# Analysing Single-Cell RNA-Seq with R

v2020-11

Simon Andrews
simon.andrews@babraham.ac.uk

Babraham Bioinformatics

# Major R scRNA Package Systems


https://satijalab.org/seurat/


https://cole-trapnell-lab.github.io/monocle3/

**Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R**

Davis J McCarthy, Kieran R Campbell, Aaron T L Lun, Quin F Wills

*Bioinformatics*, Volume 33, Issue 8, 15 April 2017, Pages 1179–1186, https://doi.org/10.1093/bioinformatics/btw777

**Published:** 14 January 2017 **Article history**

https://bioconductor.org/packages/release/bioc/html/scater.html

# What do they provide?

- Data Structure for modelling scRNA-Seq
  - Counts
  - Normalisations
  - Metadata
  - Clusters
- Convenience methods
  - Data access
  - Data parsing
  - Data access
  - Simple transformations

# What do they provide?

- Implementations of common methods
  - Data Normalisation
  - Dimensionality reduction
    - PCA
    - tSNE
    - UMAP
- Plotting
  - Projections
  - QC
  - Standard graphs (scatterplots, violin plots, stripcharts)

# What do they provide?

- Statistics
  - Enriched genes
  - Differential expression

- Novel functionality
  - Seurat
    - Feature anchors to match datasets
  - Monocle
    - Trajectory mapping

# Seurat

- Probably the most popular choice
  - Well supported and frequently updated
- Easy data model to work with
  - Documentation is good too
- Lots of built in functionality
  - Easy to extend to build your own
- Lots of nice examples on their web pages

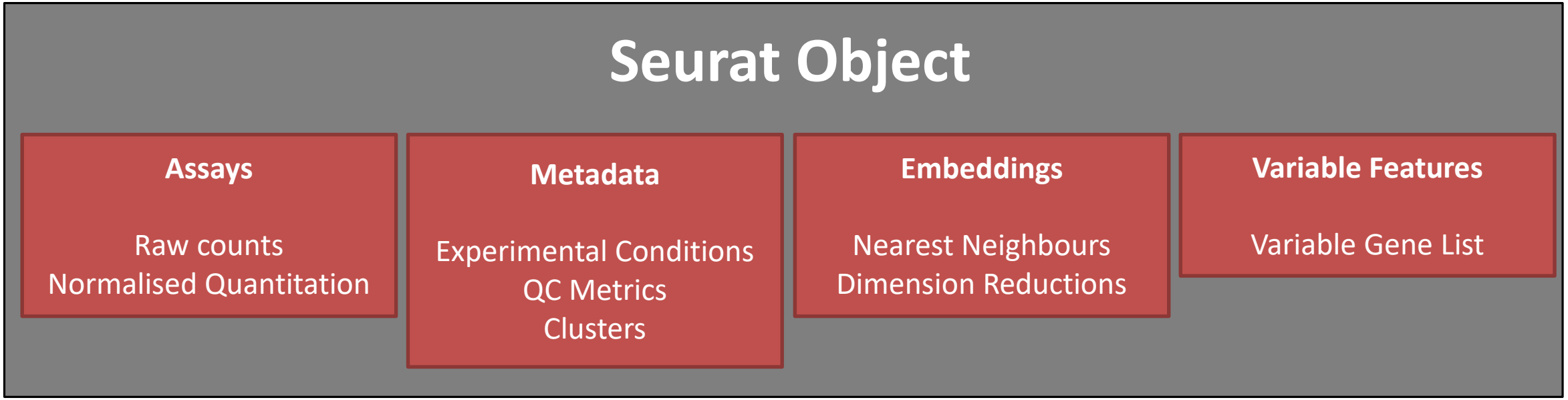
# Seurat Data Structure

- Single object holds all data
  - Build from text table or 10X output (feature matrix h5 or raw matrix)

| | | |
|---|---|---|
| data | S4 [15969 x 5058] (Seurat::Seurat) | S4 object of class Seurat |
| assays | list [1] | List of length 1 |
| meta.data | list [5058 x 3] (S3: data.frame) | A data.frame with 5058 rows and 3 columns |
| active.assay | character [1] | 'RNA' |
| active.ident | factor | Factor with 5058 levels: "course", "course", "course", "course", "course", "cour , ... |
| graphs | list [0] | List of length 0 |
| neighbors | list [0] | List of length 0 |
| reductions | list [0] | List of length 0 |
| project.name | character [1] | 'course' |
| misc | list [0] | List of length 0 |
| version | list [1] (S3: package_version, num | List of length 1 |
| commands | list [0] | List of length 0 |
| tools | list [0] | List of length 0 |

# Seurat Data Structure

- Single object holds all data
  - Build from text table or 10X output (feature matrix h5 or raw matrix)



# Seurat Metadata

- Data Frame of QC metrics (cols = metrics, rows = cells)
  - Imported classifications (not automatically carried over from cellranger, add from aggr csv file)
  - Derived clusters
  - Some automatically defined – can add your own

- Access directly or indirectly
  - `data$my.qc.metric`
  - `data@meta.data$my.qc.metric`

# Seurat Metadata

```
> head(data@meta.data)
  nCount_RNA nFeature_RNA gem_id     orig.ident
1       6538         2108      1 Influenza_day1
2       6742         1579      1 Influenza_day1
3       1420          810      1 Influenza_day1
4       1077          681      1 Influenza_day1
5       6303         2342      1 Influenza_day1
6       8947         2793      1 Influenza_day1
```

# Seurat Quantitative Data

- Counts
  - Top level is a sparse matrix (rows = genes, cols = cells)
  - Shortcut to `data@assays$RNA@counts`

- Normalised data
  - A second independent matrix
  - `data@assays$RNA@data`

- Can filter by subsetting the top level matrix

# Seurat Quantitative Data

```
> data@assays$RNA@counts[10:20,1:2]
11 x 2 sparse Matrix
          TGAGA-1 AAGGG-1
Rgs20           .       .
Atp6v1h         1       2
Alkal1          .       .
St18            .       .
Pcmtd1          .       1
Gm26901         .       .
Sntg1           .       .
```

```
> data@assays$RNA@data[10:20,1:2]
11 x 2 sparse Matrix
          TGAGA-1 AAGGG-1
Rgs20           .       .
Atp6v1h     0.928   1.377
Alkal1          .       .
St18            .       .
Pcmtd1          .   0.909
Gm26901         .       .
Sntg1           .       .
```

# Seurat Embeddings

- Reductions
  - `data$projections`
  - Rows = cells, Cols = Projection axes
    - PCA
    - tSNE
    - UMAP
- Graphs
  - `data$graphs`
  - (Sparse) Distance matrices
  - Used for graph based clustering

# Seurat Embeddings

```
> data@reductions$pca[1:5,1:5]
               PC_1         PC_2        PC_3         PC_4        PC_5
Reg3g    0.02209522  0.008870765 -0.01791399 -0.042812700 -0.18932551
Scgb3a1  0.01829751  0.005249075 -0.02640954 -0.044957754 -0.18863055
Retnla   0.02616540  0.005942941 -0.03967109 -0.049142055 -0.16423270
Bpifb1   0.01966987  0.005381136 -0.02367615 -0.045729821 -0.19999561
Cxcl13  -0.01369808 -0.010579552  0.08571013  0.004081651 -0.02472911
```

```
> data@reductions$tsne@cell.embeddings[1:5,]
                      tSNE_1     tSNE_2
AAACCTGAGAGTGAGA-1  14.182877 -12.592458
AAACCTGAGCGAAGGG-1 -24.612876   5.739248
AAACCTGAGCGTCTAT-1  21.212219   4.717356
AAACCTGAGCTACCTA-1   5.228508  21.568443
AAACCTGAGCTCCCAG-1  18.923168   6.299075
```

# Seurat Methods

- Data Parsing
  - `Read10X`
  - `Read10X_h5*`
  - `CreateSeuratObject`
- Data Normalisation
  - `NormalizeData`
  - `ScaleData`
- Graphics
  - Violin Plot – metadata or expression (`VlnPlot`)
  - Feature plot (`FeatureScatter`)
  - Projection Plot (`DimPlot, DimHeatmap`)
- Dimension reduction
  - `RunPCA`
  - `RunTSNE`
  - `RunUMAP**`
- Statistics
  - Select Variable Genes `FindVariableFeatures`
  - Build nearest neighbour graph `FindNeighbors`
  - Build graph based cell clusters `FindClusters`
  - Find genes to classify clusters (multiple tests) `FindMarkers`

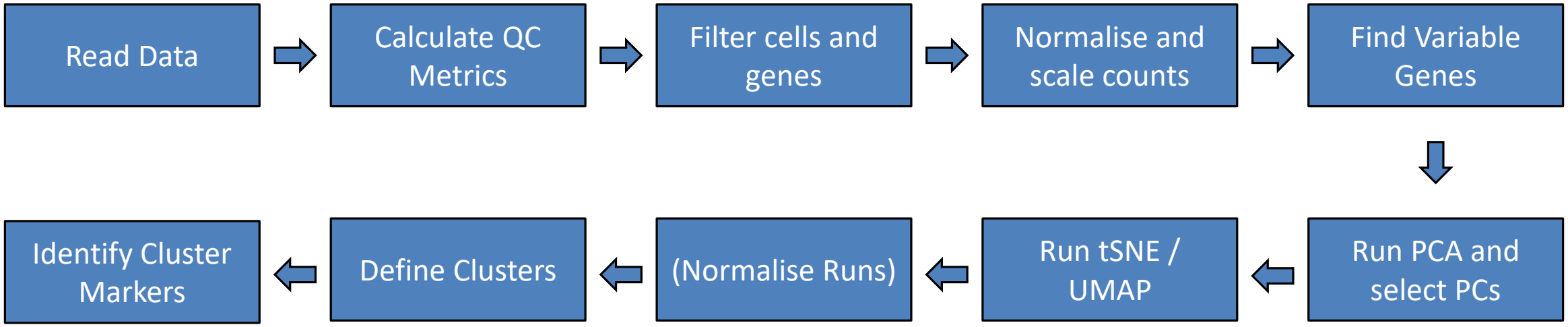
*Requires installing the `hdf5r` package

**Requires installing python and `umap-learn`

# Example 10X Seurat Workflow

Babraham Bioinformatics

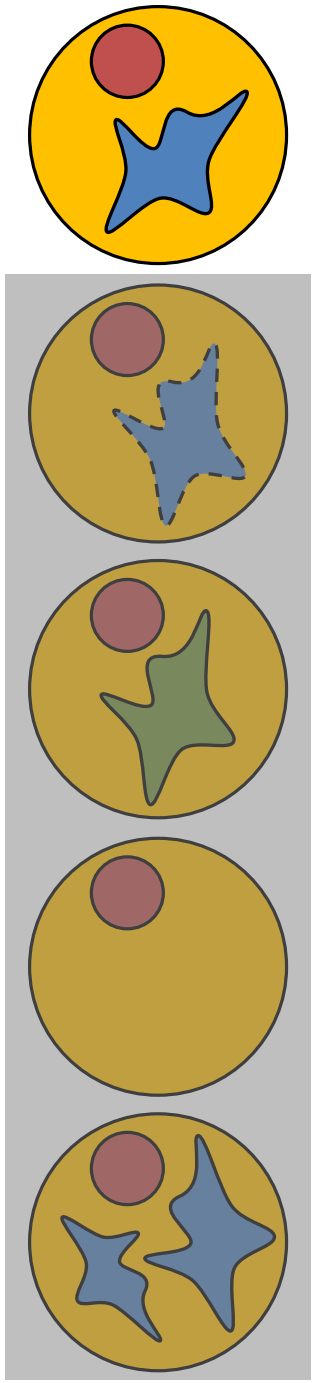
# Example Seurat Workflow



# Reading Data

```
Read10X("../filtered_feature_bc_matrix/") -> data

Read10X_h5("raw_feature_bc_matrix.h5") -> data

CreateSeuratObject(
     counts=data,
     project="course",
     min.cells = 3,
     min.features=200
) -> data
```

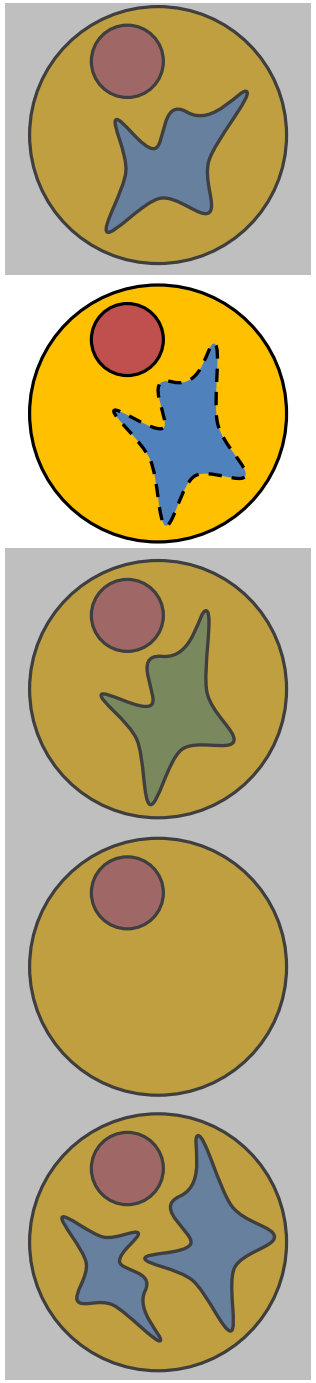
# QC – What problems are likely?



- Lysed cells
- Dead or dying cells
- Empty GEMs
- Double (or more) occupied GEMs
- Cells in different cell cycle stages

# Lysed Cells

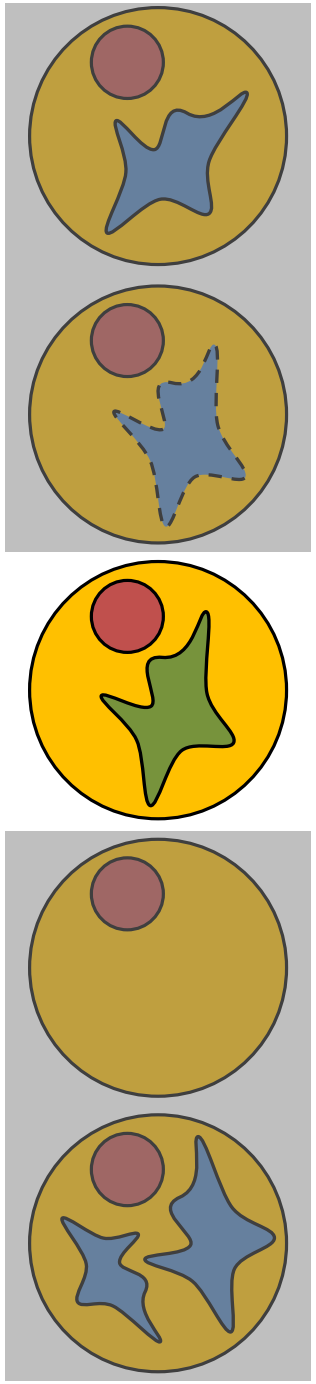
- Outer membrane is ruptured – cytoplasmic RNAs leak out
  - Loss of mature RNA, increase in pre-mRNA
    - Higher proportion mapping to introns
    - Loss of 3’ sequencing bias
  - Increase in nuclear RNAs
    - MALAT1 is an easy marker to use
  - Increase in Membrane associated transcripts
    - MS4A family
    - IL7R
    - Complement C3



# Dead or Dying Cells

- Cells undergoing apoptosis have very different transcriptomes
  - Lower total RNA production
  - Huge upregulation of mitochondrial transcription
  - Upregulation of caspases

