Analysing Single-Cell RNA-Seq with R

v2024-02

Simon Andrews

simon.andrews@babraham.ac.uk

Babraham Bioinformatics

Major scRNA Package Systems

https://satijalab.org/seurat/

scater

Single-Cell Analysis Toolkit for Gene Expression Data in R

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

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

https://scanpy.readthedocs.io/en/stable/

scanpy

What do they provide?

- Data Structure for modelling scRNA-Seq
  - Counts
  - Normalisations
  - Metadata
  - Clusters
- Convenience methods
  - 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

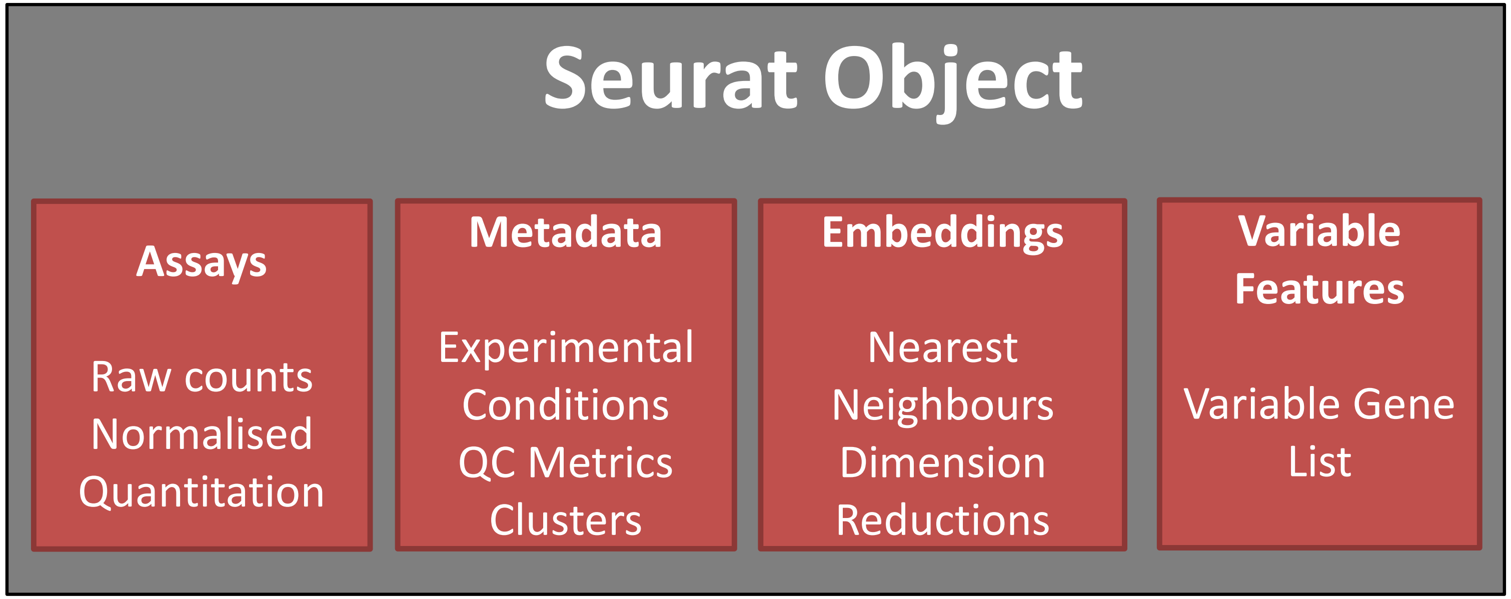
- Easy data model to work with
  - Documentation is good too

- 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)



Seurat Object

Assays

Raw counts
Normalised
Quantitation

Metadata

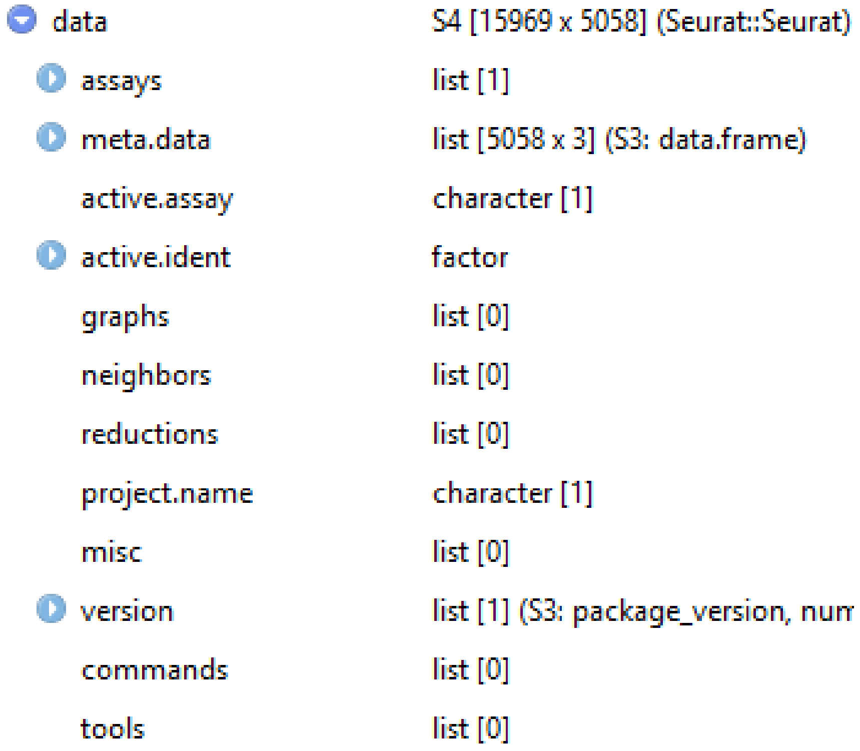
Experimental
Conditions
QC Metrics
Clusters

Embeddings

Nearest
Neighbours
Dimension
Reductions

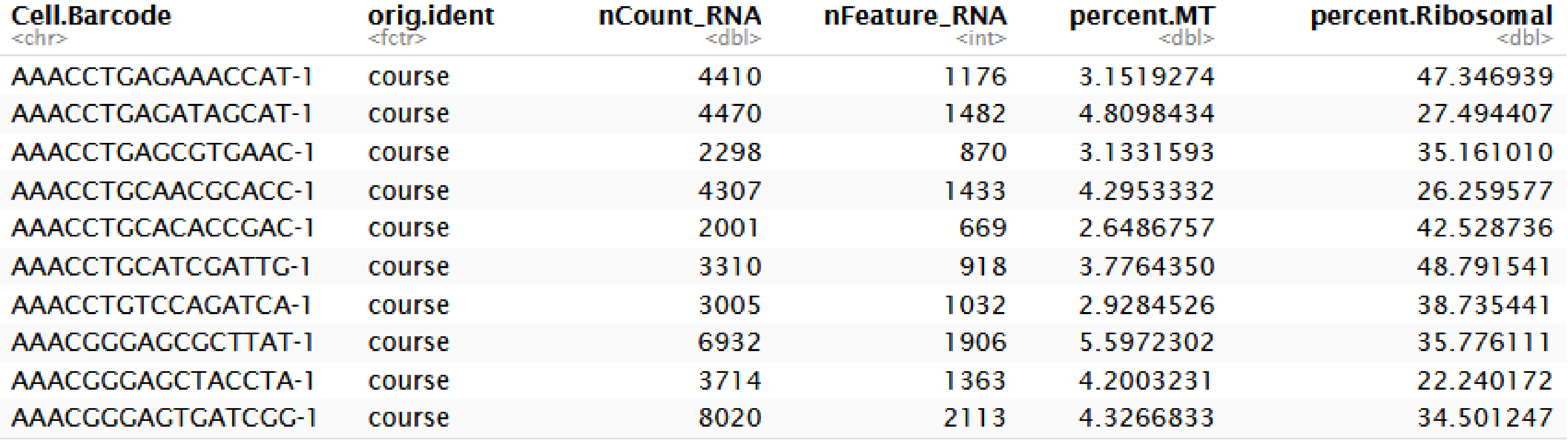
Variable Features

Variable Gene
List



```
data            S4 [15969 x 5058] (Seurat::Seurat)
  assays        list [1]
  meta.data     list [5058 x 3] (S3: data.frame)
  active.assay  character [1]
  active.ident  factor
  graphs        list [0]
  neighbors     list [0]
  reductions    list [0]
  project.name  character [1]
  misc          list [0]
  version       list [1] (S3: package_version, num
  commands      list [0]
  tools         list [0]
```

Seurat Metadata



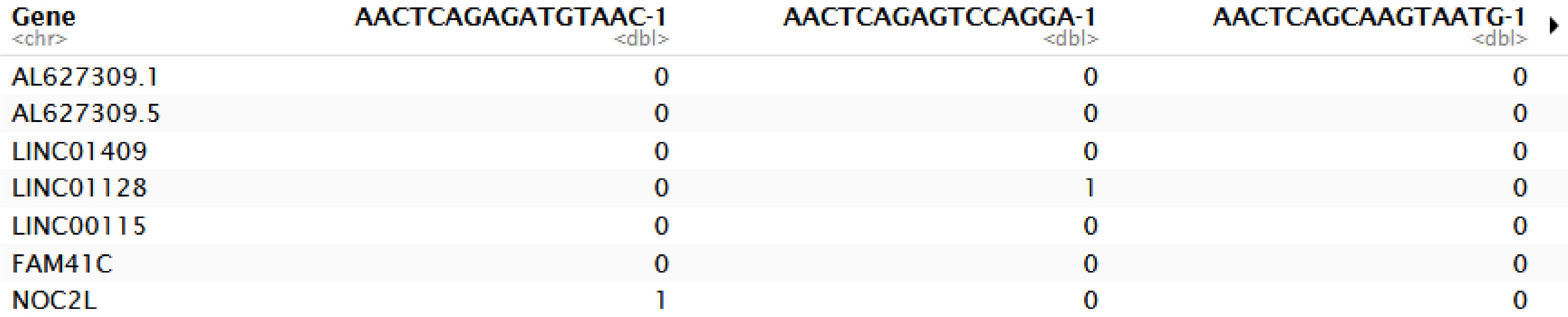
Cell.Barcode <chr>	orig.ident <fctr>	nCount_RNA <dbl>	nFeature_RNA <int>	percent.MT <dbl>	percent.Ribosomal <dbl>
AAACCTGAGAAACCAT-1	course	4410	1176	3.1519274	47.346939
AAACCTGAGATAGCAT-1	course	4470	1482	4.8098434	27.494407
AAACCTGAGCGTGAAC-1	course	2298	870	3.1331593	35.161010
AAACCTGCAACGCACC-1	course	4307	1433	4.2953332	26.259577
AAACCTGCACACCGAC-1	course	2001	669	2.6486757	42.528736
AAACCTGCATCGATTG-1	course	3310	918	3.7764350	48.791541
AAACCTGTCCAGATCA-1	course	3005	1032	2.9284526	38.735441
AAACGGGAGCGCTTAT-1	course	6932	1906	5.5972302	35.776111
AAACGGGAGCTACCTA-1	course	3714	1363	4.2003231	22.240172
AAACGGGAGTGATCGG-1	course	8020	2113	4.3266833	34.501247

- QC
- Conditions
- Clusters

```
data[[]]
data$nCount_RNA
new_data -> data$new_metric
```

