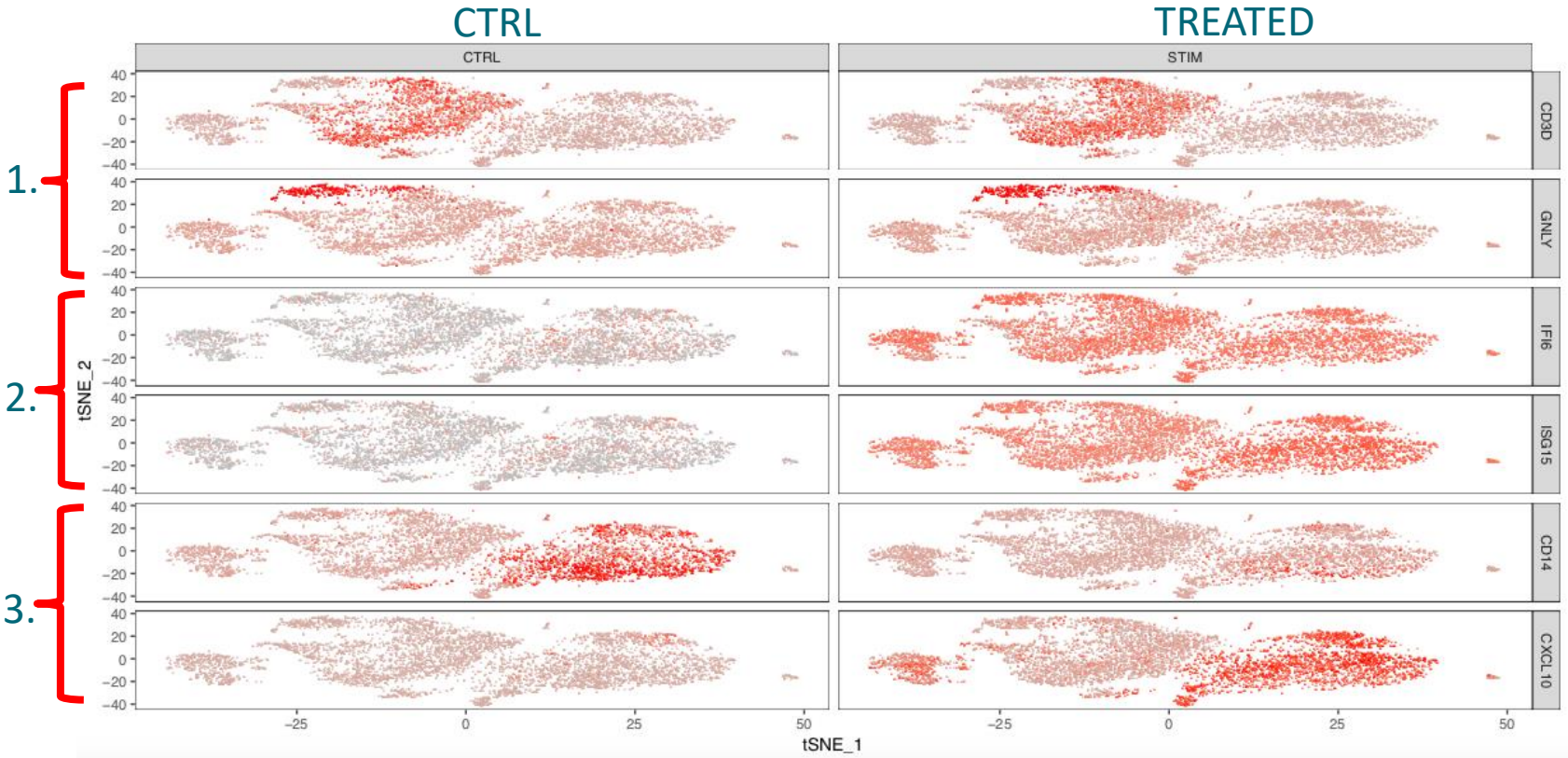
Visualize interesting genes in split dot plot

- Size = the percentage of cells in a cluster expressing a given gene
- Brightness = the average expression level in the expressing cells in a cluster



Visualize interesting genes in tSNE/UMAP plots

1. No change between the samples: conserved cell type markers
2. Change in all clusters: cell type independent marker for the treatment
3. Change in one/some clusters: cell type dependent behavior to the treatment



Different marker genes

Visualize interesting genes in violin plots

1. No change between the samples: conserved cell type markers

2. Change in all clusters: cell type independent marker for the treatment

3. Change in one/some clusters: cell type dependent behavior to the treatment