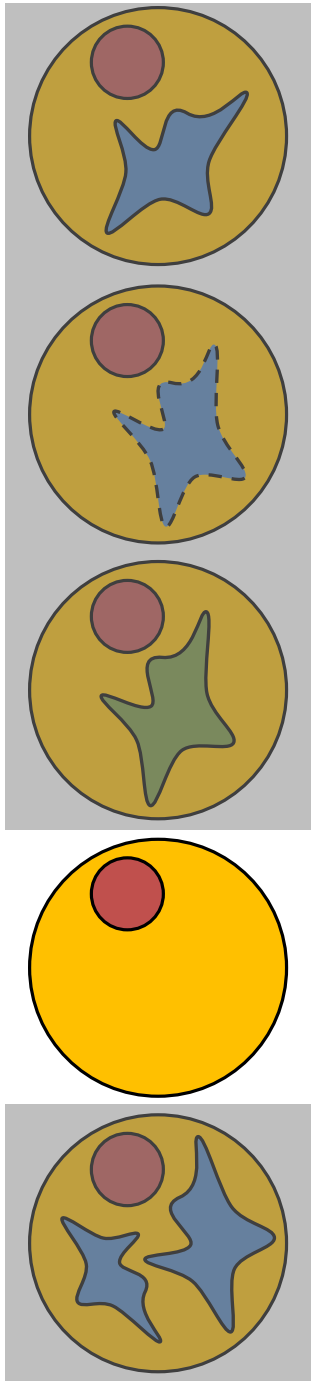
- Degraded transcripts are short
  - Read through into template switching oligo (seen earlier)



# Empty GEMs

- GEMs containing no cell will still produce some sequence
  - Background RNA in the flow medium
  - Will be worse with higher numbers of lysed cells

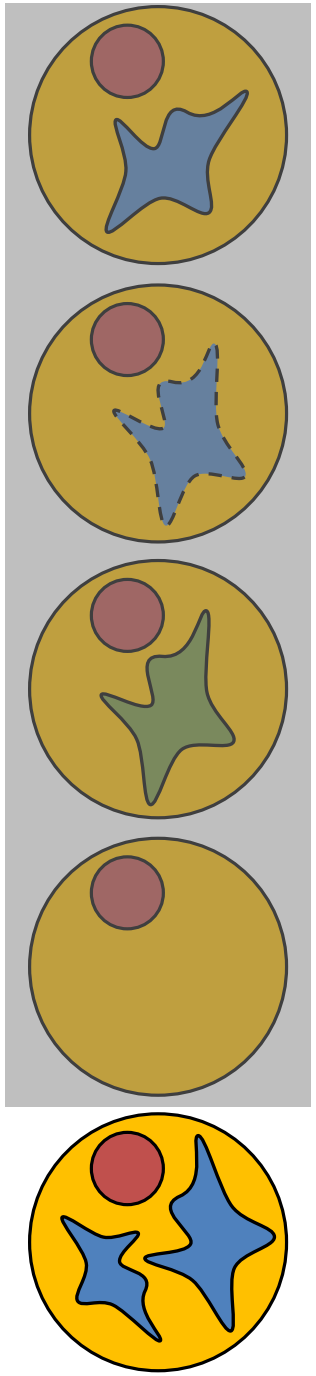
- Total amount of signal will be greatly reduced
- Will often look similar to each other (will cluster together)



# Double occupied GEMs

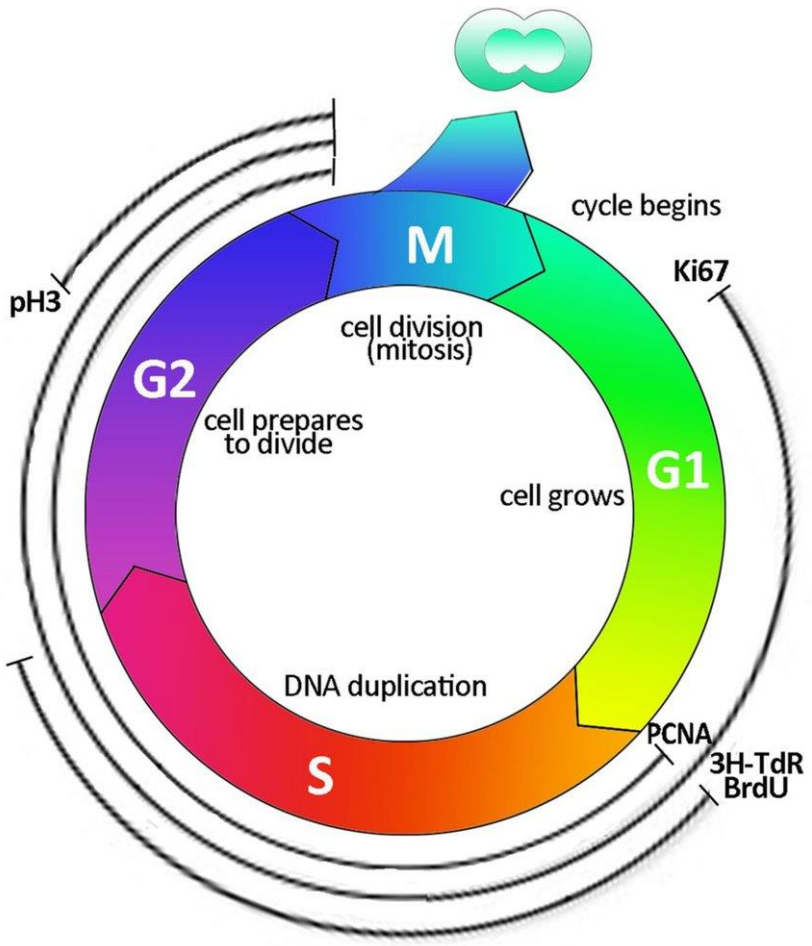
- Will get a mixed signal from two different cells

- Not as obvious a signal as empty GEMs
  - Greater diversity
  - More UMIs per cell
  - Intermediate clustering



# Cell Cycle Variation

- Cells in different stages of the cell cycle have quite different expression profiles
  - Use genes which classify different phases to classify cells in different phases
  - Exclude unusual cells
  - Attempt to include cell cycle as a factor during quantitation / differential expression

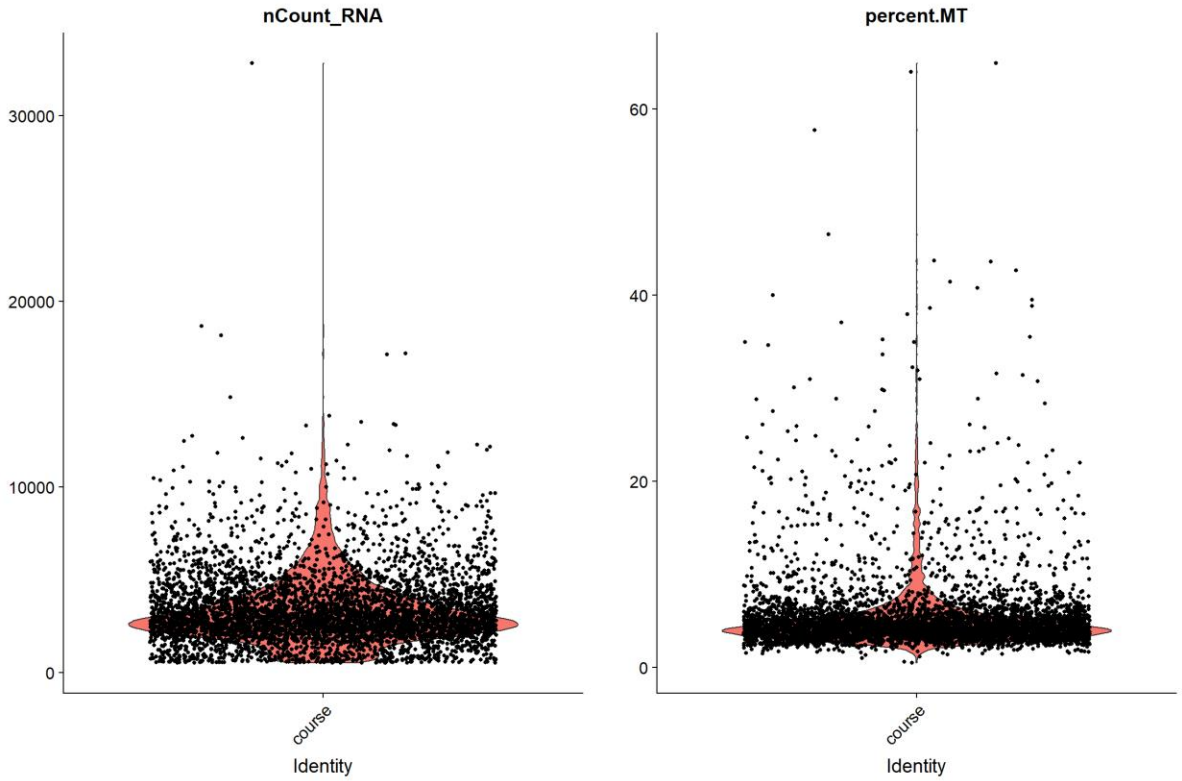



Lombardi, et al. (2015). 10.1093/ndt/gfv262.

# QC and Cell Filtering

- Standard QC Measures
  - Number of observed genes per cell
  - Number of reads per cell
  - Relationship between the two

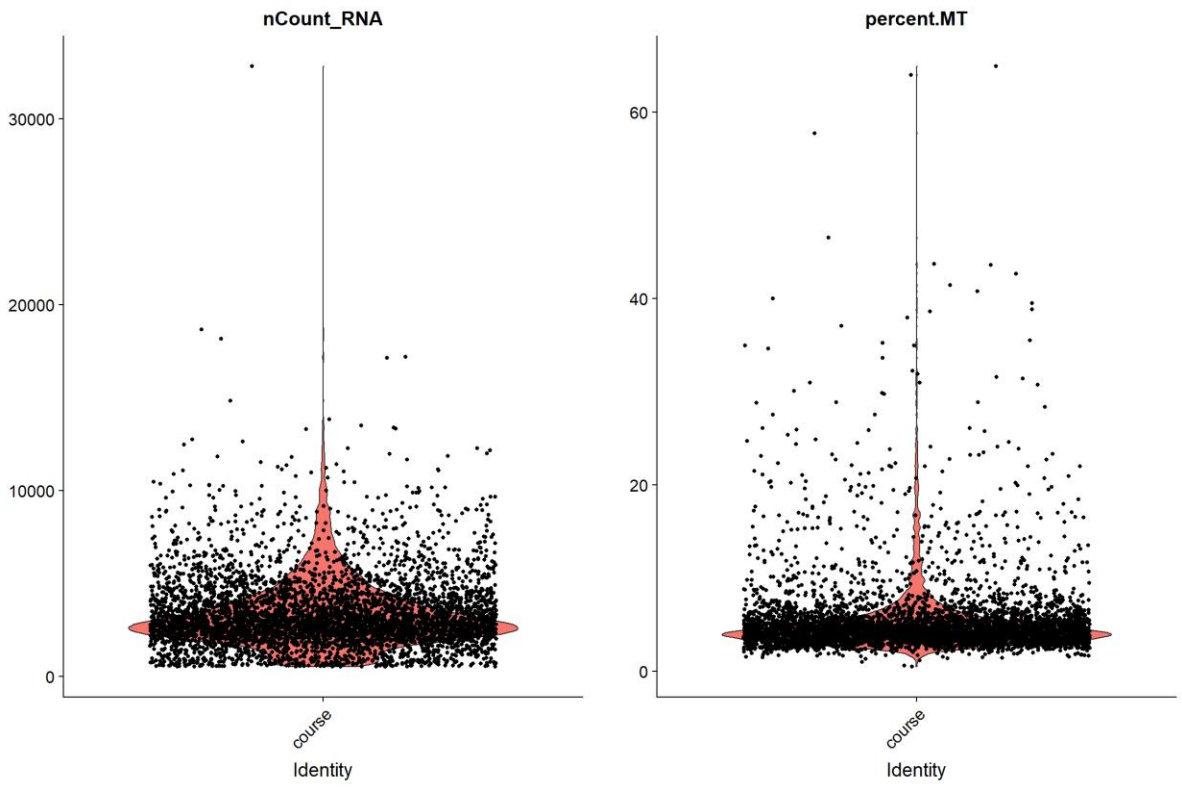
- Calculated QC Measures
  - Amount of mitochondrial reads
  - Amount of ribosomal reads
  - Marker genes (eg MALAT1)
  - Cell cycle



# QC and Cell Filtering

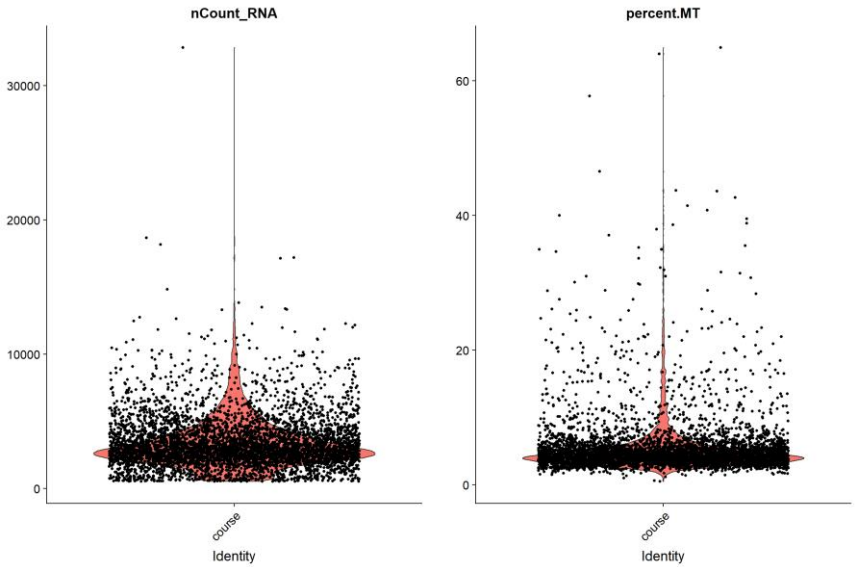
```
PercentageFeatureSet(
     data,
     pattern="^MT-“
) -> data$percent.MT
```

```
apply(
  data@assays$RNA@counts,
  2,
  function(x)(100*max(x))/sum(x)
) -> data$Percent.Largest.Gene
```

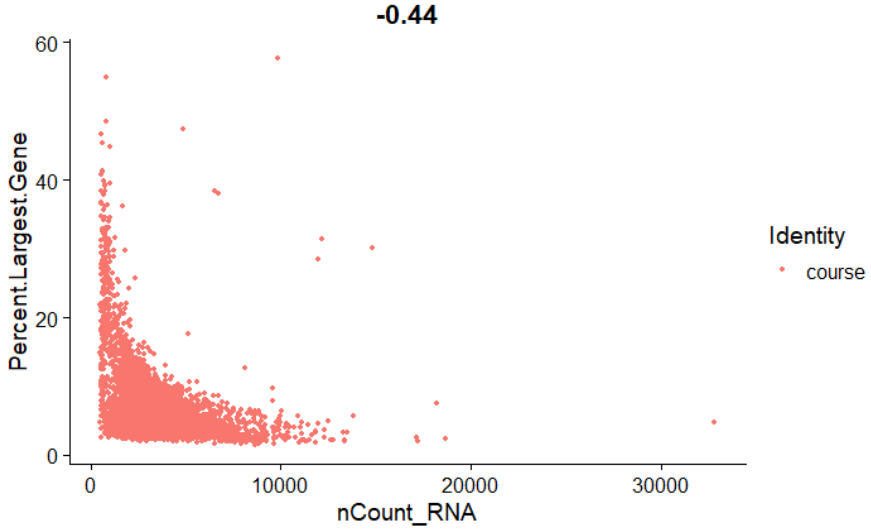


# QC and Cell Filtering

```
vlnPlot(
        data,
        features=c("nCount_RNA","percent.MT")
)
```

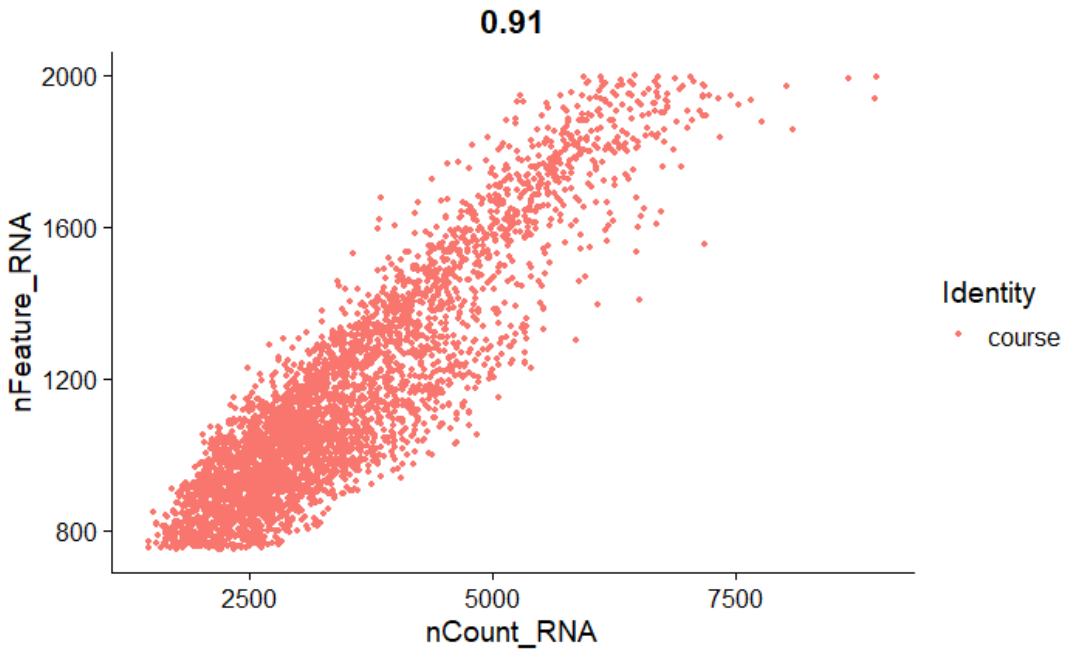



```
FeatureScatter(
        data,
        feature1 = "nCount_RNA",
        feature2 = "Percent.Largest.Gene“
)
```