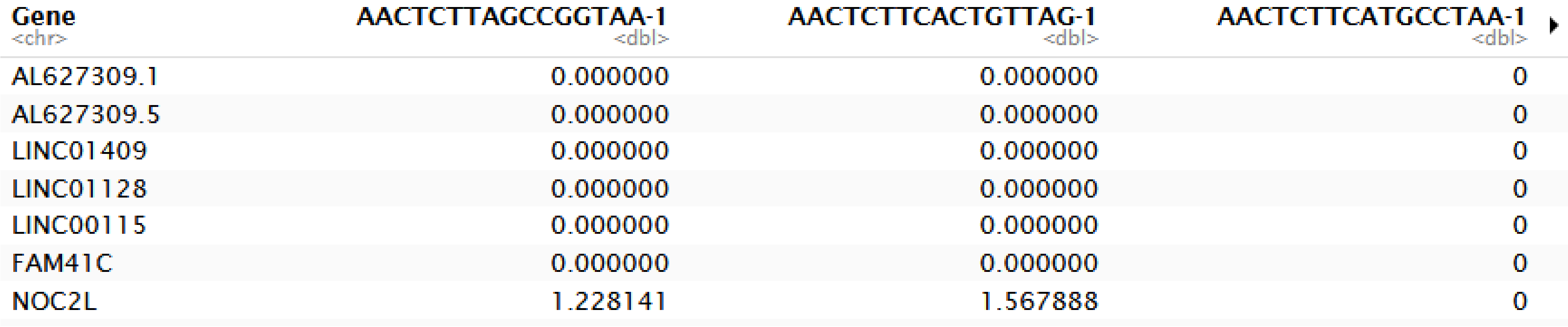
Seurat Quantitative Data

```
> GetAssayData(data, layer="counts")
```



Gene	AACTCAGAGATGTAAC-1	AACTCAGAGTCCAGGA-1	AACTCAGCAAGTAATG-1	▸
<chr>	<dbl>	<dbl>	<dbl>	
AL627309.1	0	0	0	
AL627309.5	0	0	0	
LINC01409	0	0	0	
LINC01128	0	1	0	
LINC00115	0	0	0	
FAM41C	0	0	0	
NOC2L	1	0	0	

```
> GetAssayData(data, layer="data")
```



Gene	AACTCTTAGCCGGTAA-1	AACTCTTCACTGTTAG-1	AACTCTTCATGCCTAA-1	▸
<chr>	<dbl>	<dbl>	<dbl>	
AL627309.1	0.000000	0.000000	0	
AL627309.5	0.000000	0.000000	0	
LINC01409	0.000000	0.000000	0	
LINC01128	0.000000	0.000000	0	
LINC00115	0.000000	0.000000	0	
FAM41C	0.000000	0.000000	0	
NOC2L	1.228141	1.567888	0	

Seurat Dimensionality Reductions

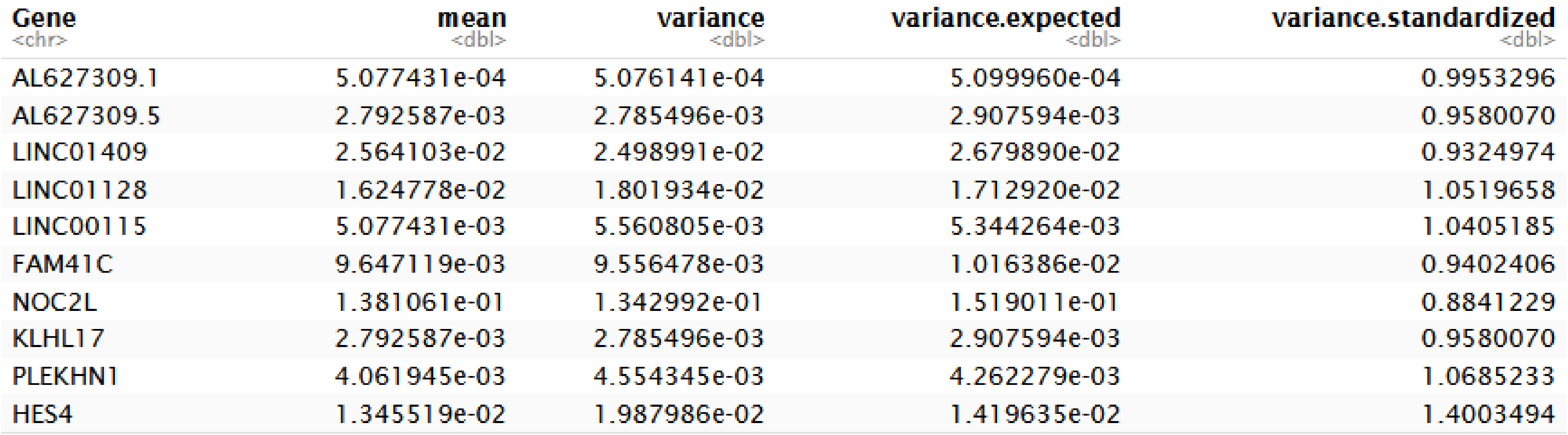
```
> Embeddings(data,reduction = "pca")
                 PC_1       PC_2         PC_3
GAGAAACCAT-1 -4.135099 -8.7585629  0.004920869
GAGATAGCAT-1 10.279728  0.8451562  0.287589575
GAGCGTGAAC-1 -6.002598  4.9504875 -3.022266598
GCAACGCACC-1 10.610838  0.4030928  0.165408128
GCATCGATTG-1 -5.581052 -0.4542359  5.166186308
```

```
> Loadings(data, reduction="pca")
              PC_1        PC_2         PC_3
IGLC3  -0.01326925 -0.07255036 -0.017863727
IGLC1  -0.01039493 -0.05451957 -0.008700478
IGLC2  -0.01569606 -0.08276587 -0.021715297
IGKC   -0.02226017 -0.12668741 -0.038651318
S100A9  0.11325462  0.01836318  0.013403732
```

```
> Embeddings(data,reduction = "tsne")
                      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
```

Variable Gene Information

```
> HVFInfo(data)
```



Gene	mean	variance	variance.expected	variance.standardized
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
AL627309.1	5.077431e-04	5.076141e-04	5.099960e-04	0.9953296
AL627309.5	2.792587e-03	2.785496e-03	2.907594e-03	0.9580070
LINC01409	2.564103e-02	2.498991e-02	2.679890e-02	0.9324974
LINC01128	1.624778e-02	1.801934e-02	1.712920e-02	1.0519658
LINC00115	5.077431e-03	5.560805e-03	5.344264e-03	1.0405185
FAM41C	9.647119e-03	9.556478e-03	1.016386e-02	0.9402406
NOC2L	1.381061e-01	1.342992e-01	1.519011e-01	0.8841229
KLHL17	2.792587e-03	2.785496e-03	2.907594e-03	0.9580070
PLEKHN1	4.061945e-03	4.554345e-03	4.262279e-03	1.0685233
HES4	1.345519e-02	1.987986e-02	1.419635e-02	1.4003494

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`

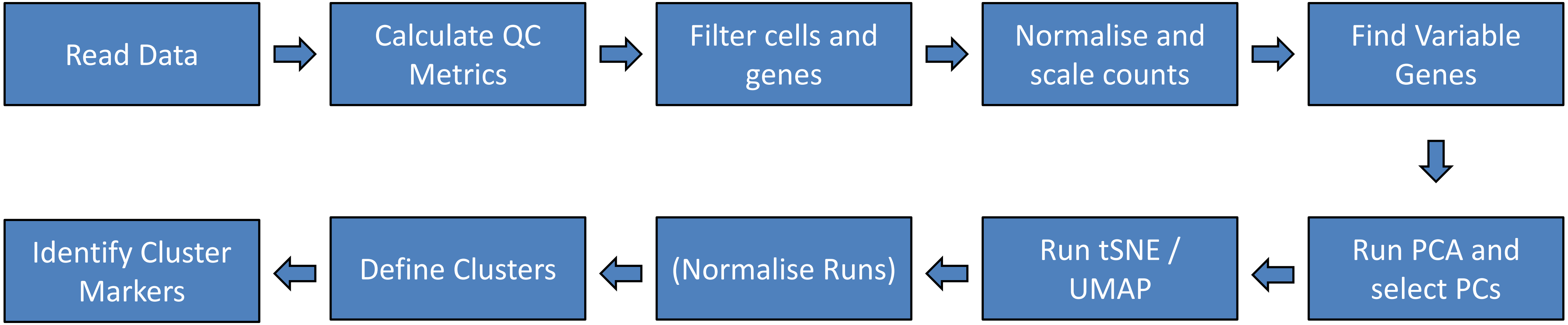
- Statistics
  - Select Variable Genes `FindVariableFeatures`
  - Build nearest neighbour graph `FindNeighbors`
  - Build graph based cell clusters `FindClusters`
  - Find genes to classify clusters (multiple tests) `FindMarkers`

*Requires installing the `hdf5r` package

Example 10X Seurat Workflow

Babraham Bioinformatics

Example Seurat Workflow



Read Data

Calculate QC Metrics

Filter cells and genes

Normalise and scale counts

Find Variable Genes

Run PCA and select PCs

Run tSNE / UMAP

(Normalise Runs)

Define Clusters

Identify Cluster Markers

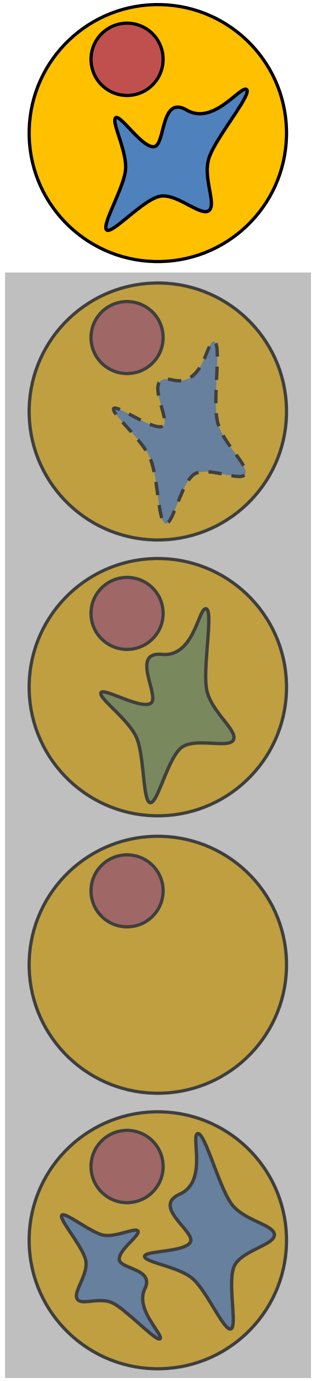
Reading Data

```
Read10X_h5("filtered_feature_bc_matrix.h5") -> data

CreateSeuratObject(
      counts=data,
      project="course",
) -> data
```

```
> data
An object of class Seurat
17136 features across 3939 samples within 1 assay
Active assay: RNA (17136 features, 500 variable
features)
 3 layers present: counts, data, scale.data
 2 dimensional reductions calculated: pca, tsne
```

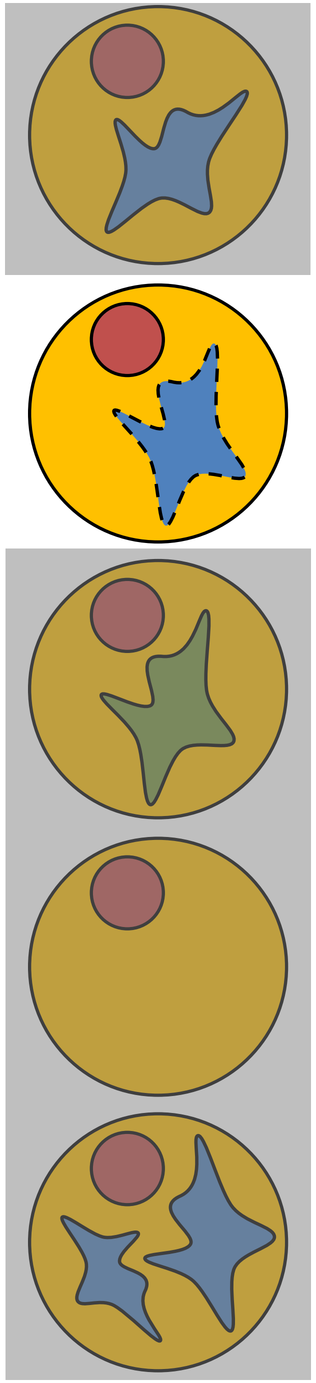
QC – What problems are likely?