# QC and Cell Filtering

```
subset(
  data,
  nFeature_RNA>750 &
    nFeature_RNA < 2000 &
    percent.MT < 10 &
    Percent.Largest.Gene < 20
) -> data
```



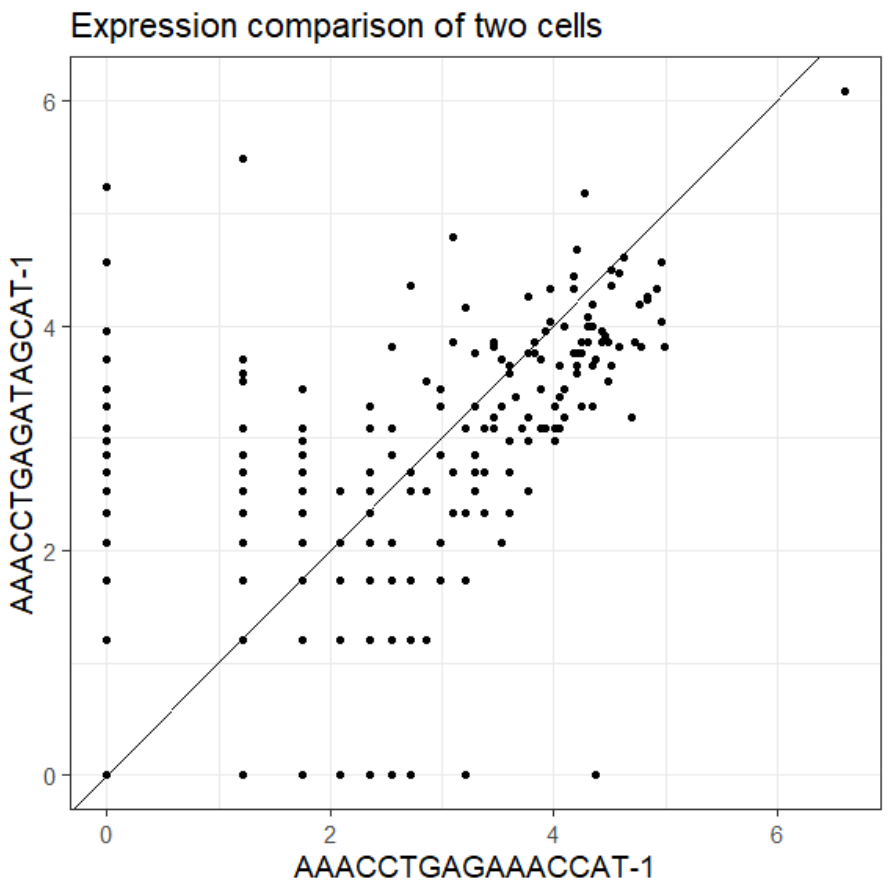
# Count Normalisation and Scaling

- Raw counts are biased by total reads per cell
- Counts are more stable on a log scale
- Standard normalisation is just log reads per 10,000 reads
- Can use an additional centring step which may help
  - Similar to size factor normalisation in conventional RNA-Seq

- For PCA counts scale each gene's expression to a z-score
  - Can also use this step to try to regress out unwanted effects

# Count Normalisation and Scaling

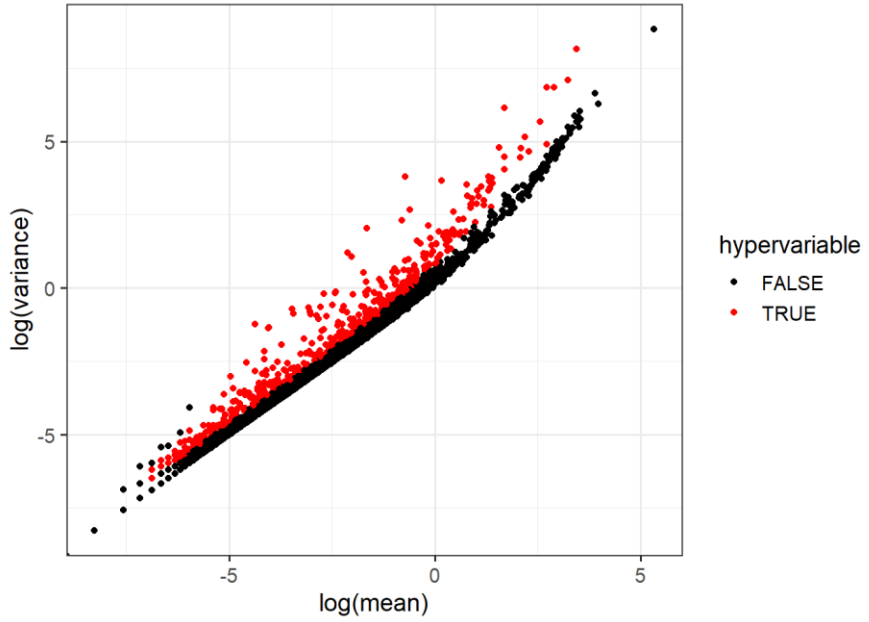
```
NormalizeData(
    data,
    normalization.method = "CLR"
) -> data


ScaleData(
    data,
    features=rownames(data)
) -> data
```



# Variable Feature Selection

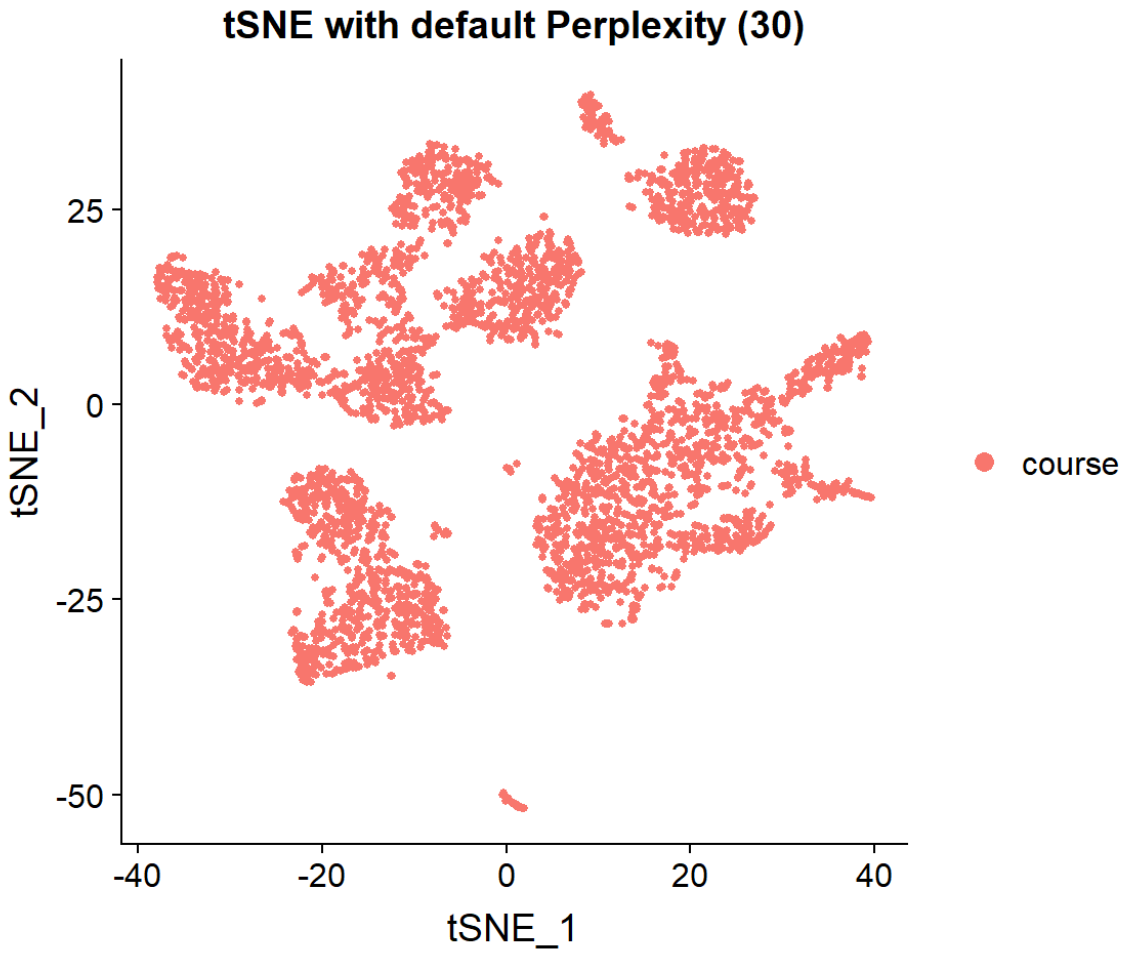
- Selects a subset of genes to use for downstream analysis
- Identify genes with an unusual amount of variability
- Link the variability with the expression level to find variation which is high in the context of the expression level
- Keep only the most variable genes

```
FindVariableFeatures(
  data,
  selection.method = "vst",
  nfeatures=500
) -> data
```



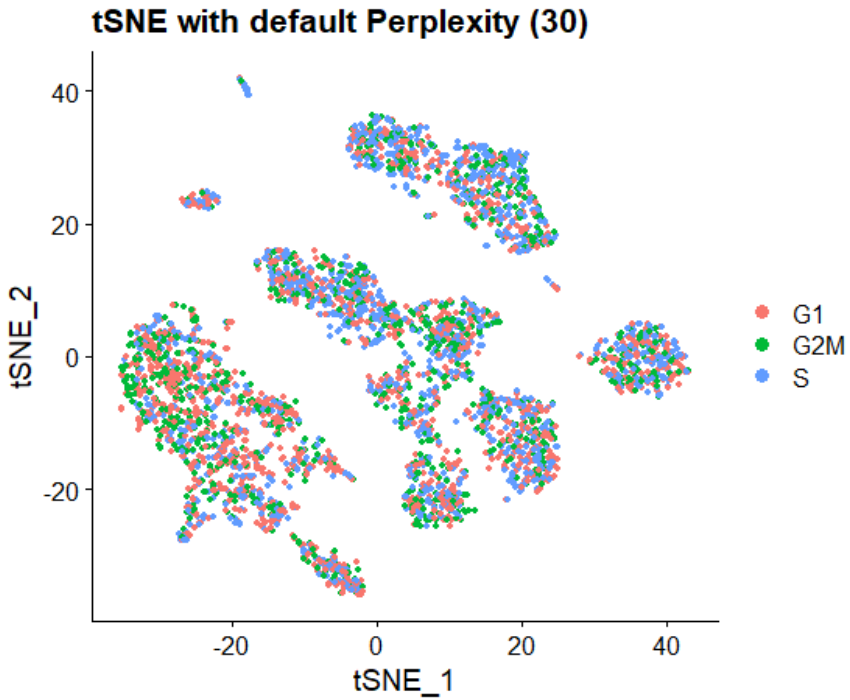
# Dimensionality Reduction

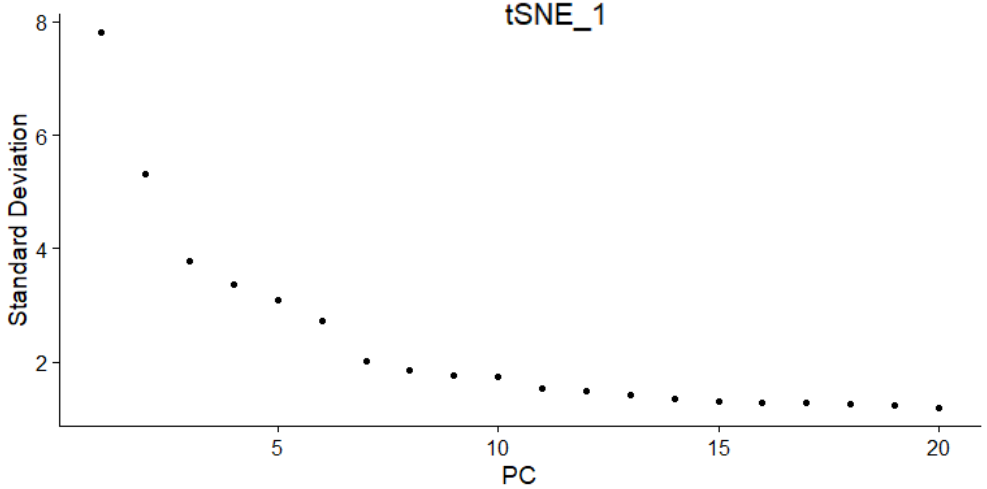
- Start with PCA on the normalised, filtered (both cells and genes), scaled data
- Scree / Elbow plot to decide how many PCs are informative
- Pass only the interesting PCs to subsequent tSNE or UMAP reduction to get down to 2 dimensions



# Dimensionality Reduction

```
RunPCA(
     data,
     features=VariableFeatures(data)
) -> data


RunTSNE(
  data,
  dims=1:15,
  seed.use = saved.seed,
  perplexity=30
) -> data
```



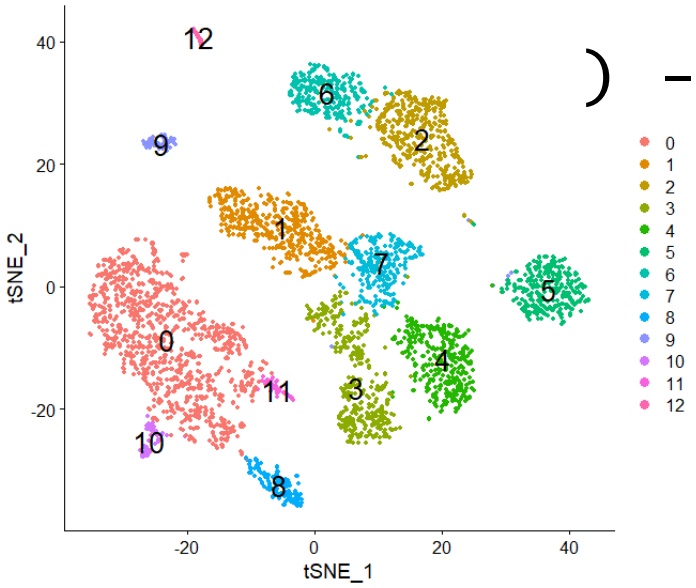



# Defining clusters

- Construct nearest neighbour graph (can specify how many neighbours)
  - Constructed from PCA
  - Normally use the same number of dimensions as for tSNE/UMAP

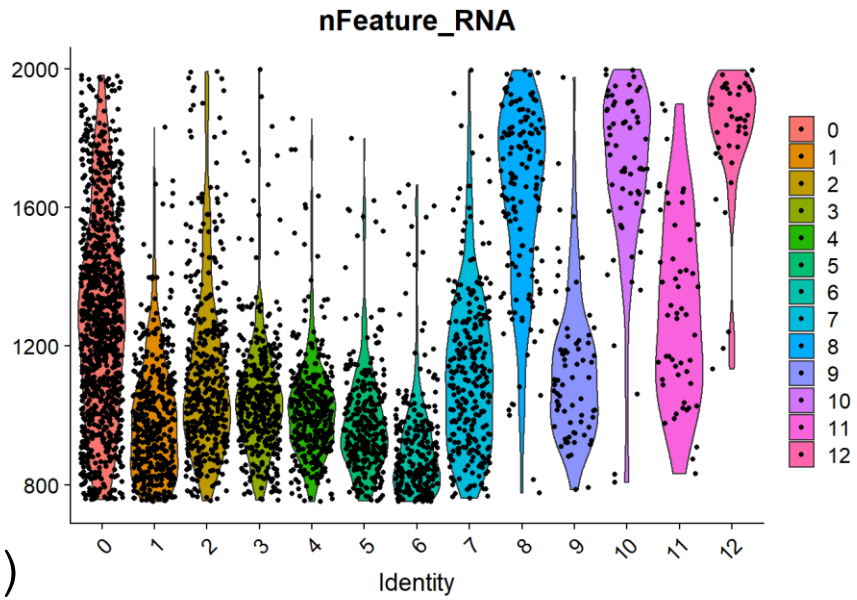
```
FindNeighbors(
      data,
      dims=1:15
) -> data

FindClusters(
      data,
      resolution = 0.5
) -> data
```

- Find local clusters
  - All cells are classified
  - Graph Based Clustering (Louvain method)
  - Resolution defines granularity



# Comparing Properties of Clusters

- We want to know that clusters are occurring because of biological changes, not technical differences
- We can plot out the aggregate QC metrics for clusters
  - Read/Gene counts
  - Mitochondrion
  - MALAT1



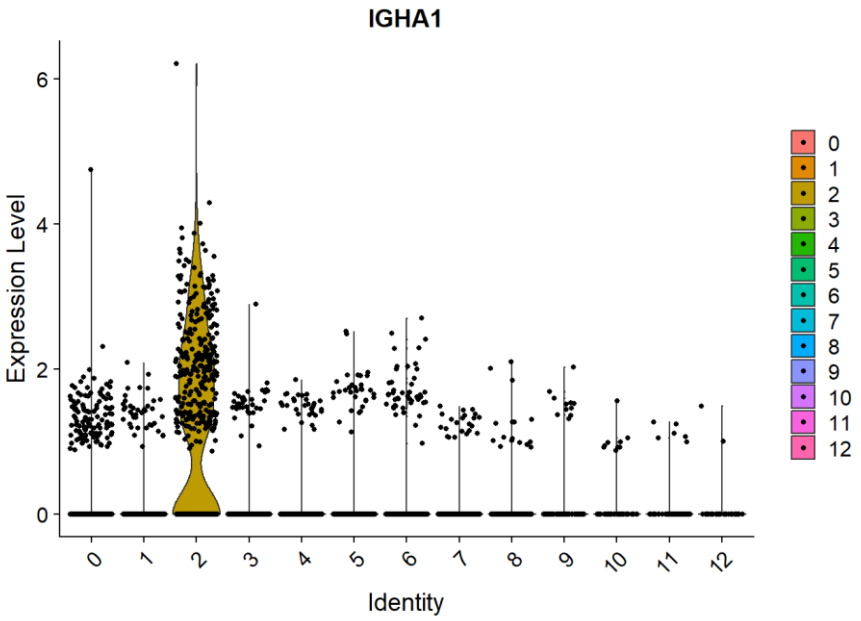

```
VlnPlot(data,features="nFeature_RNA")
```

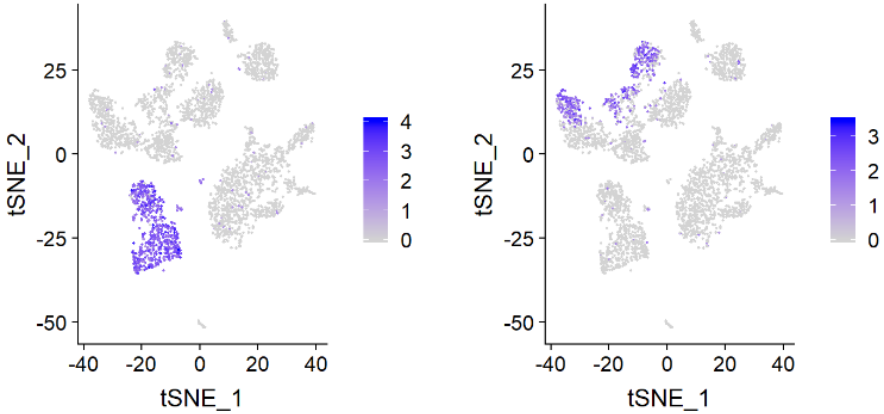
# Statistical analysis of differences between clusters

- Different types of hits
  - Quantitatively significant between clusters
  - Qualitatively different (predictive) of cluster membership
- Different type of markers
  - Global: Distinguish one cluster from all of the rest of the data
  - Local: Distinguish one cluster from another defined set of clusters
- Often filter genes based on coverage in the set or the size of groups
- Several choices of method to identify genes

# Statistical analysis of differences between clusters

- Non-parametric
  - Wilcox rank sum test
- Parametric
  - T-test
  - Negative binomial (eg DESeq)
- Classification
  - ROC analysis
- Specialised
  - MAST




```
FindMarkers(
        data,
        ident.1 = 2,
        ident.2 = 6,
        test.use = "roc",
        only.pos = TRUE
)
```

Wang *et al. BMC Bioinformatics* (2019) 20:40
https://doi.org/10.1186/s12859-019-2599-6

BMC Bioinformatics

RESEARCH ARTICLE Open Access

# Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data

Tianyu Wang[1], Boyang Li[2], Craig E. Nelson[3] and Sheida Nabavi[4*]

**Conclusions:** In general, agreement among the tools in calling DE genes is not high. There is a trade-off between true-positive rates and the precision of calling DE genes. Methods with higher true positive rates tend to show low precision due to their introducing false positives, whereas methods with high precision show low true positive rates due to identifying few DE genes. We observed that current methods designed for scRNAseq data do not tend to show better performance compared to methods designed for bulk RNAseq

# Automated Cell Assignment

- Can automatically assign cell identities to clusters
- Need a source of marker genes
  - Result of a previous run/experiment
  - Publicly available data
  - Biggest hurdle
- Many packages to do this
  - We use SCINA in the exercise

Abdelaal *et al. Genome Biology* (2019) 20:194
https://doi.org/10.1186/s13059-019-1795-z

Genome Biology

RESEARCH Open Access

A comparison of automatic cell identification methods for single-cell RNA sequencing data

Tamim Abdelaal[1,2†], Lieke Michielsen[1,2†], Davy Cats[3], Dylan Hoogduin[3], Hailiang Mei[3], Marcel J. T. Reinders[1,2] and Ahmed Mahfouz[1,2*]

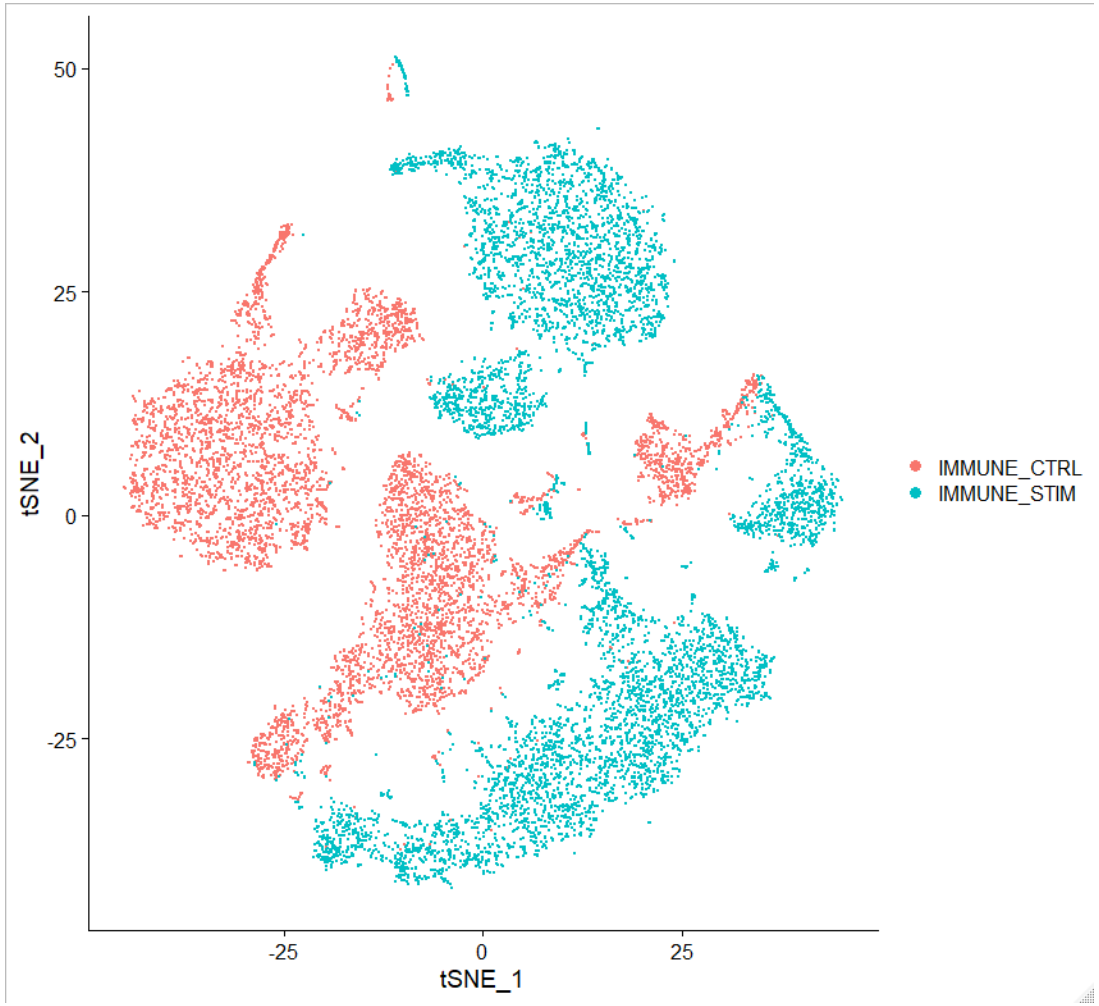
# Integrating Multiple Runs

- When multiple runs are combined (eg Unstim and Stim), the batch differences between the runs can overwhelm the biological differences

- Raw comparisons can therefore miss changes between what are actually matched subgroups

# Raw merged runs

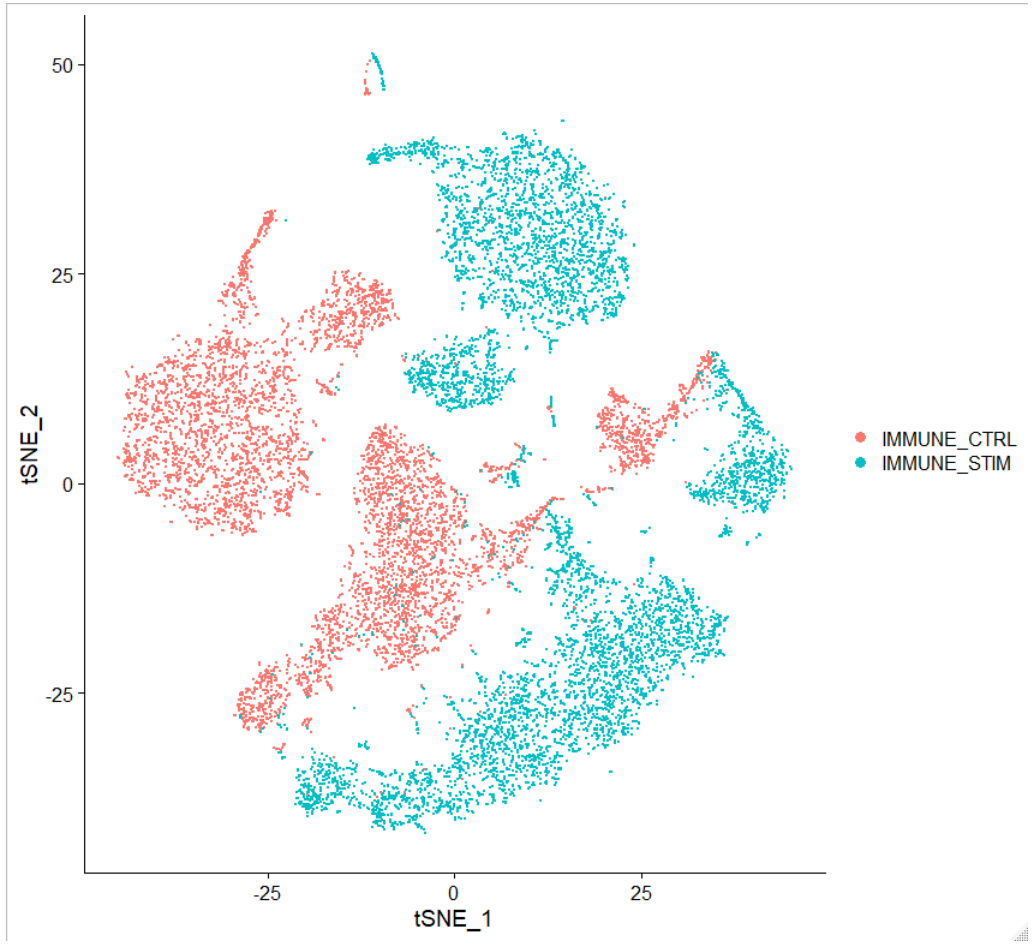
- Two PBMC populations run at different times
- tSNE spread coloured by library
- Little to no overlap between cell populations



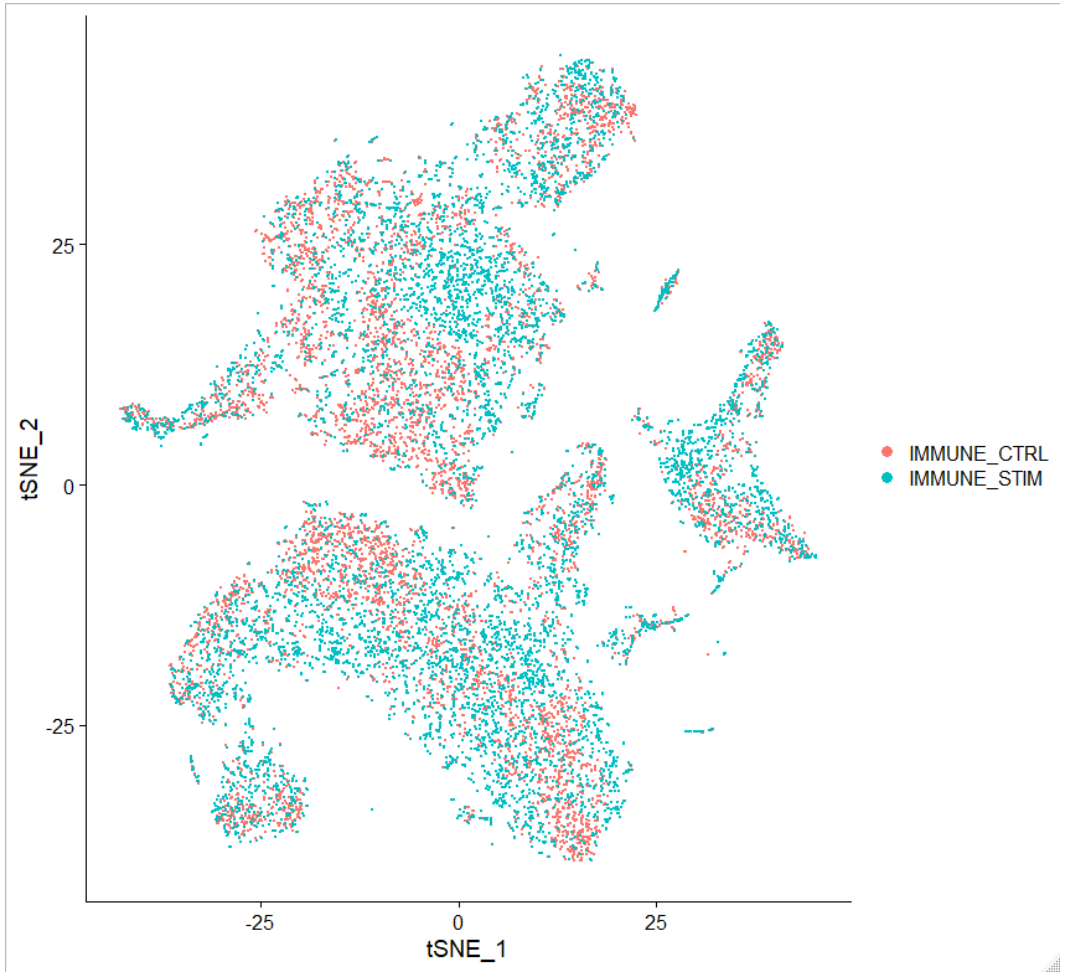
# Anchoring Runs

- Method to try to re-align different runs
- Uses mutual nearest neighbour searches between runs to pair up cells
- Uses pairs to align the dimension reduction plots