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



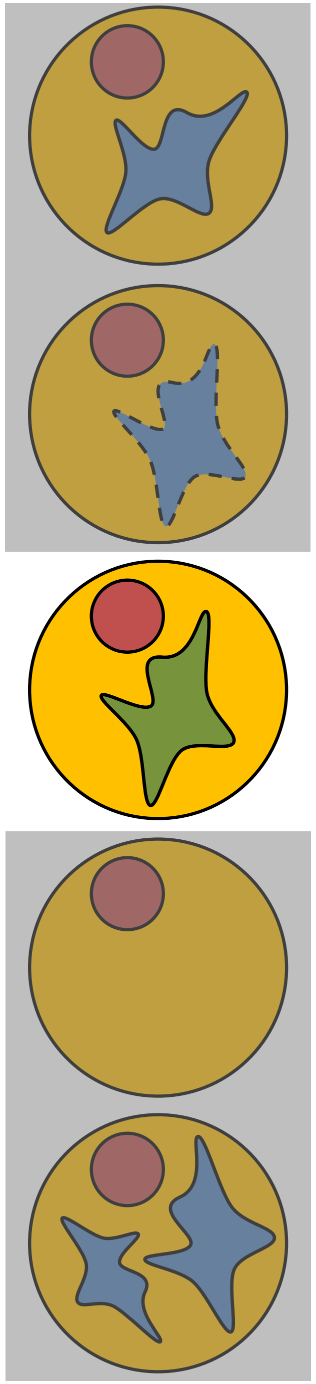
Lysed Cells

- 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

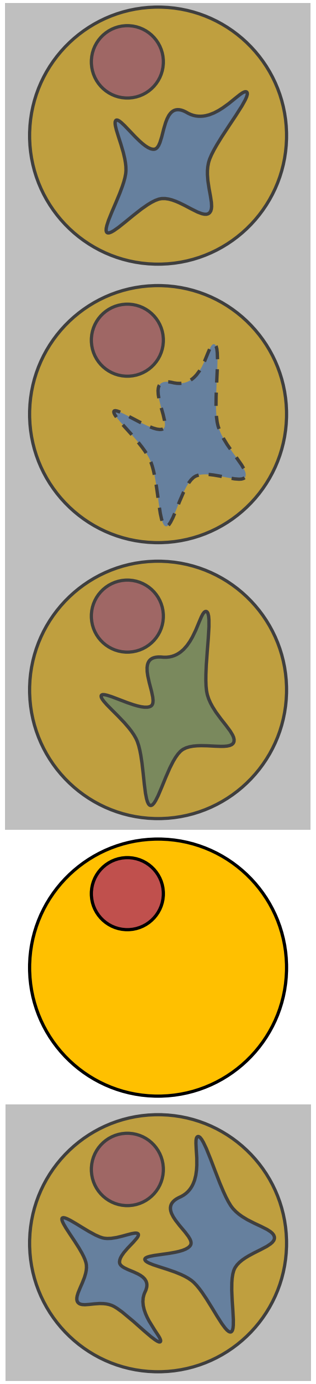
- Cells undergoing apoptosis have very different transcriptomes
  - – Lower total RNA production
  - – Huge upregulation of mitochondrial transcription
  - – Upregulation of caspases



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

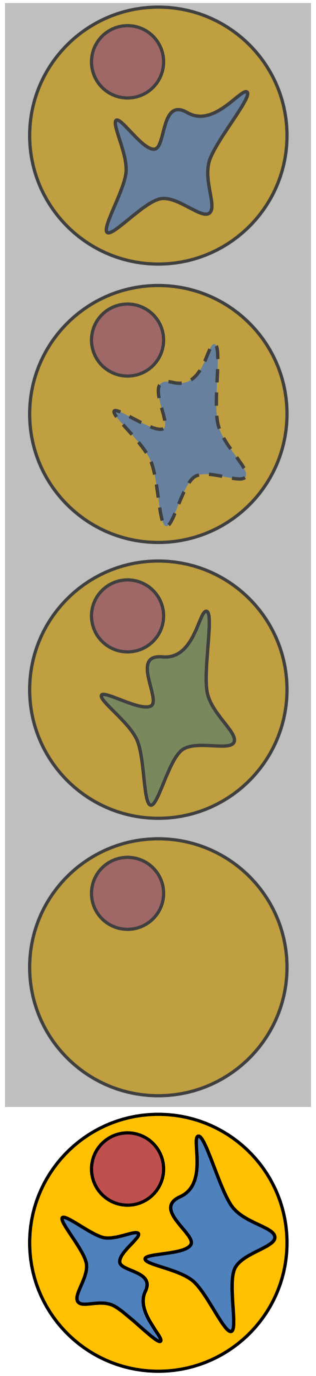
- Total amount of signal will be greatly reduced
- Will often look similar to each other (will cluster together)



Double occupied GEMs

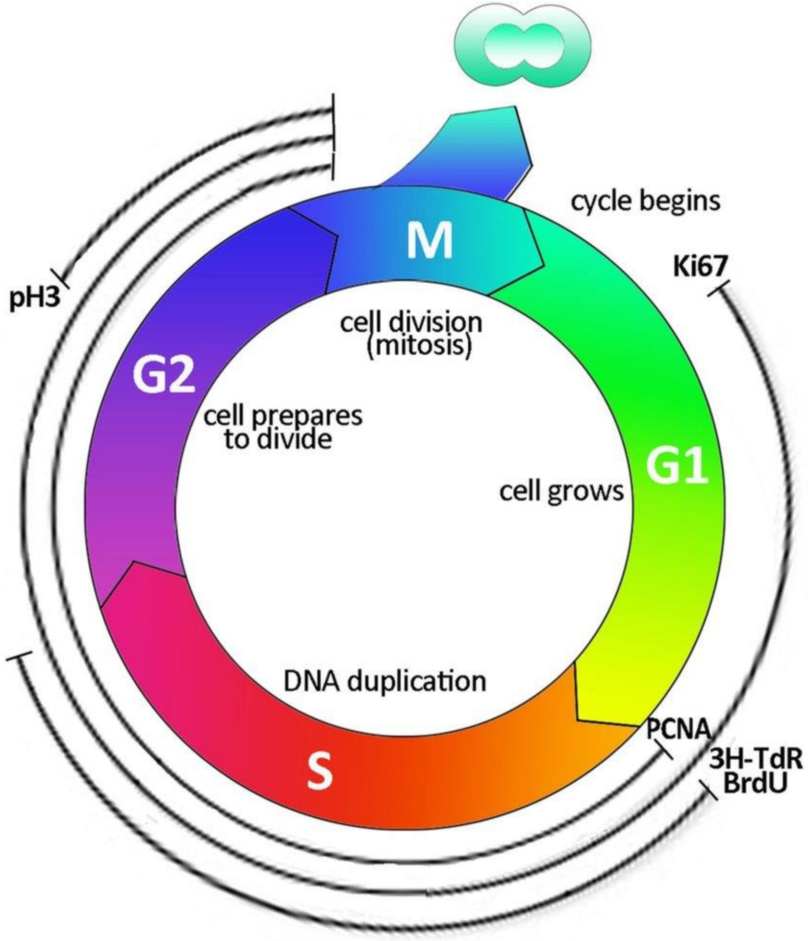
- Will get a mixed signal from two different cells

- 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

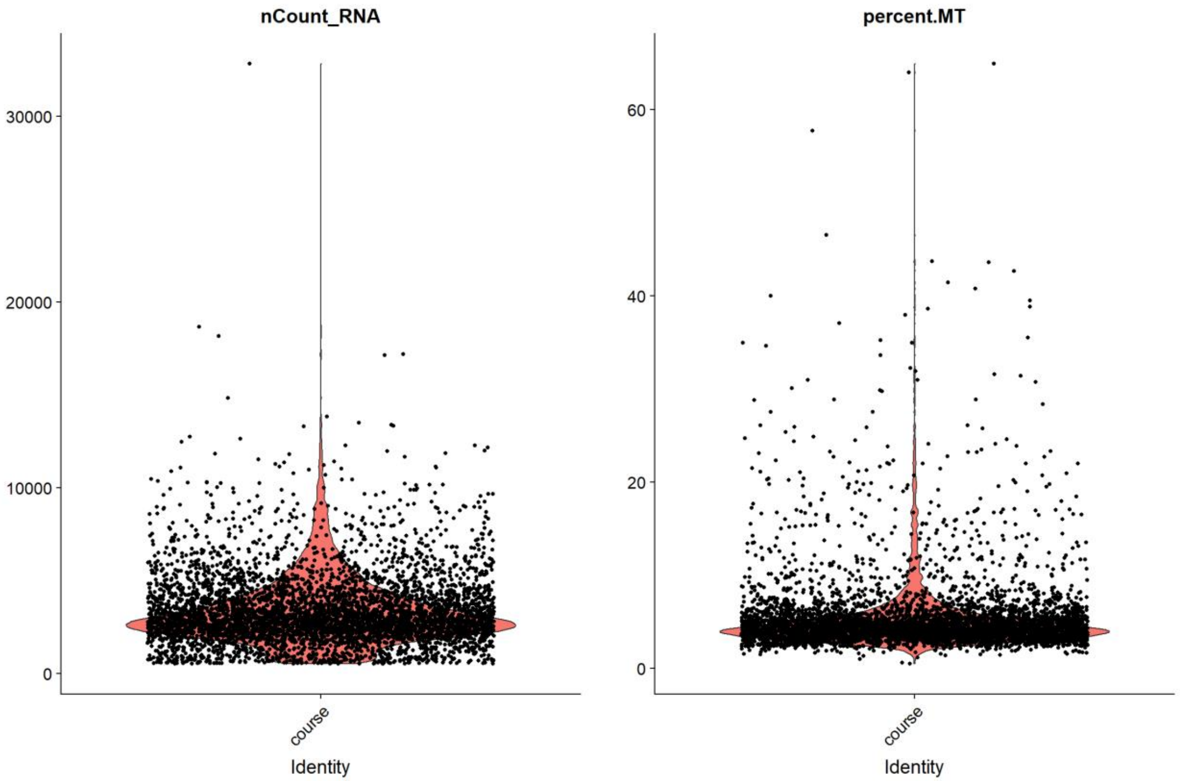


Lombardi, et al. (2015). 10.1093/ndt/gfv262.

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



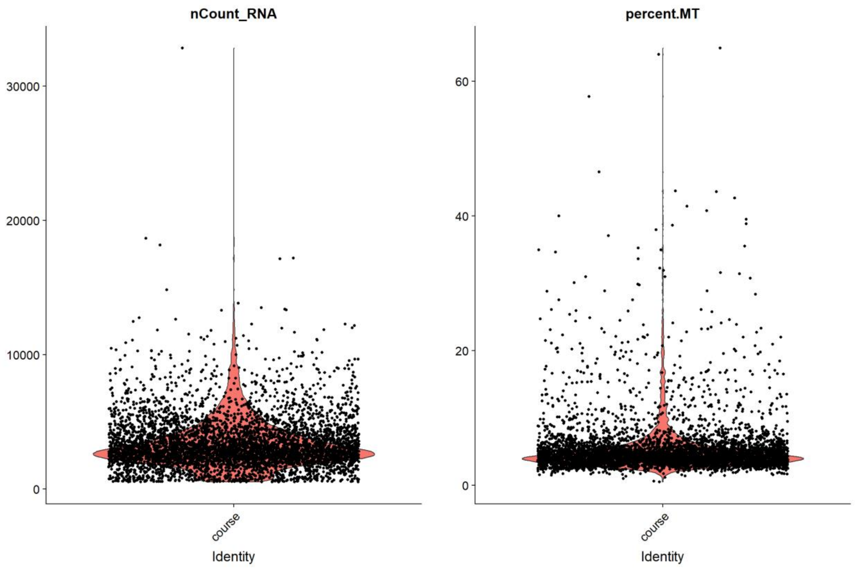
Custom QC Metrics

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

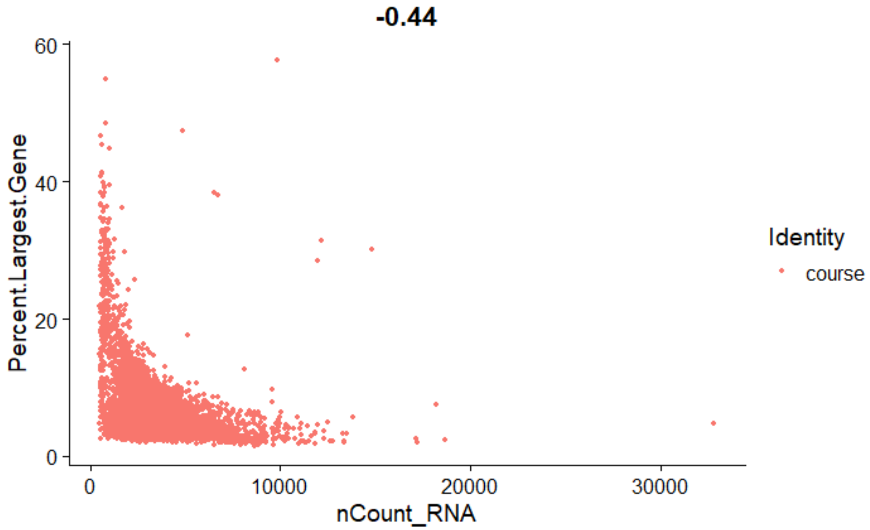
```
apply(
  GetAssayData(data, layer="counts"),
  2,
  function(x)(100*max(x))/sum(x)
) -> data$Percent.Largest.Gene
```