# Anchoring Runs

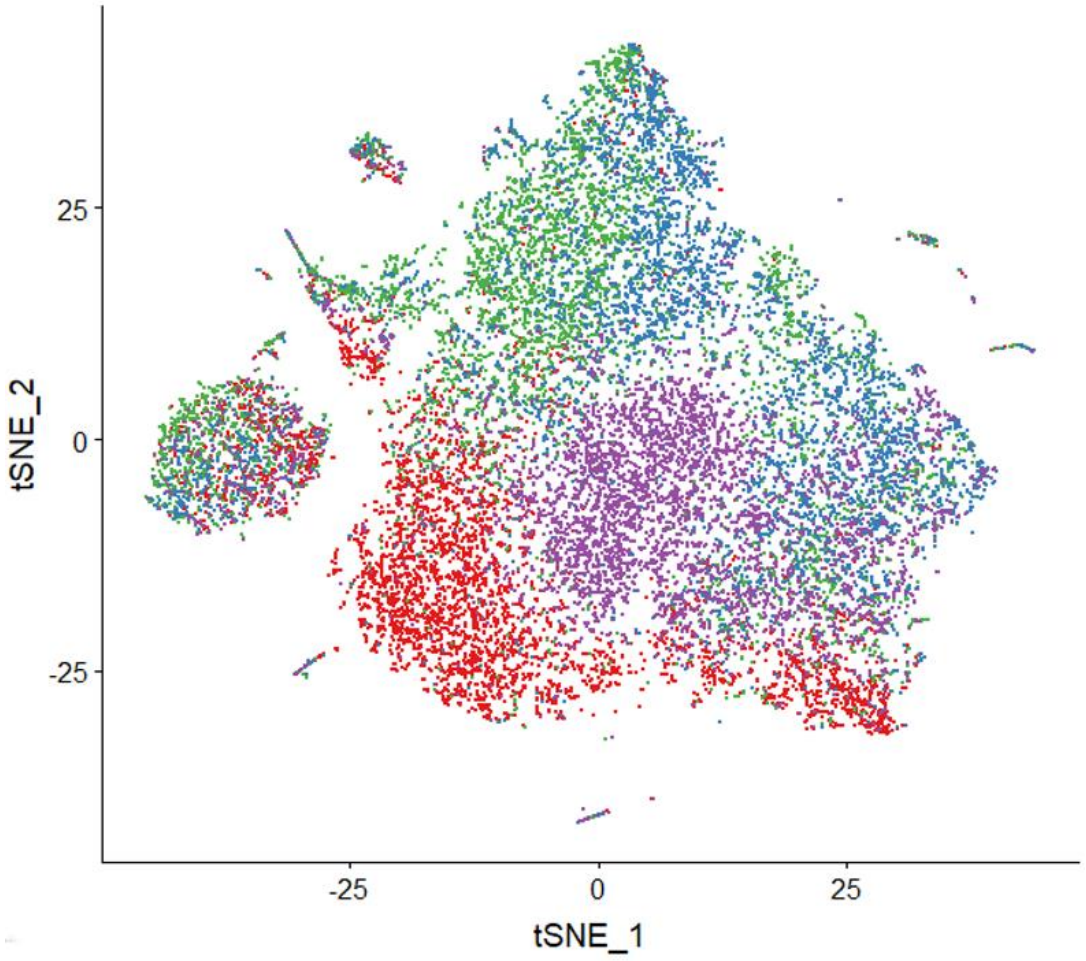


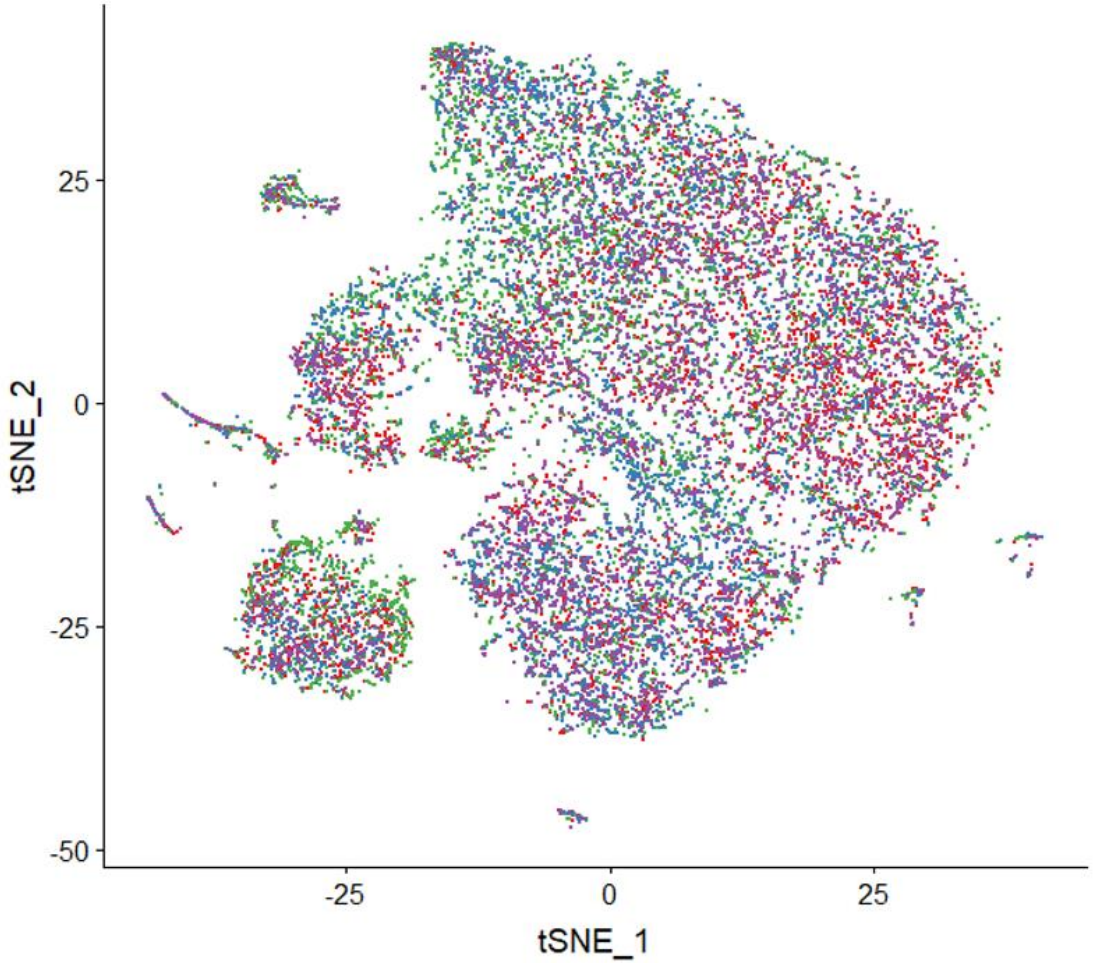

Raw




Anchored

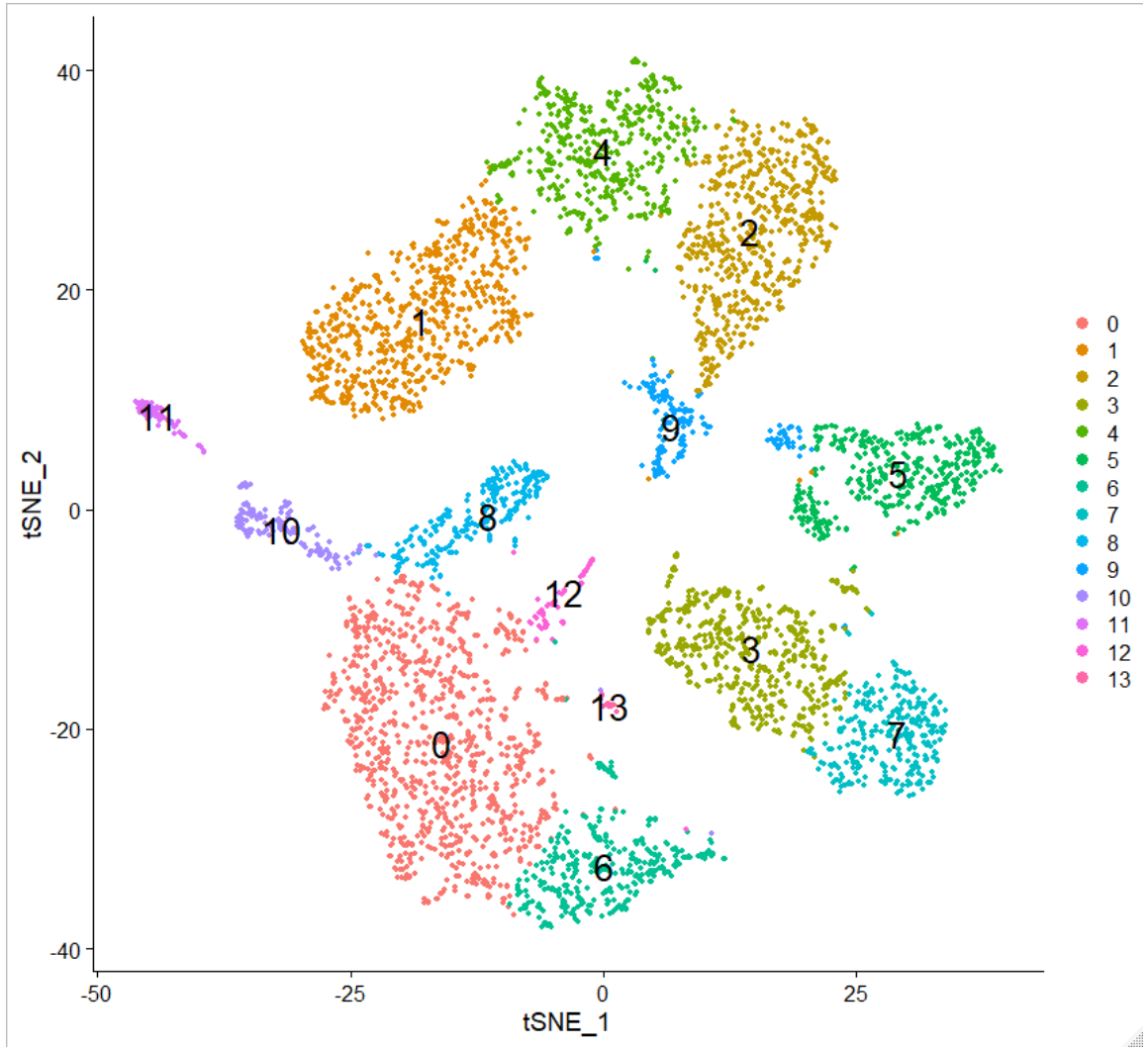
# Over-Anchoring



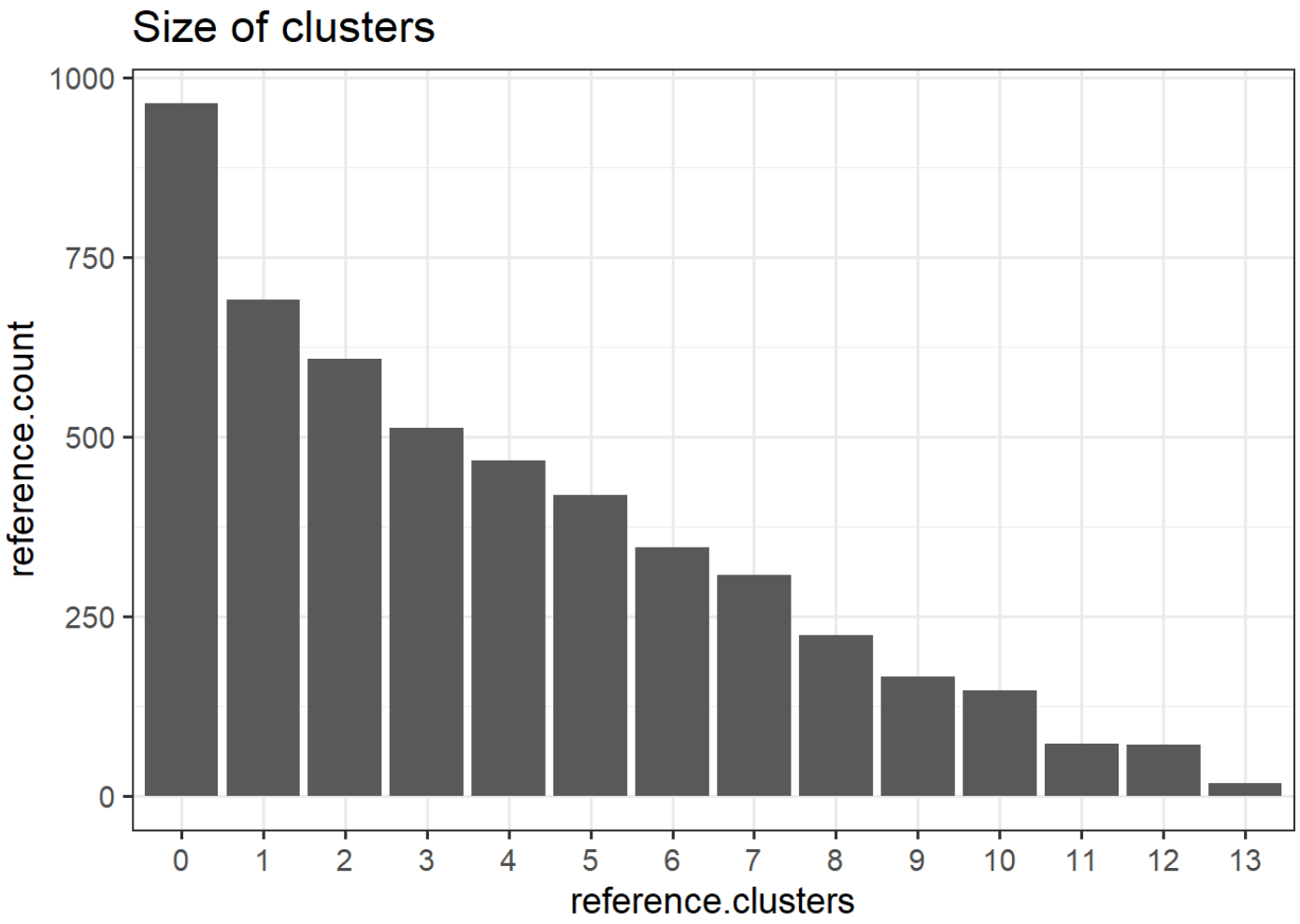
# Exercise – Using Seurat to analyse 10X data

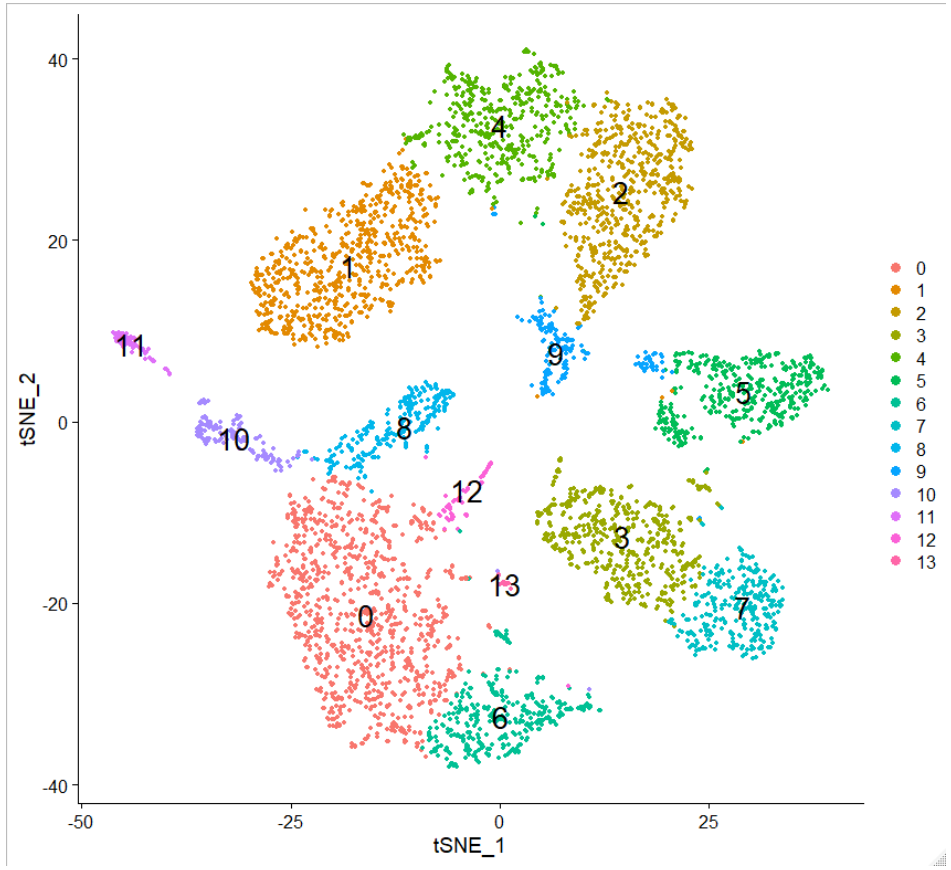
# Which factors matter?



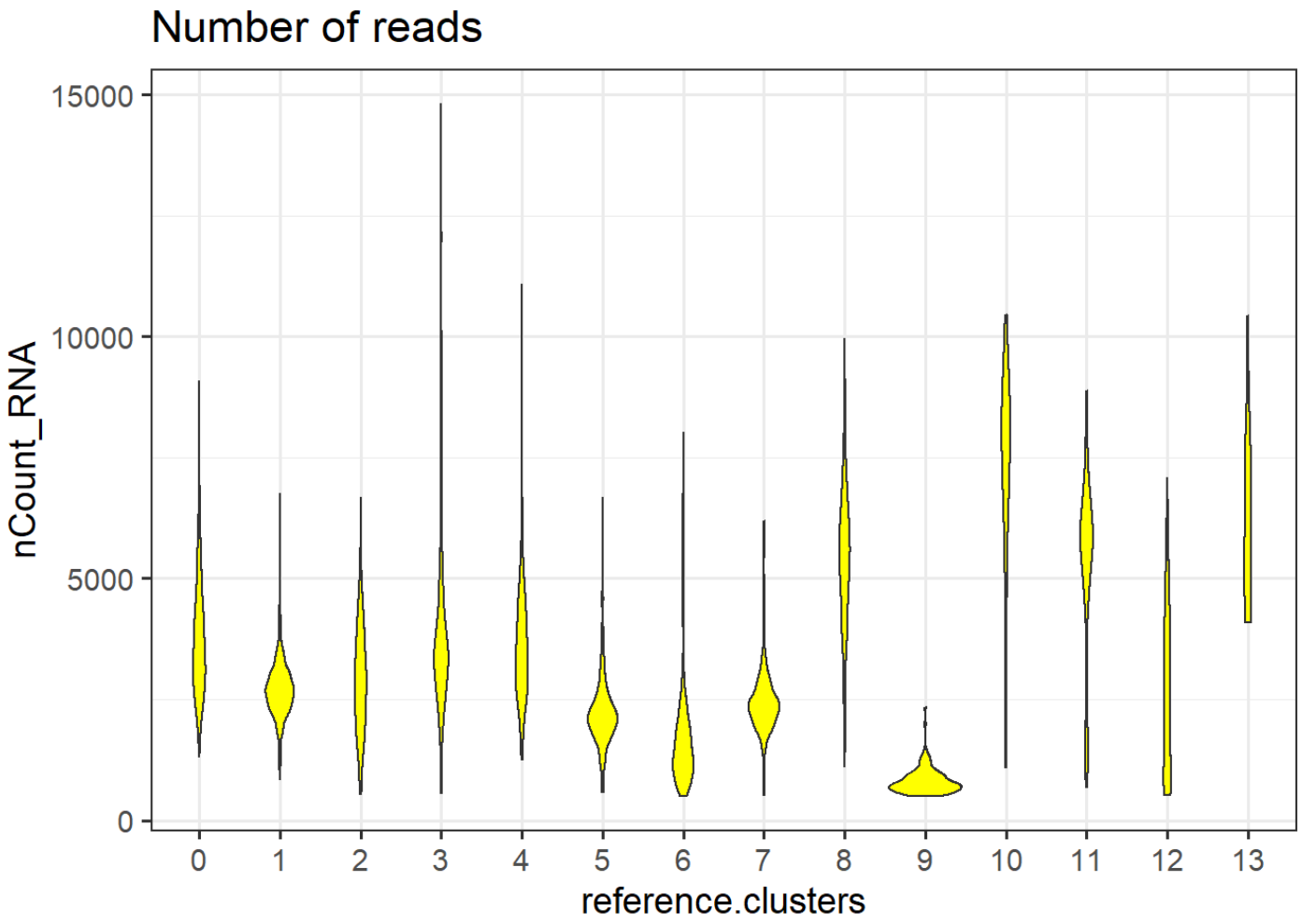

```
run.seurat.analysis <- function (
  data,
  number.of.genes.min = 200,
  number.of.genes.max = 2500,
  percent.mito = 100,
  percent.ribo = 100,
  normalise.method = "LogNormalize",
  number.variable.features = 2000,
  apply.scaling = TRUE,
  pcs.to.keep = 10,
  cluster.resolution = 0.5,
  remove.mito = FALSE,
  remove.ribo = FALSE) {
```