QC and Cell Filtering

```
VlnPlot(
        data,
        features=c("nCount_RNA","percent.MT"),
        log=TRUE
)
```



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



Applying Filters

```
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

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

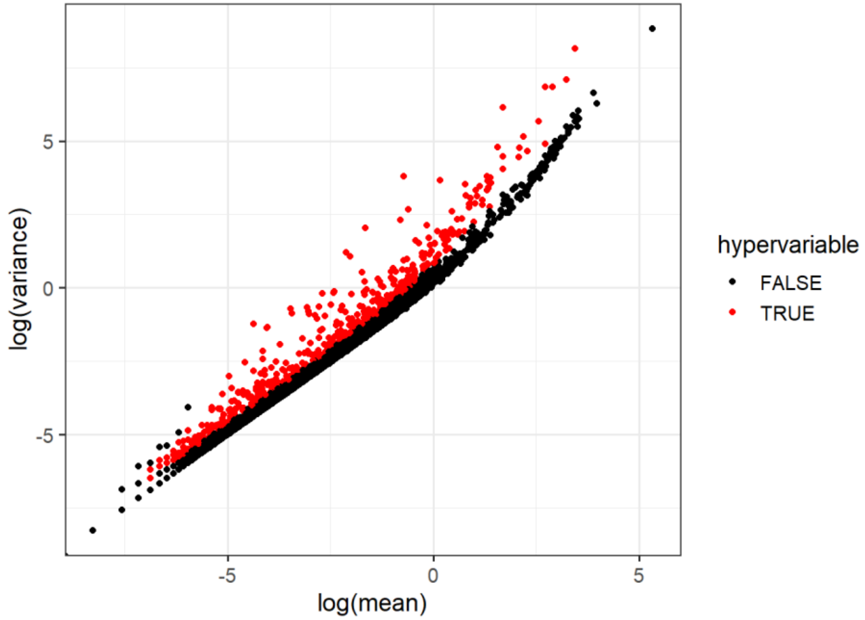
Count Normalisation and Scaling

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

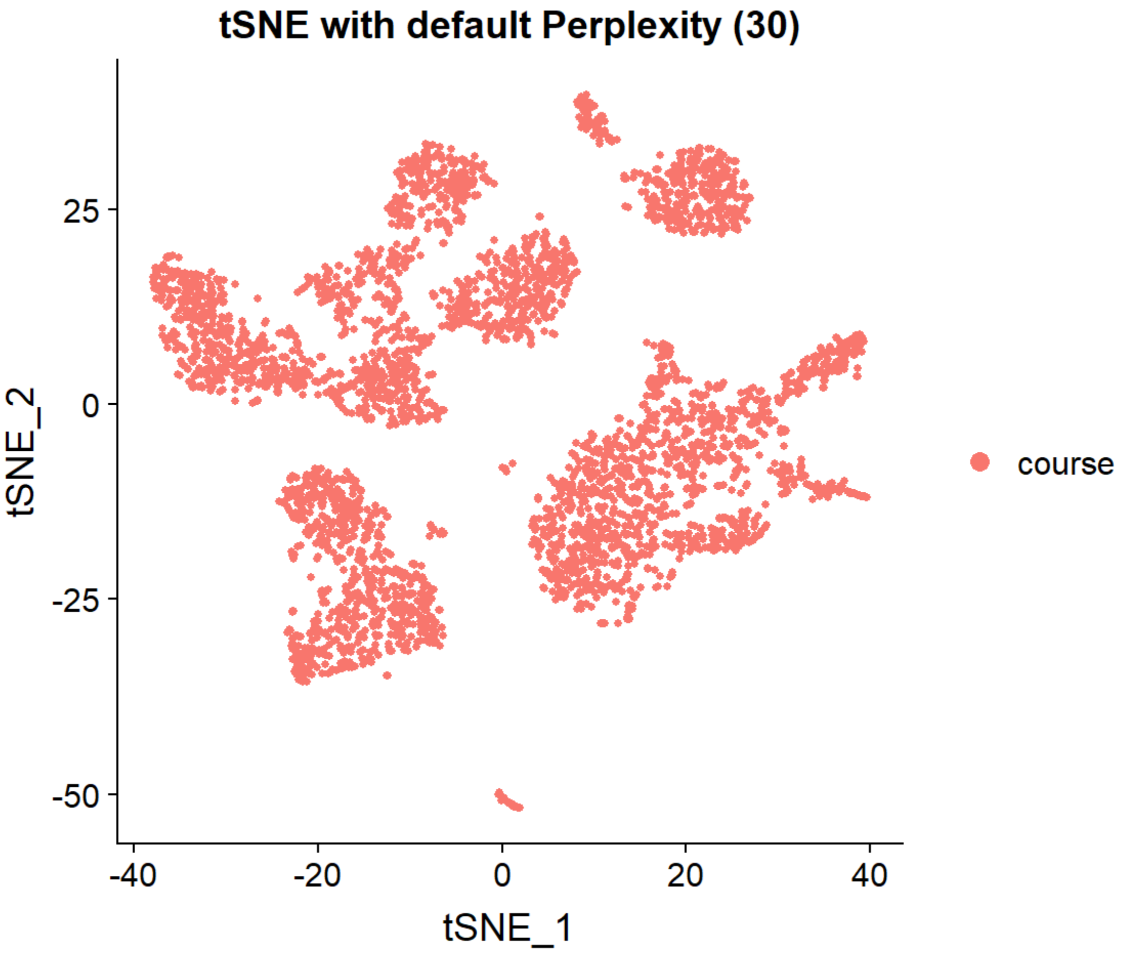


Dimensionality Reduction

- Start with PCA on the normalised, filtered (both cells and genes), scaled data

- Scree / Elbow plot to decide how many PCs are informative

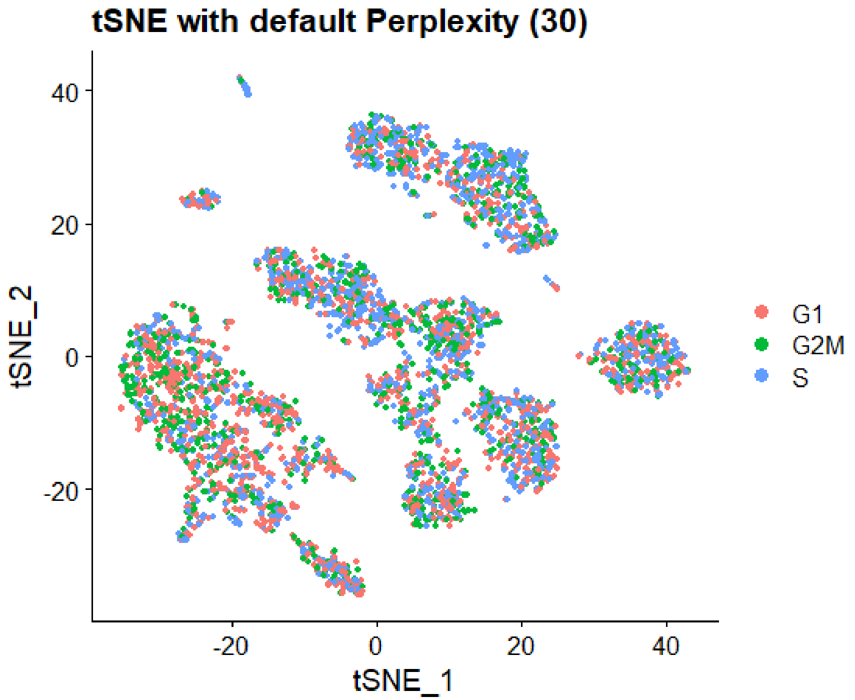
- Pass only the interesting PCs to subsequent tSNE or UMAP reduction to get down to 2 dimensions

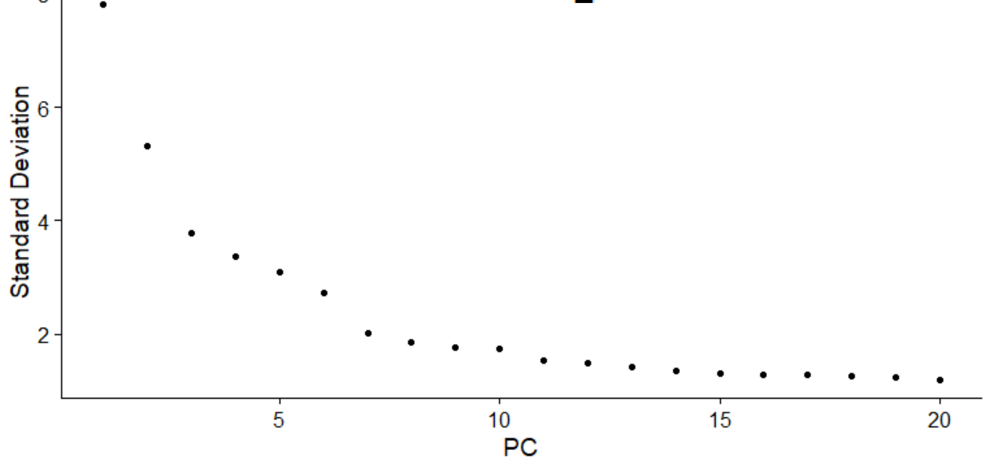


Dimensionality Reduction

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

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


tSNE with default Perplexity (30)

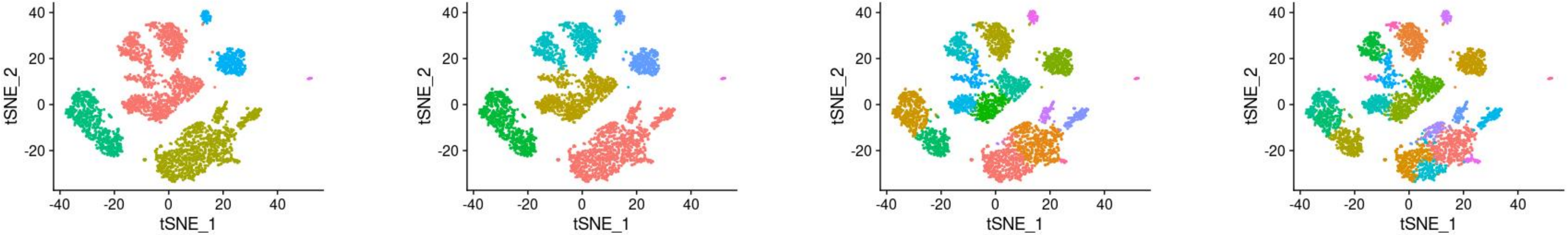


Defining clusters

- Construct nearest neighbour graph
  - Constructed from PCA
  - Same dimensions as tSNE/UMAP

- Find clusters
  - All cells are classified
  - Graph Based (Louvain) Clustering
  - Resolution (0.01 - 5) defines granularity

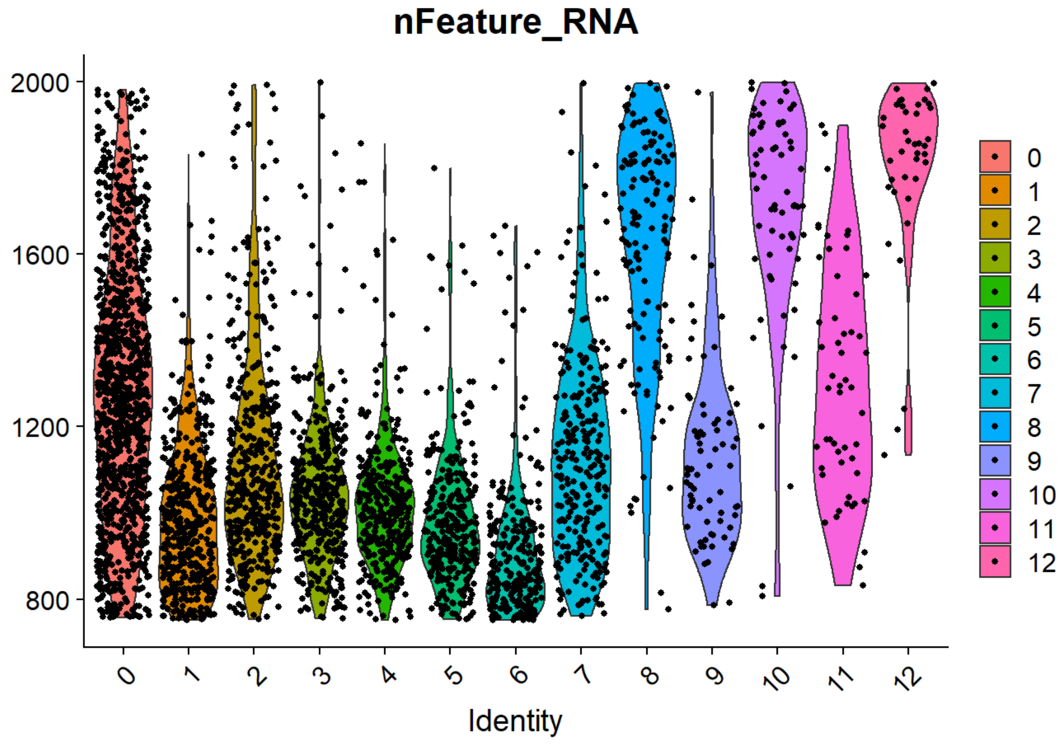
```
FindNeighbors(
     data,
     dims=1:15
) -> data