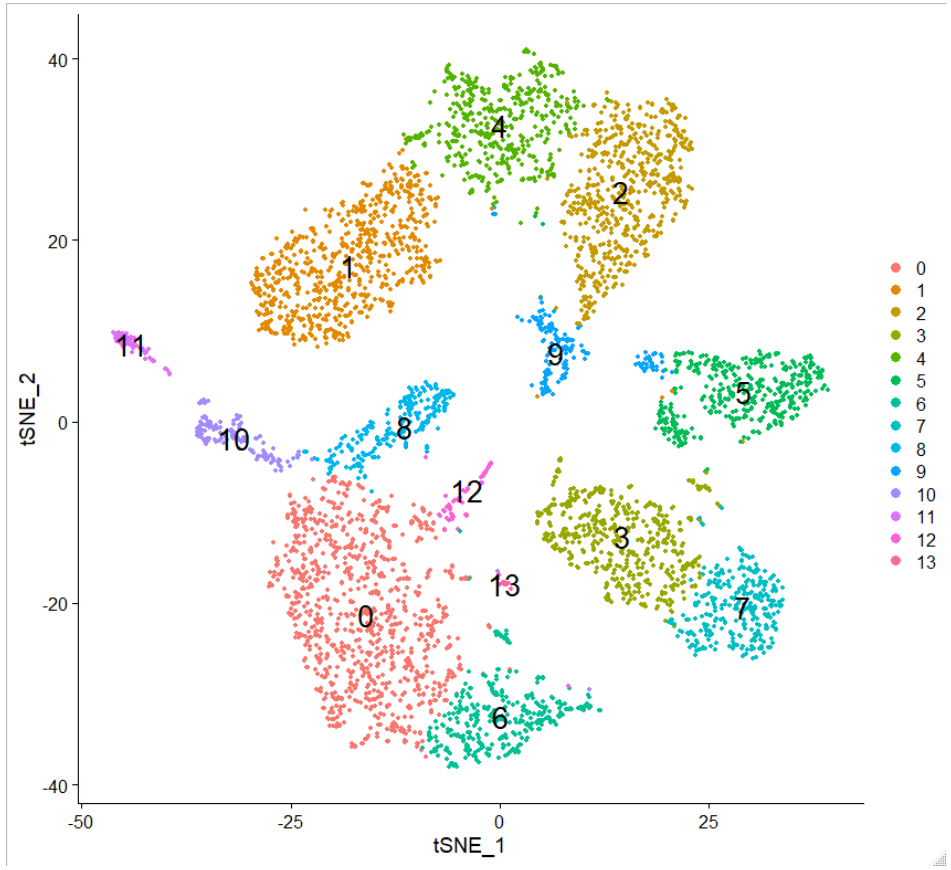
# Cluster Properties - Cluster Sizes



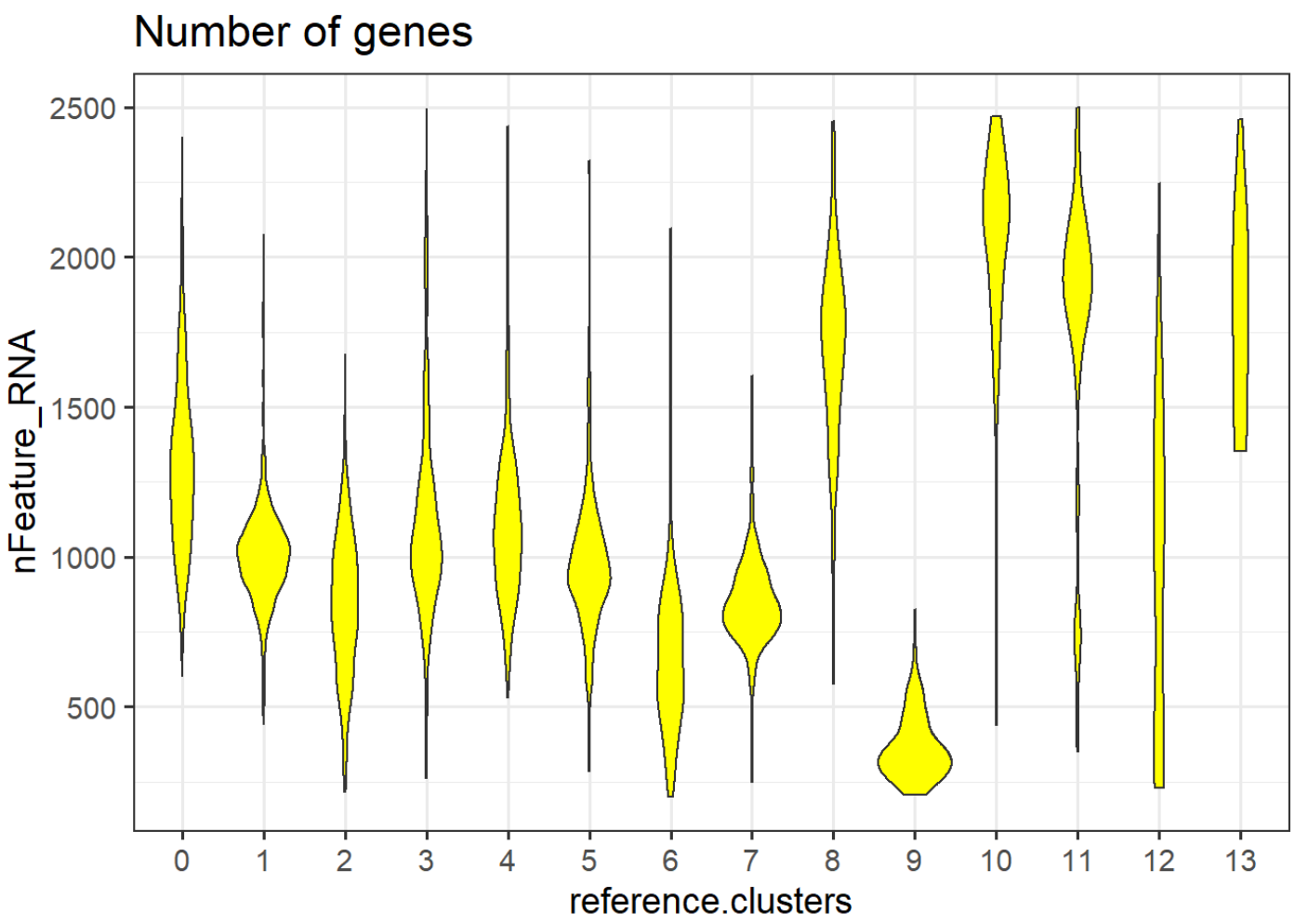



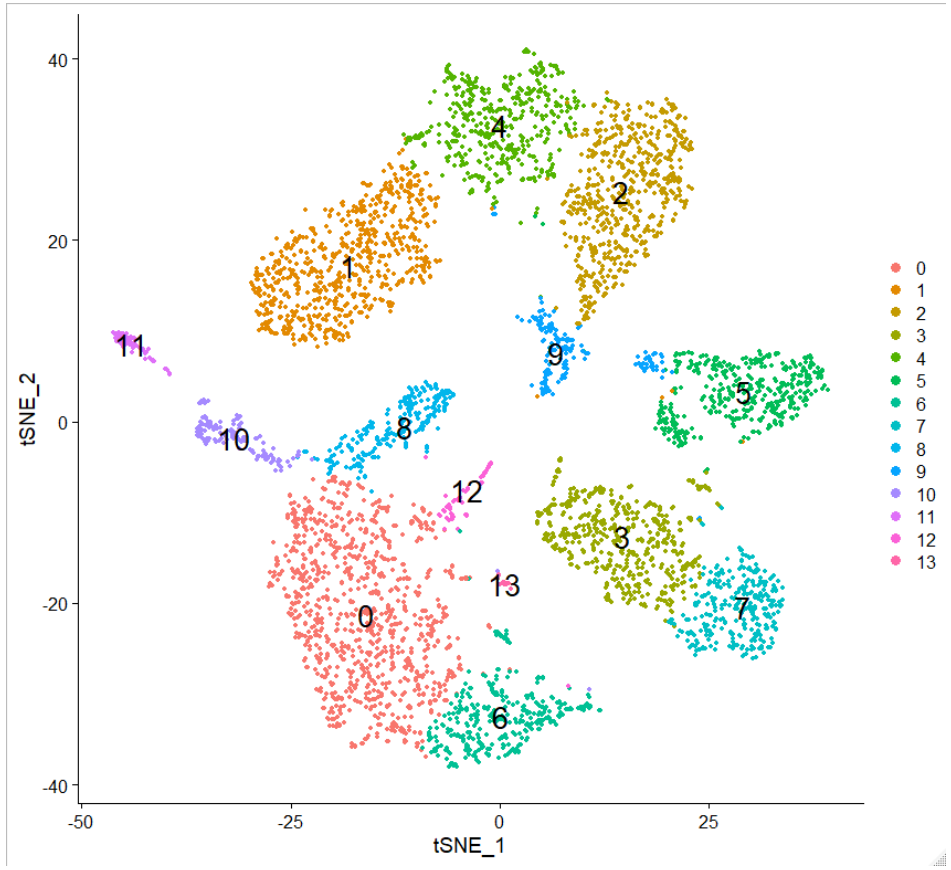
# Cluster Properties – Reads per cell



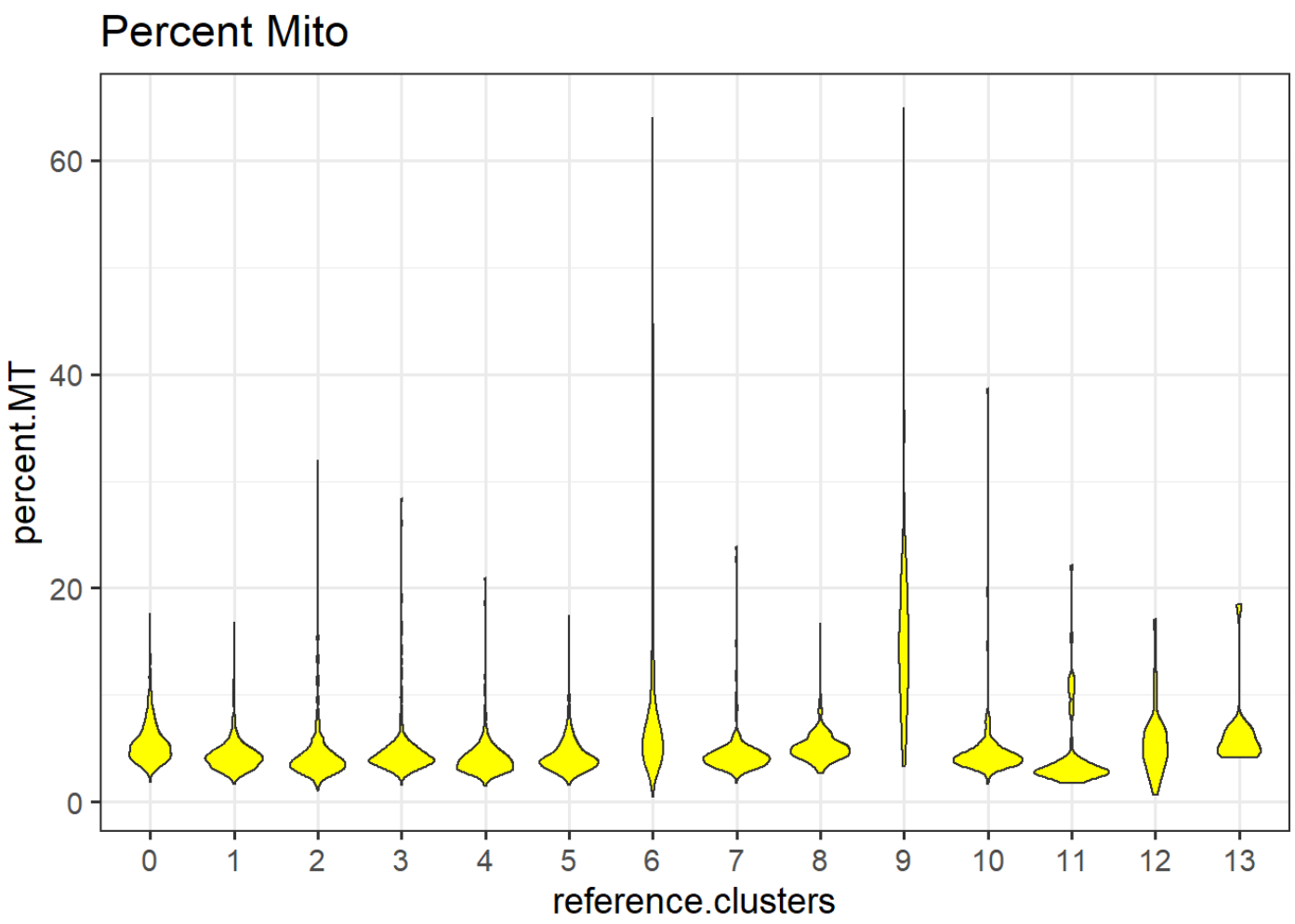



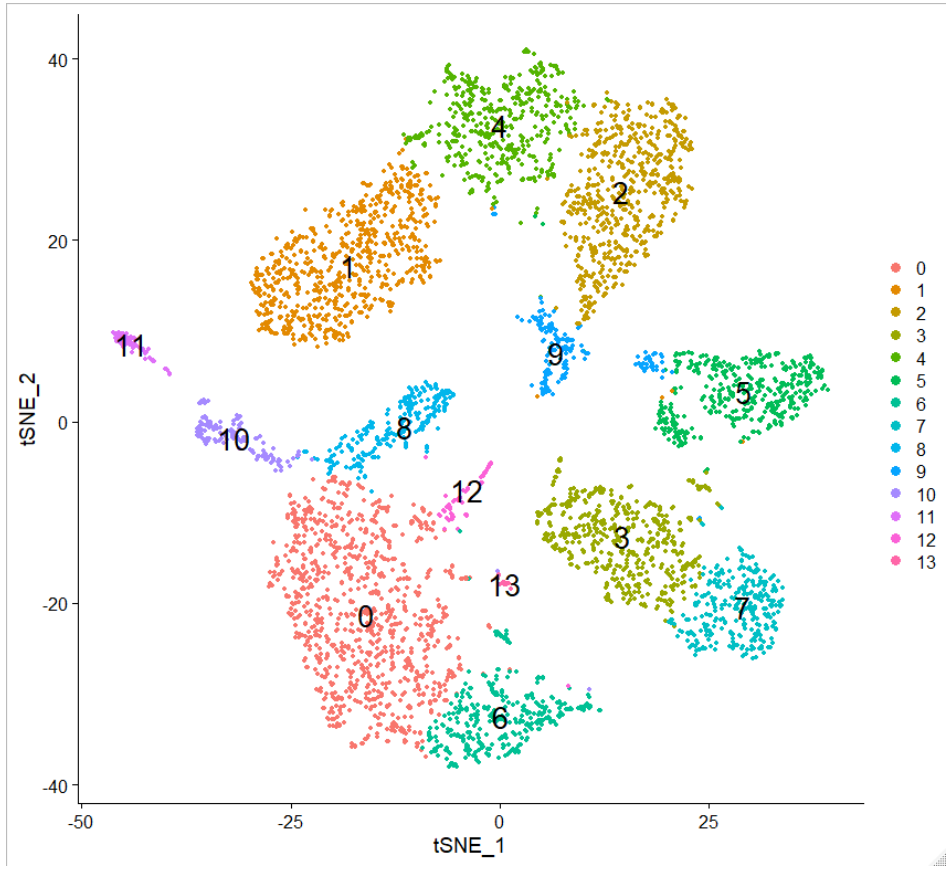
# Cluster Properties – Genes per cell



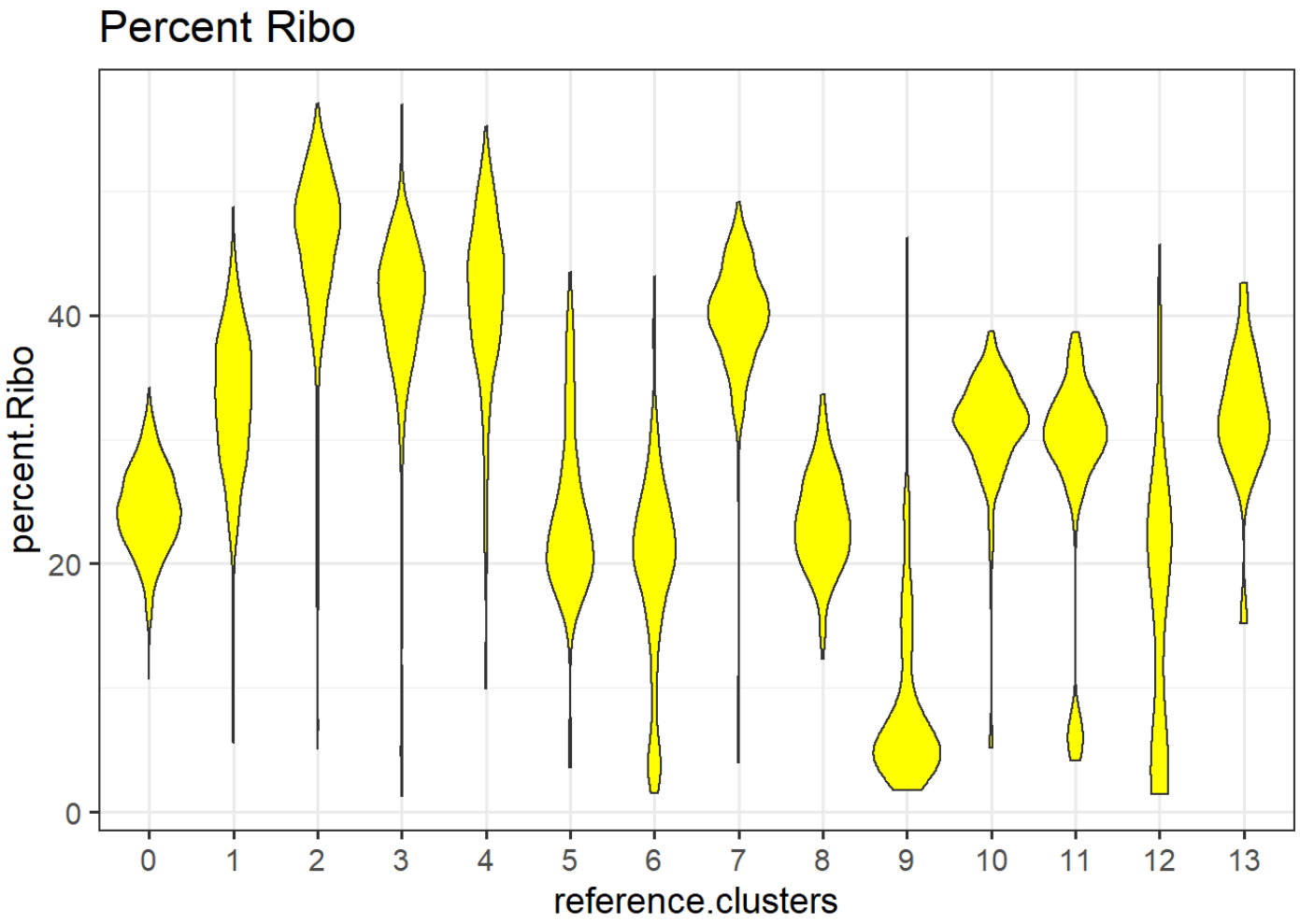



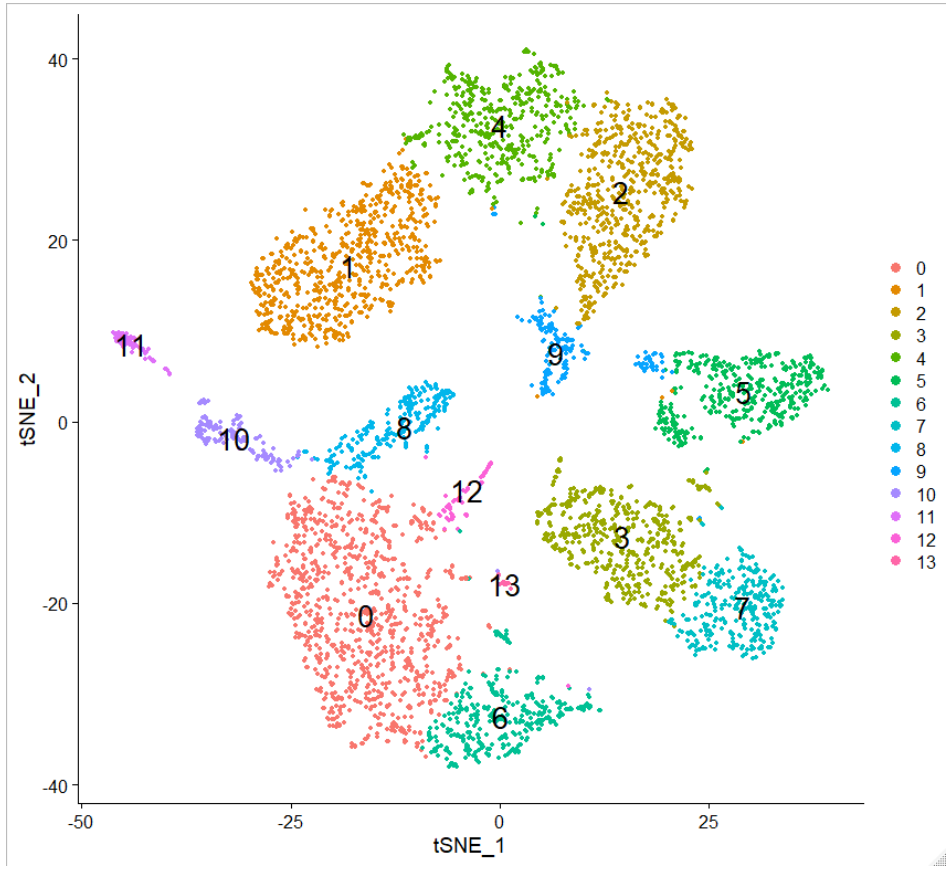
# Cluster Properties – Amount of MT



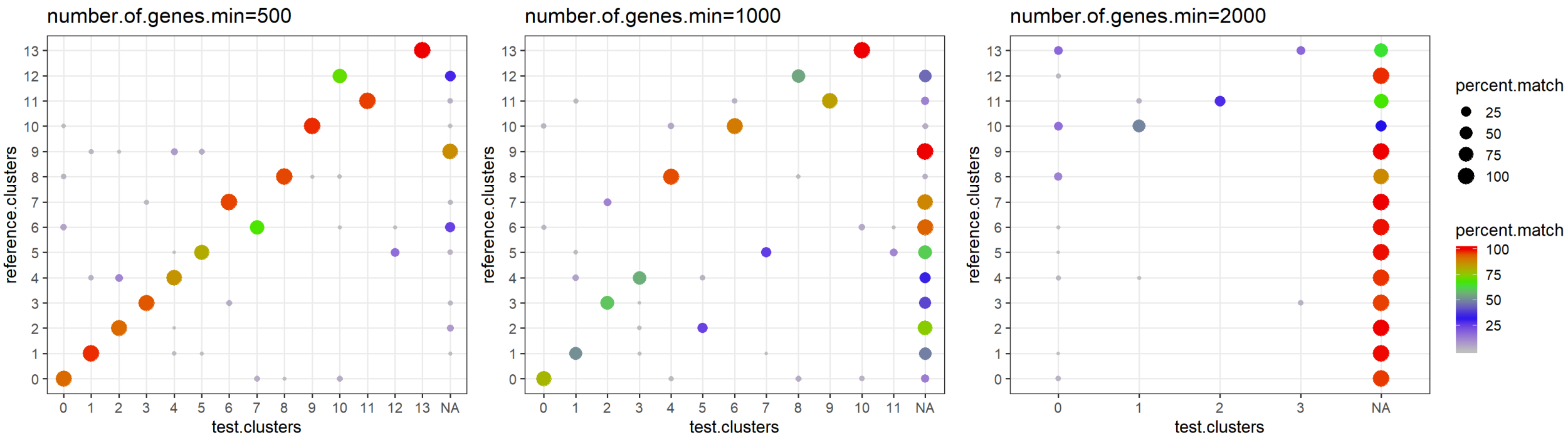



# Cluster Properties – Amount of Ribosomal

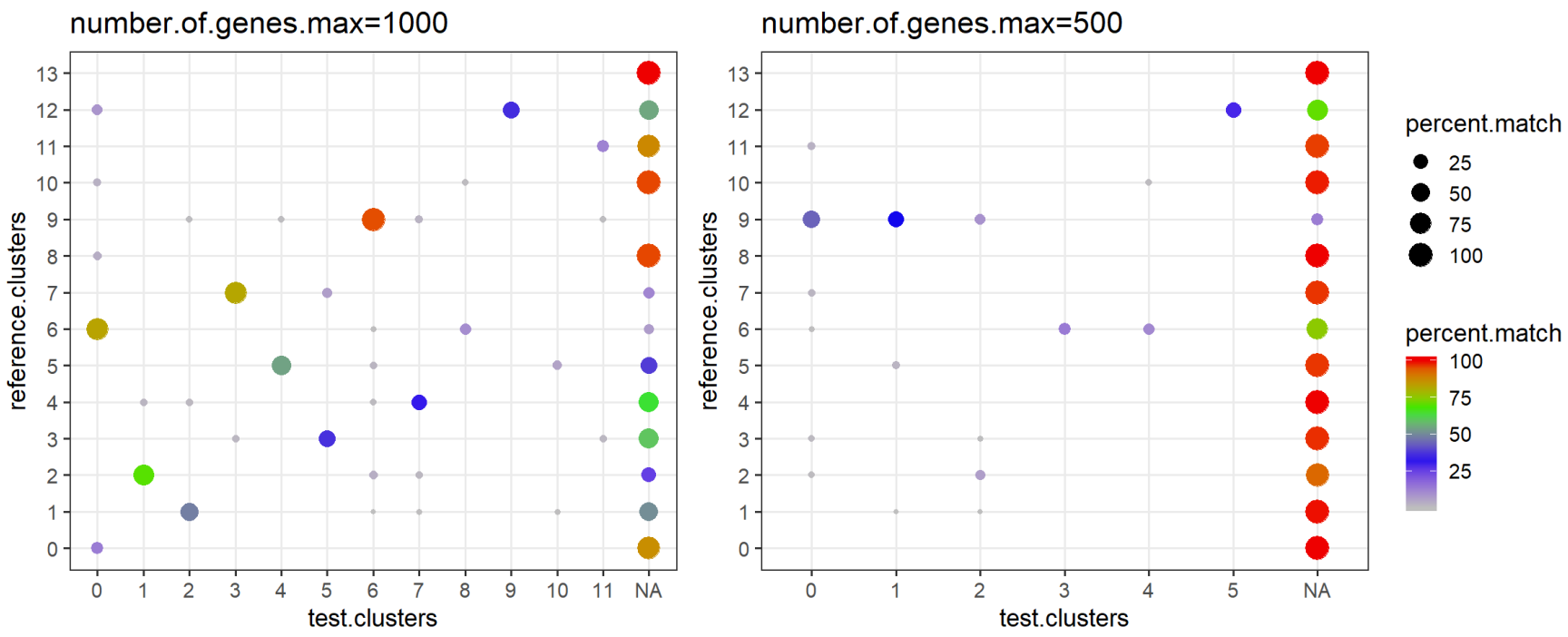


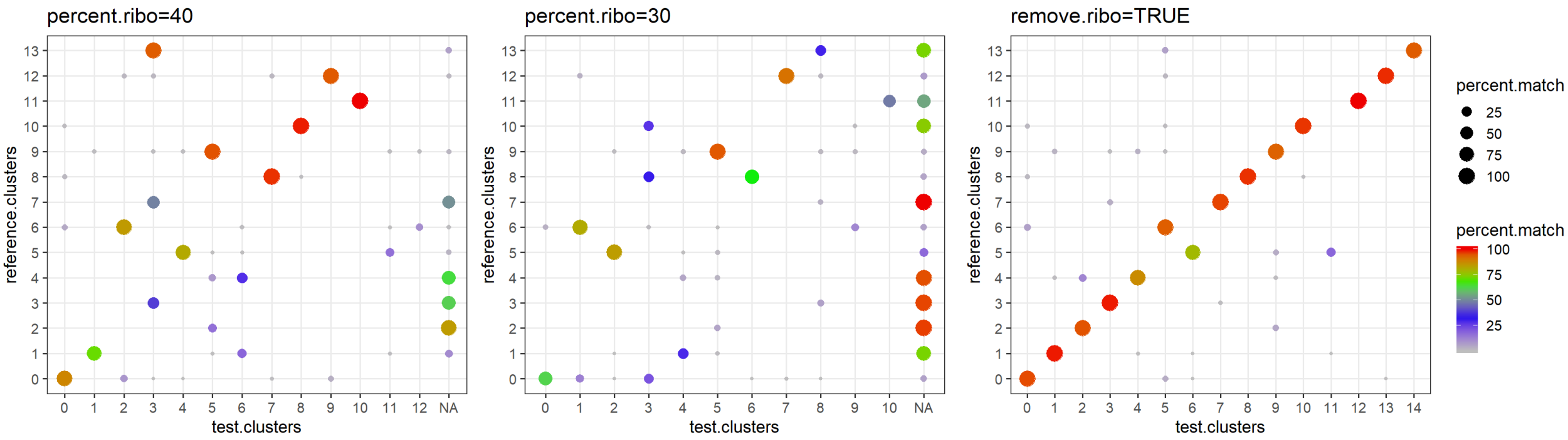
# Varying Parameters – Min Genes per Cell (200)



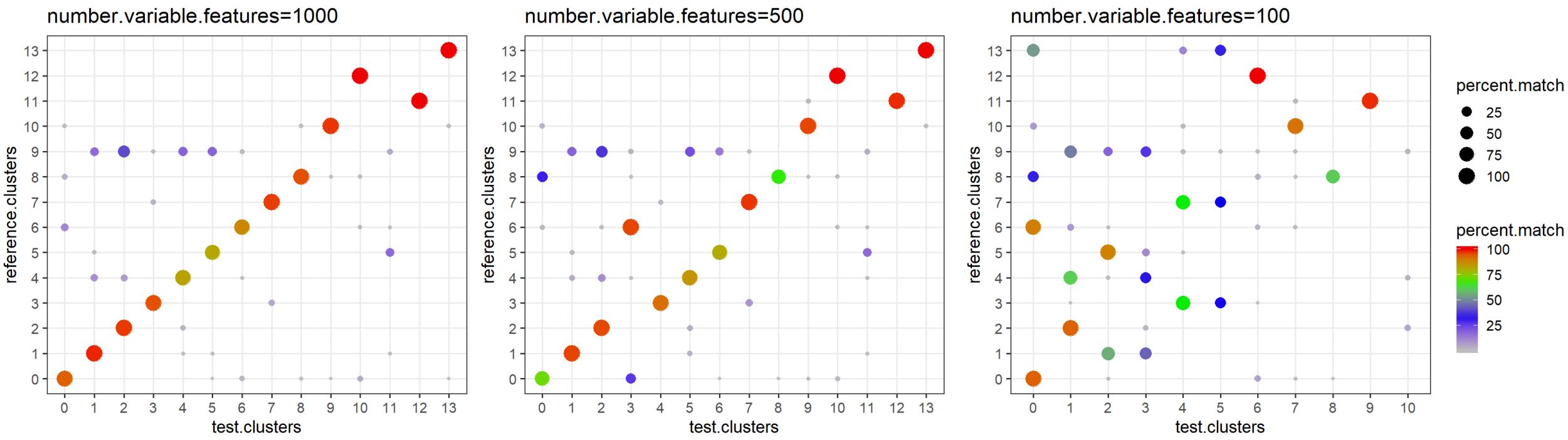
# Varying Parameters – Max Genes per Cell (2500)



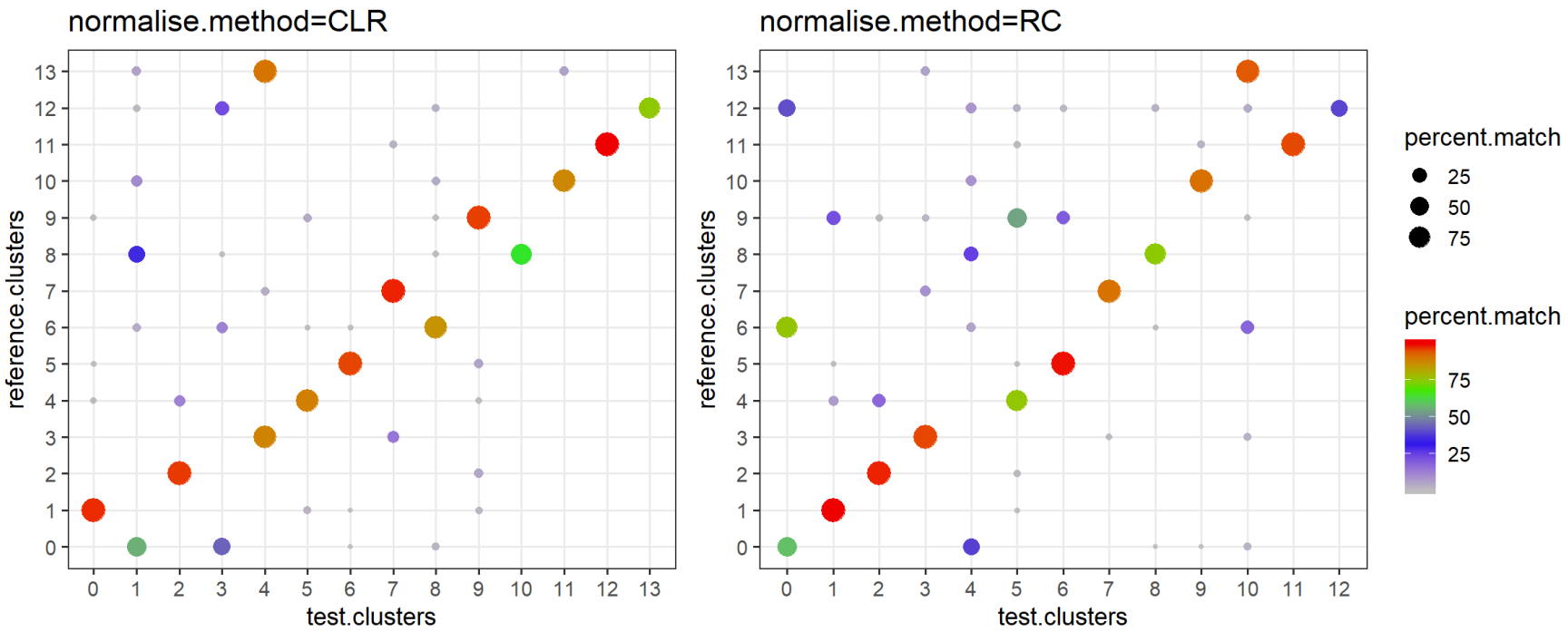
# Varying Parameters – Ribo Removal (100)



# Varying Parameters – Variable Features (2000)



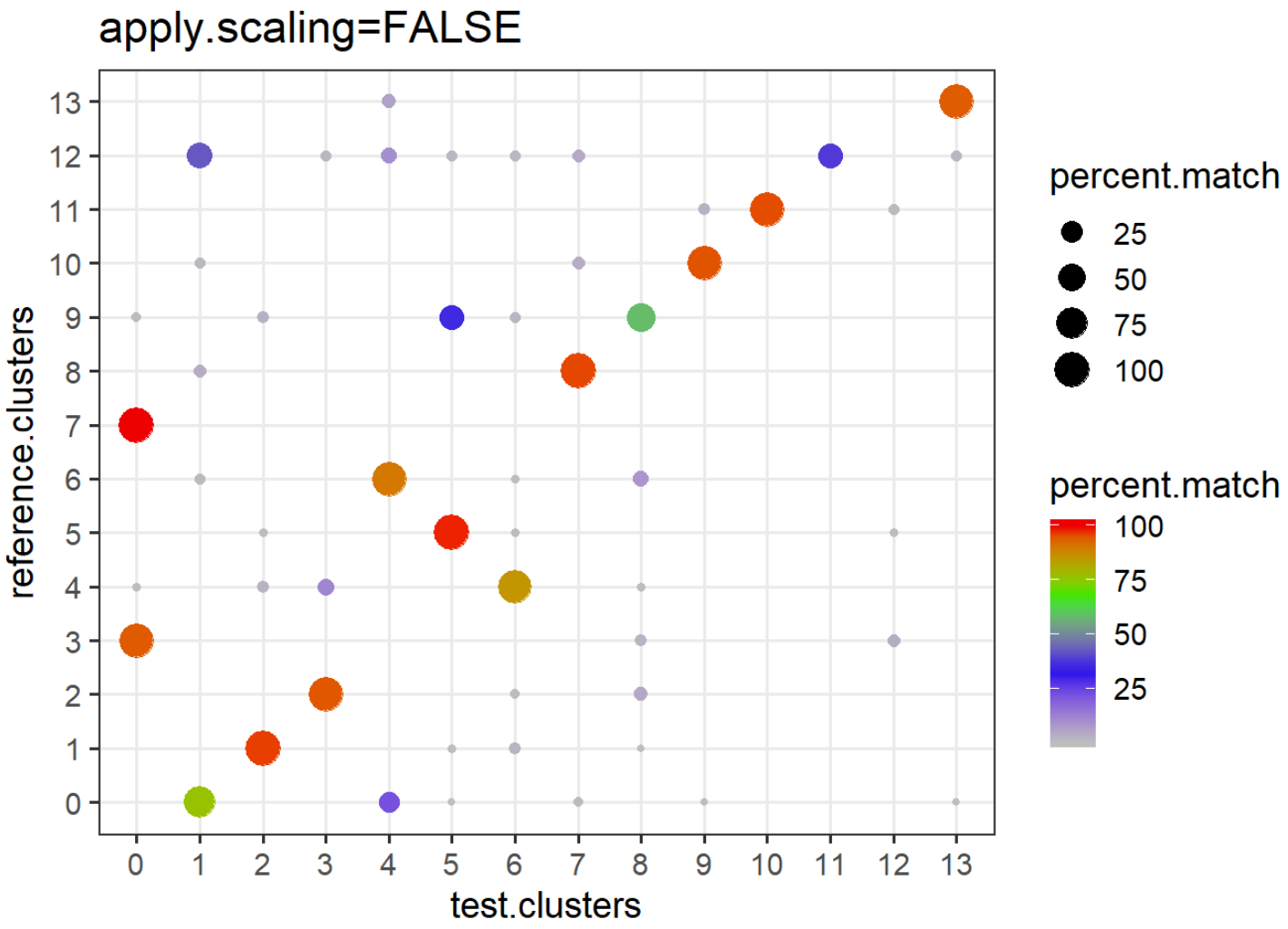
# Varying Parameters – Normalisation (Log Norm)



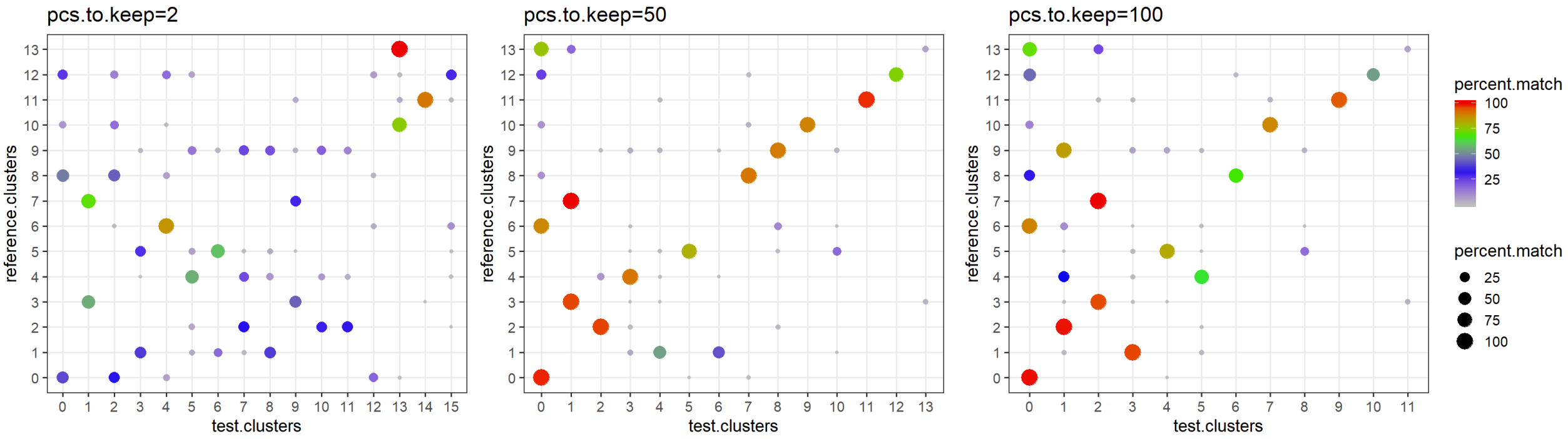

Centred Log Ratio
Similar to Size Factors

Relative counts
Linear Globally Normalised

# Varying Parameters – Apply Scaling (TRUE)



# Varying Parameters – PCs to keep (10)



# Varying Parameters – Cluster Resolution (0.5)