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



Comparing Properties of Clusters

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

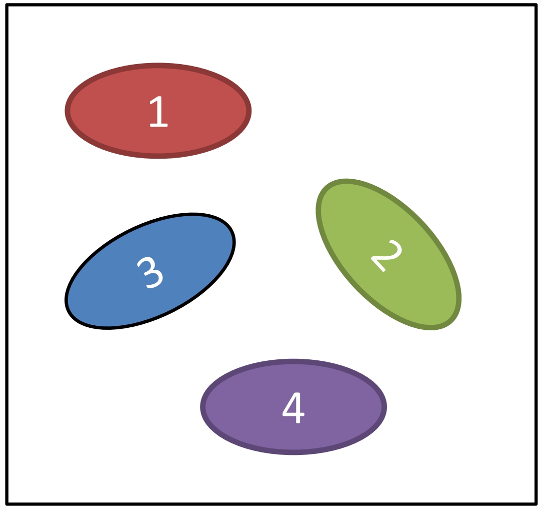
- We want to know that clusters are occurring because of biological changes, not technical differences
- We plot QC metrics for clusters
  - Read/Gene counts
  - Mitochondrion
  - MALAT1
- Can remove suspect clusters



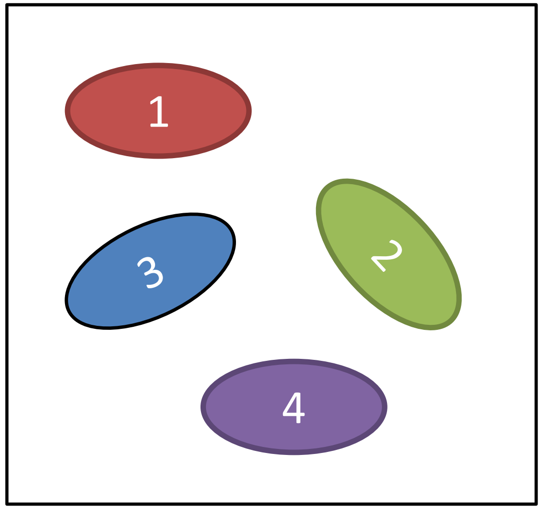
```
subset(data, !seurat.clusters %in% c(8,10,12)) -> data
```

Statistical Analysis

Sample A



Sample B



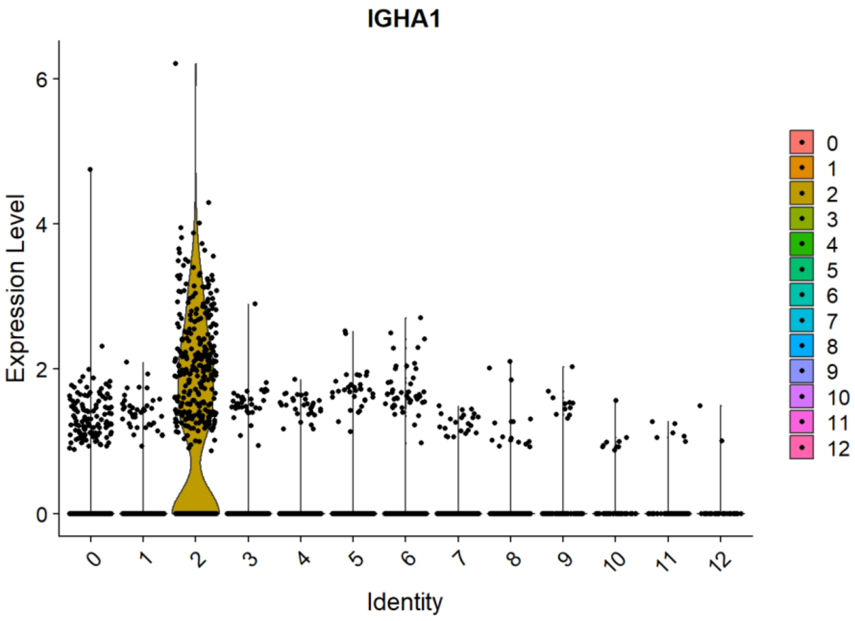
- Cluster 1 vs Clusters [2,3,4]
- Cluster 1 vs Cluster 3

- Cluster A1 vs Cluster B1

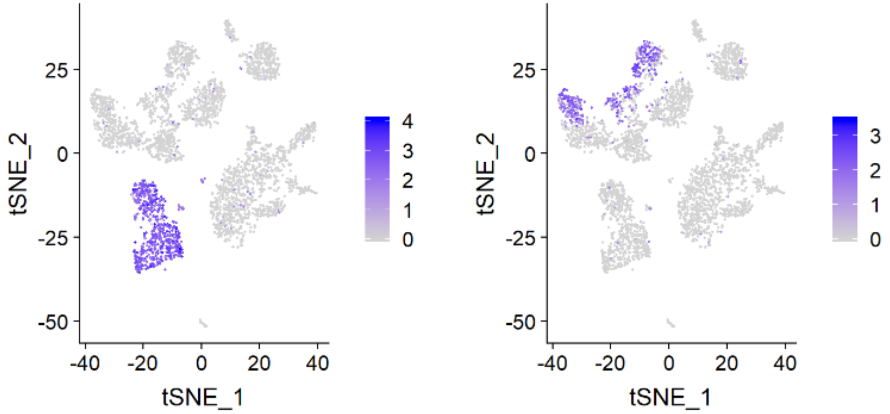
Statistical analysis of differences between clusters

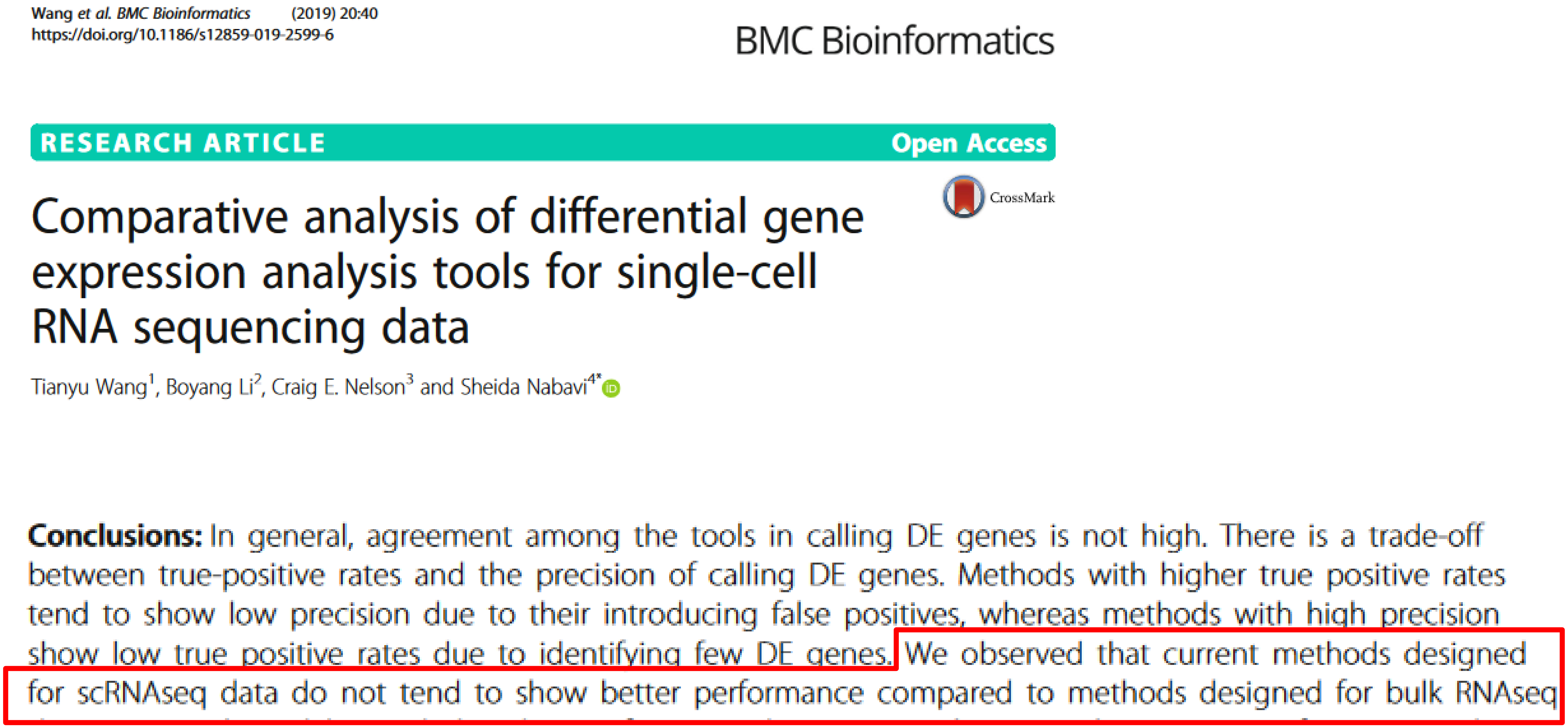
- Non-parametric
  - Wilcox rank sum test
- Parametric
  - T-test
  - Negative binomial (eg DESeq)

- 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

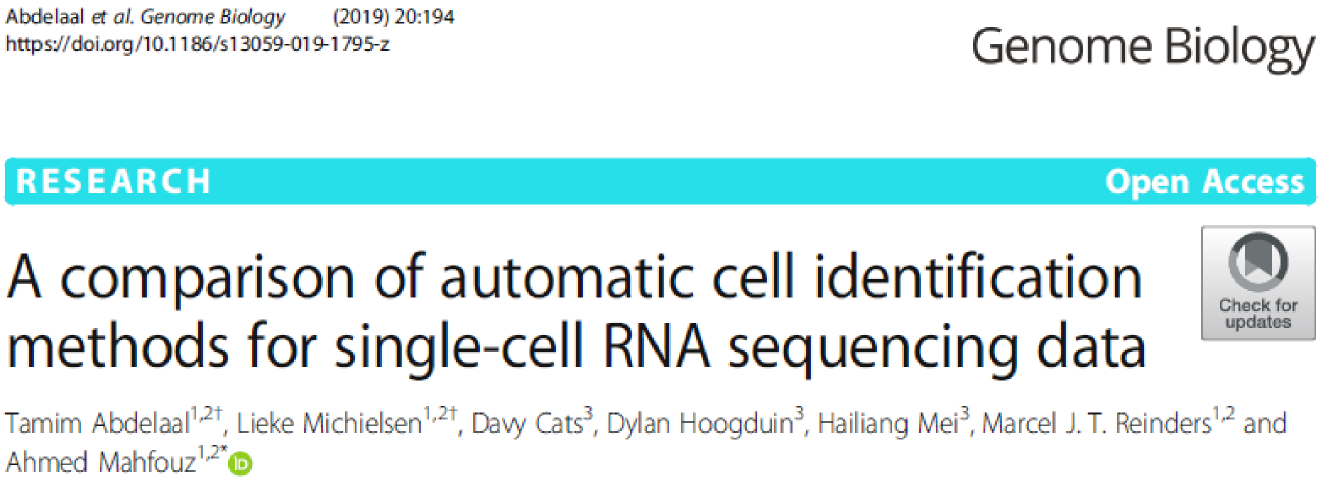
Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data

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

Check for updates

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

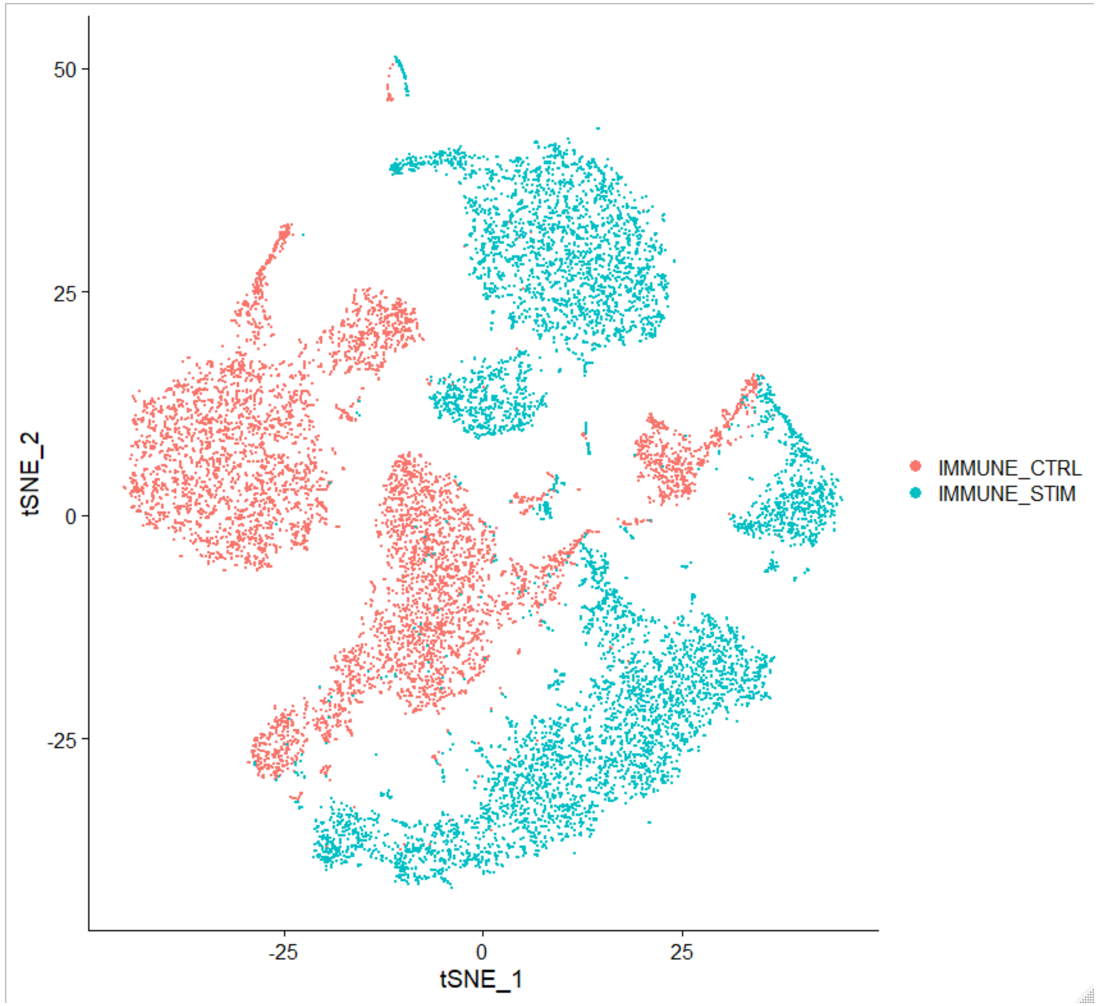
- Raw comparisons can therefore miss changes between what are actually matched subgroups

Raw merged runs

- Two PBMC populations run at different times

- tSNE spread coloured by library

- Little to no overlap between cell populations



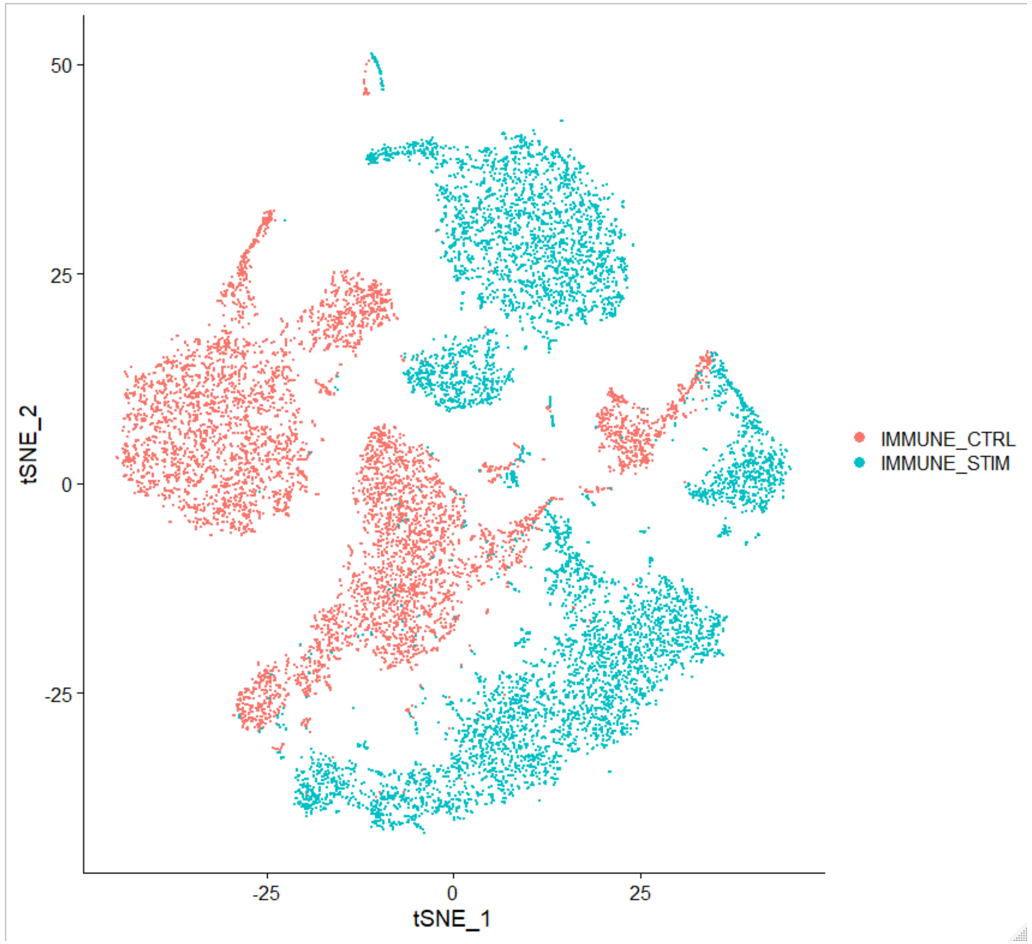
Anchoring Runs

- Method to try to re-align different runs

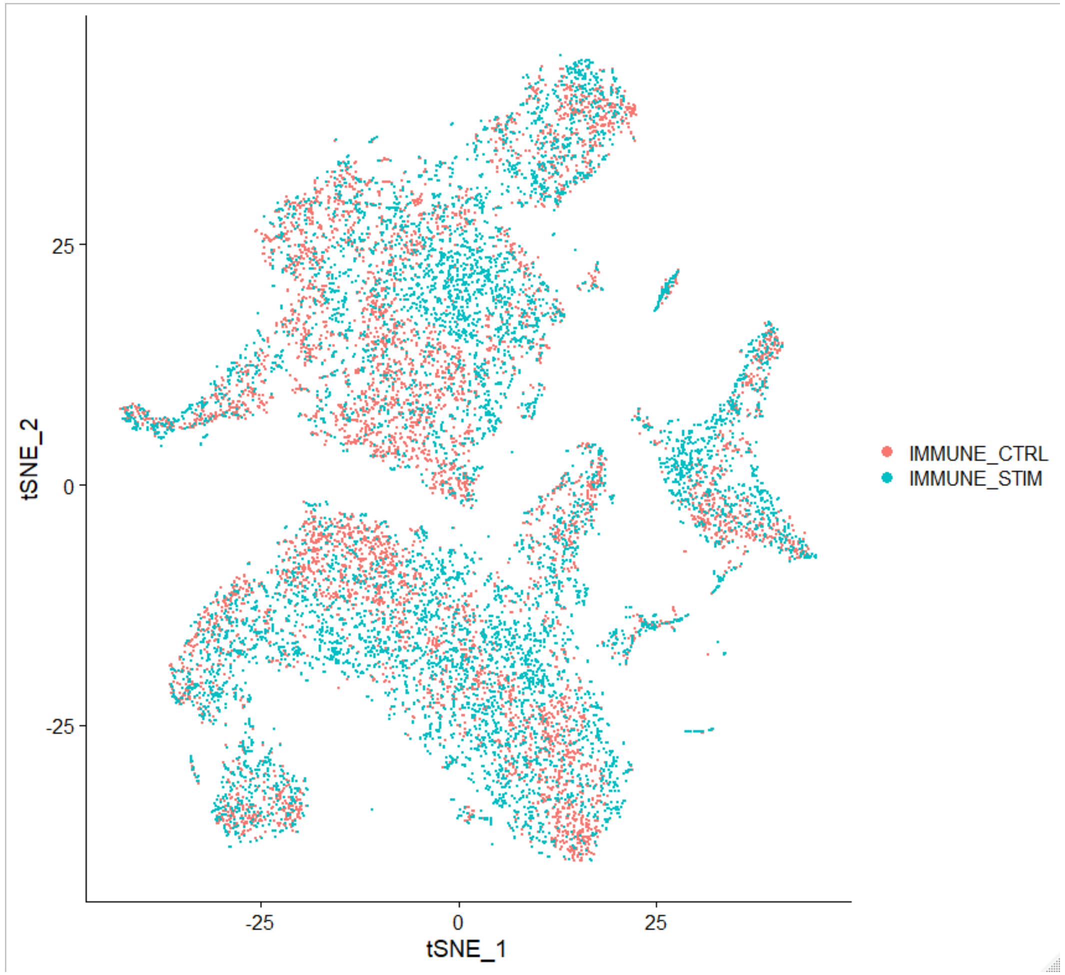
- Uses mutual nearest neighbour searches between runs to pair up cells

- Uses pairs to align the dimension reduction plots

Anchoring Runs

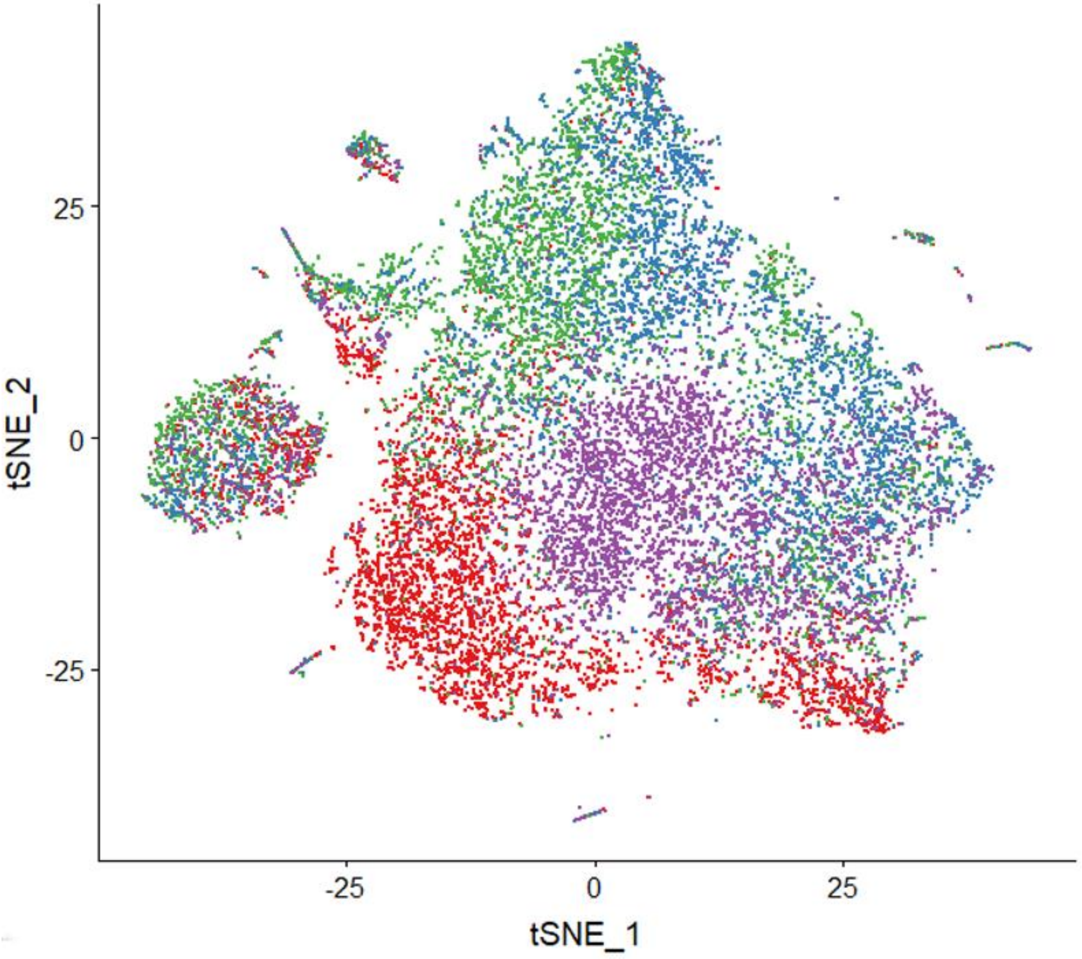


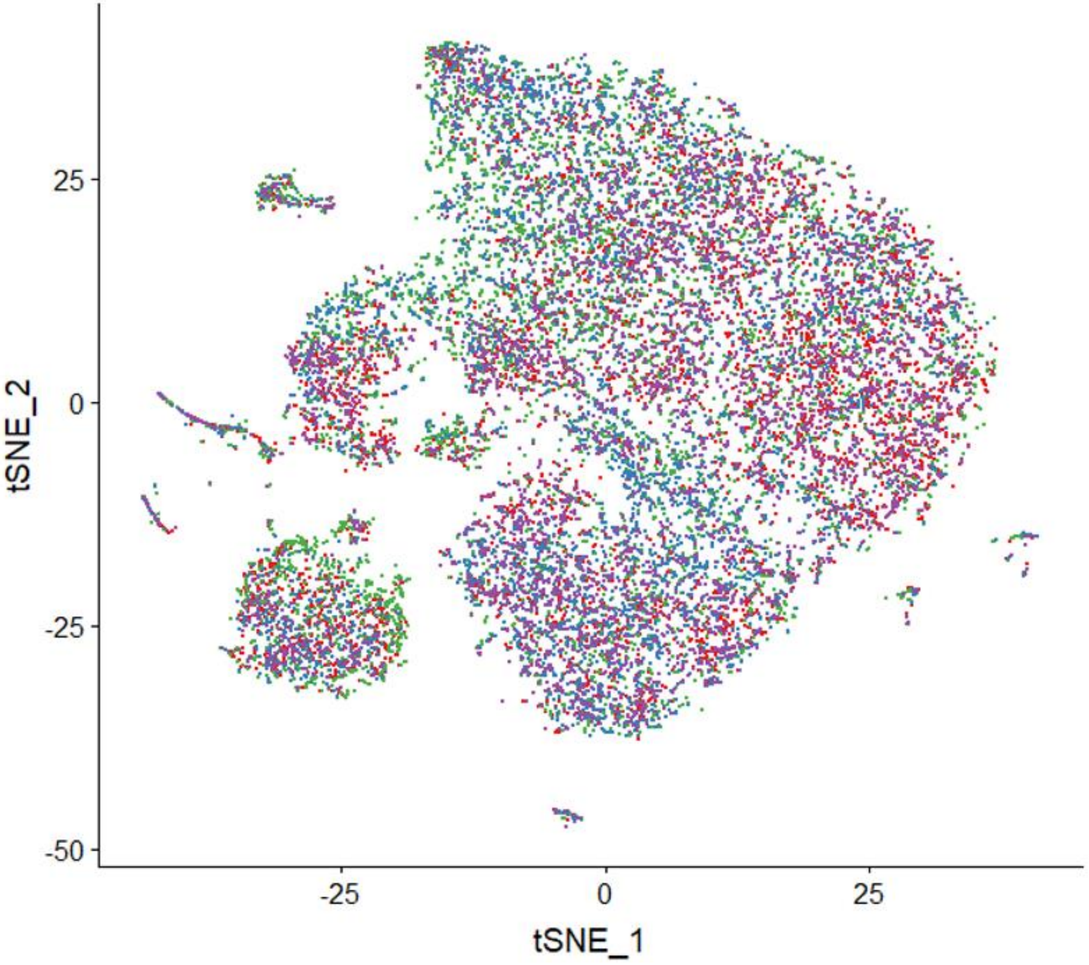
Raw



Anchored

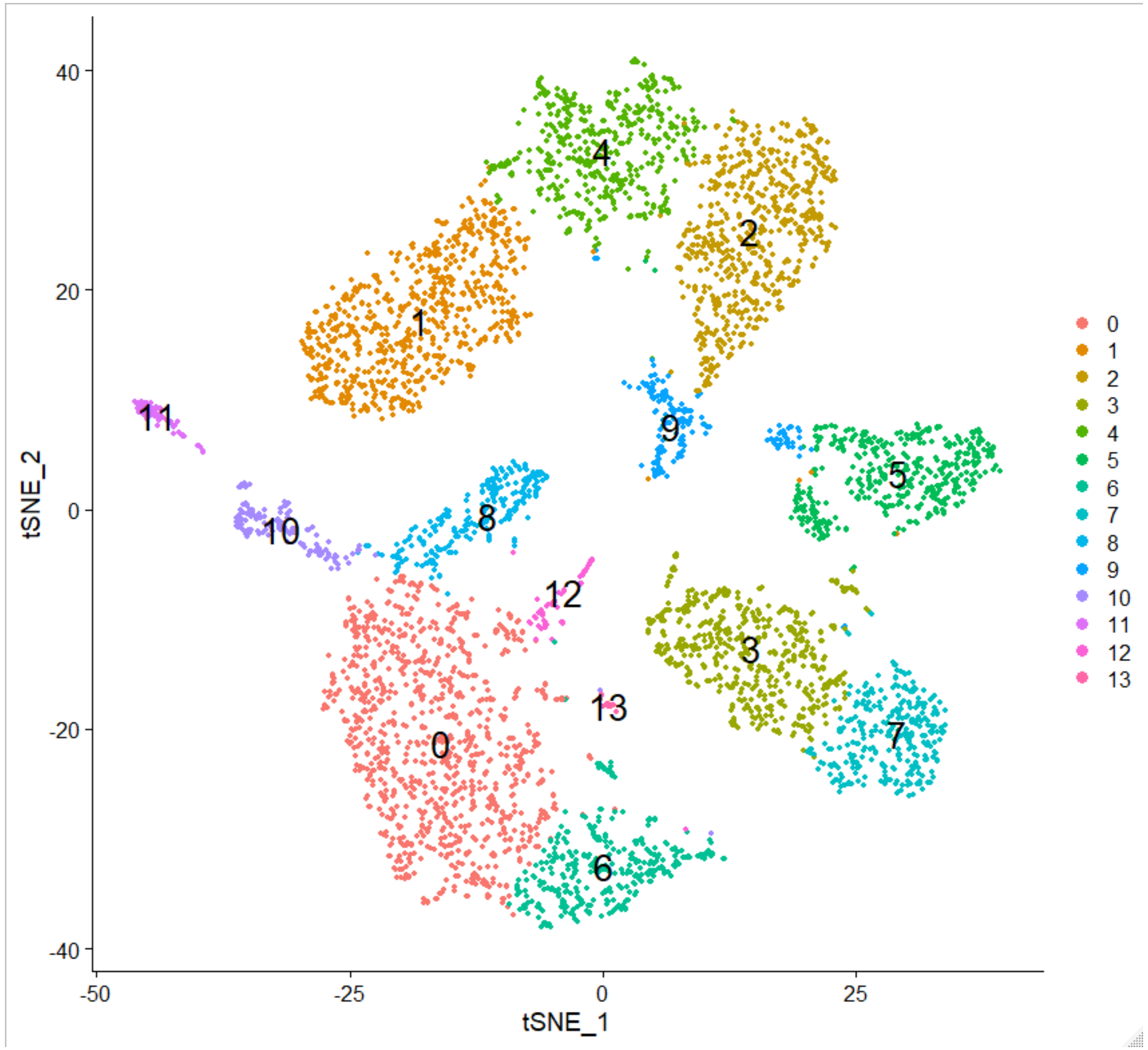
Over-Anchoring





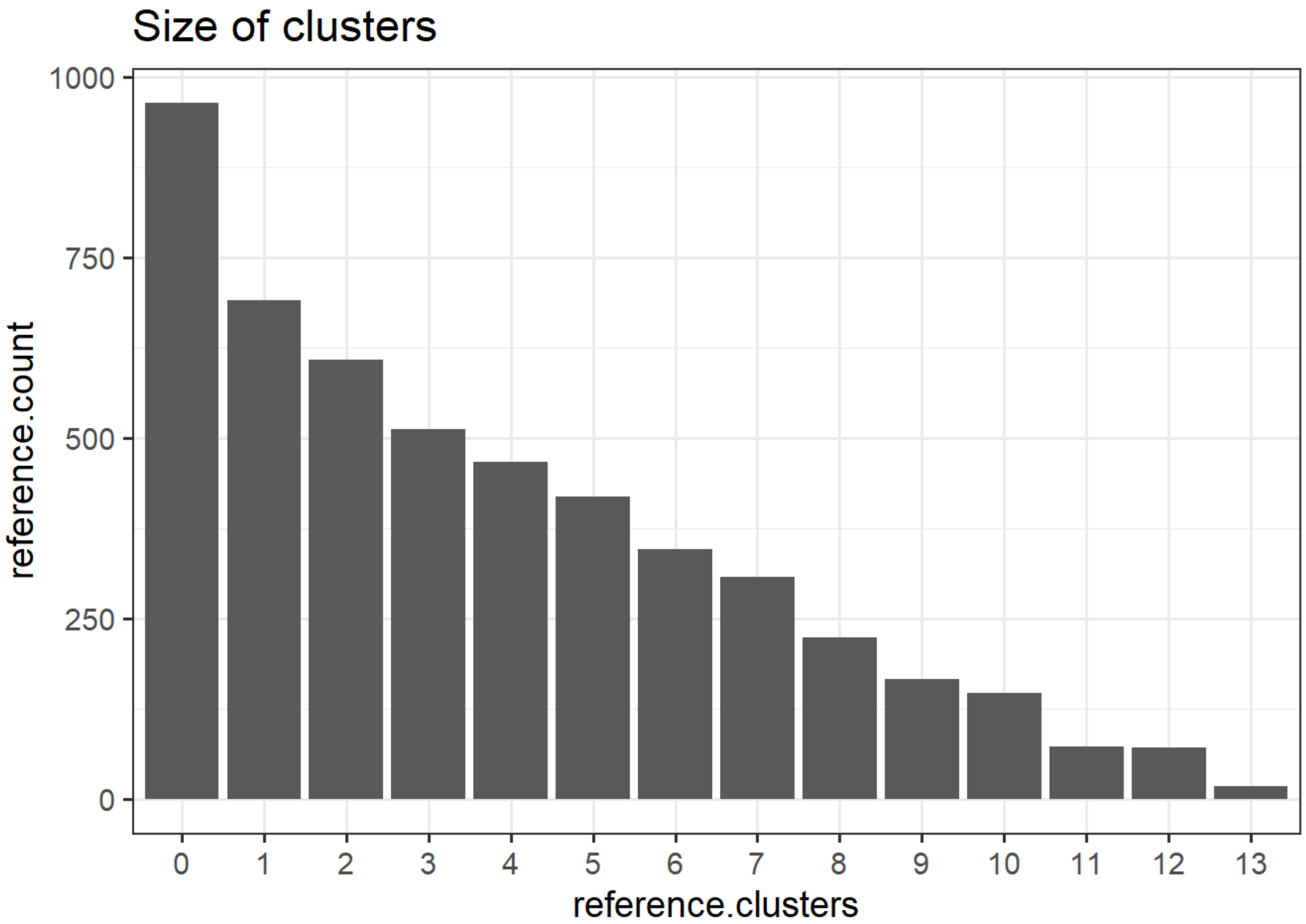
Exercise – Using Seurat to analyse 10X data

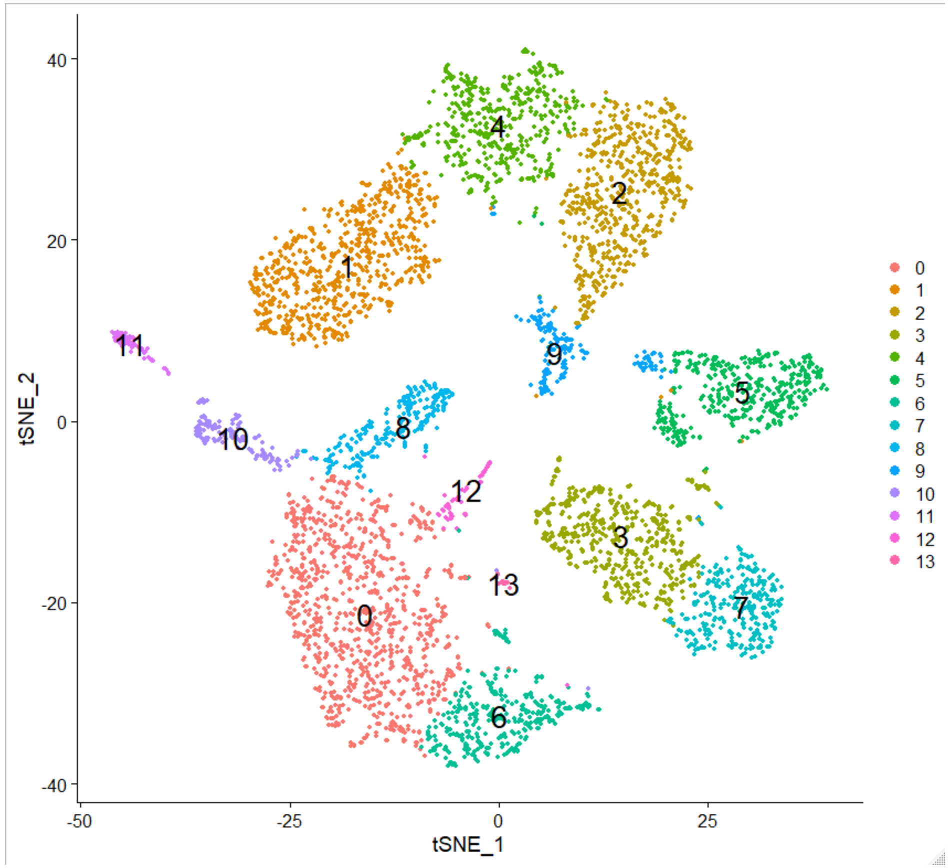
Which factors matter?



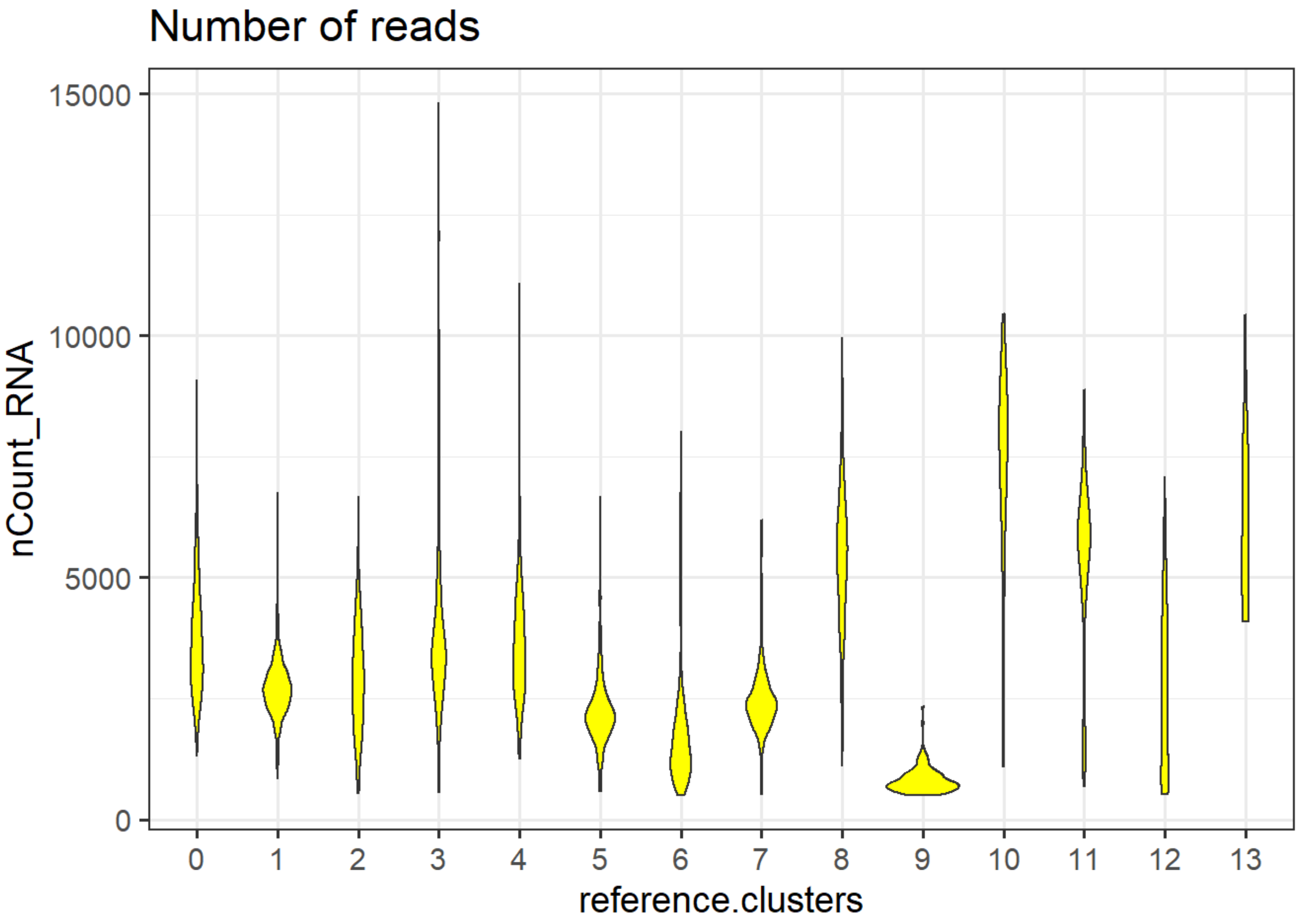
```
run.seurat.analysis <- function (
  data,
  number.of.genes.min = 200,
  number.of.genes.max = 2500,
  percent.mito = 100,
  percent.ribo = 100,
  normalise.method = "LogNormalize",
  number.variable.features = 2000,
  apply.scaling = TRUE,
  pcs.to.keep = 10,
  cluster.resolution = 0.5,
  remove.mito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

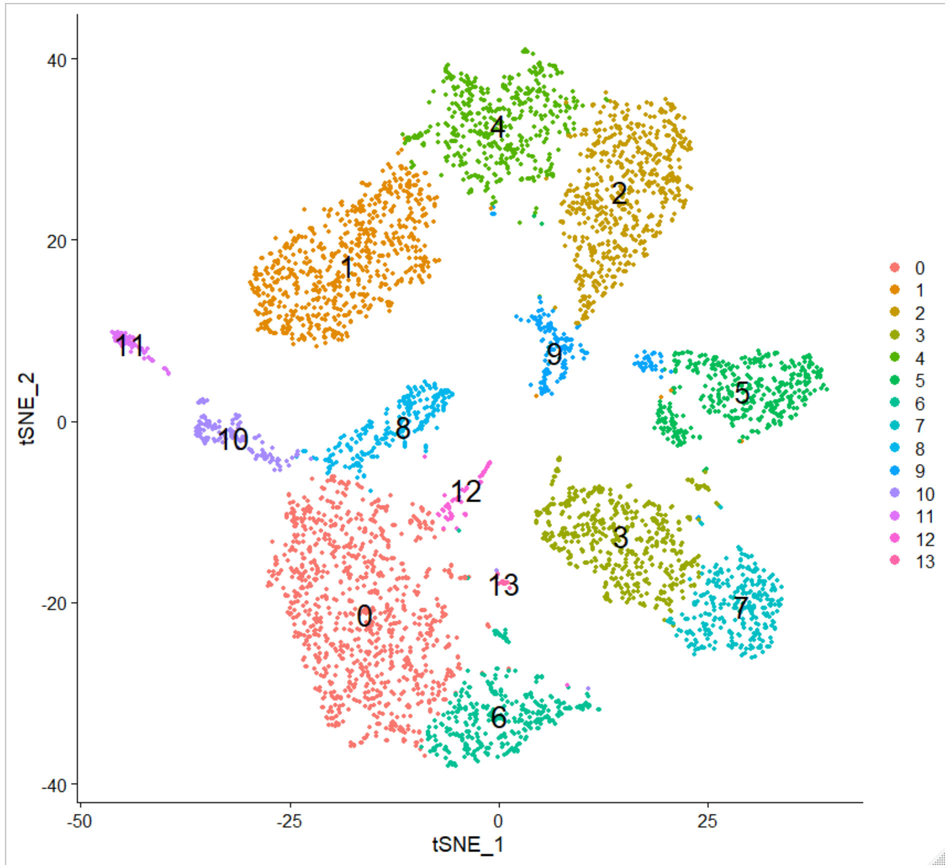
Cluster Properties - Cluster Sizes



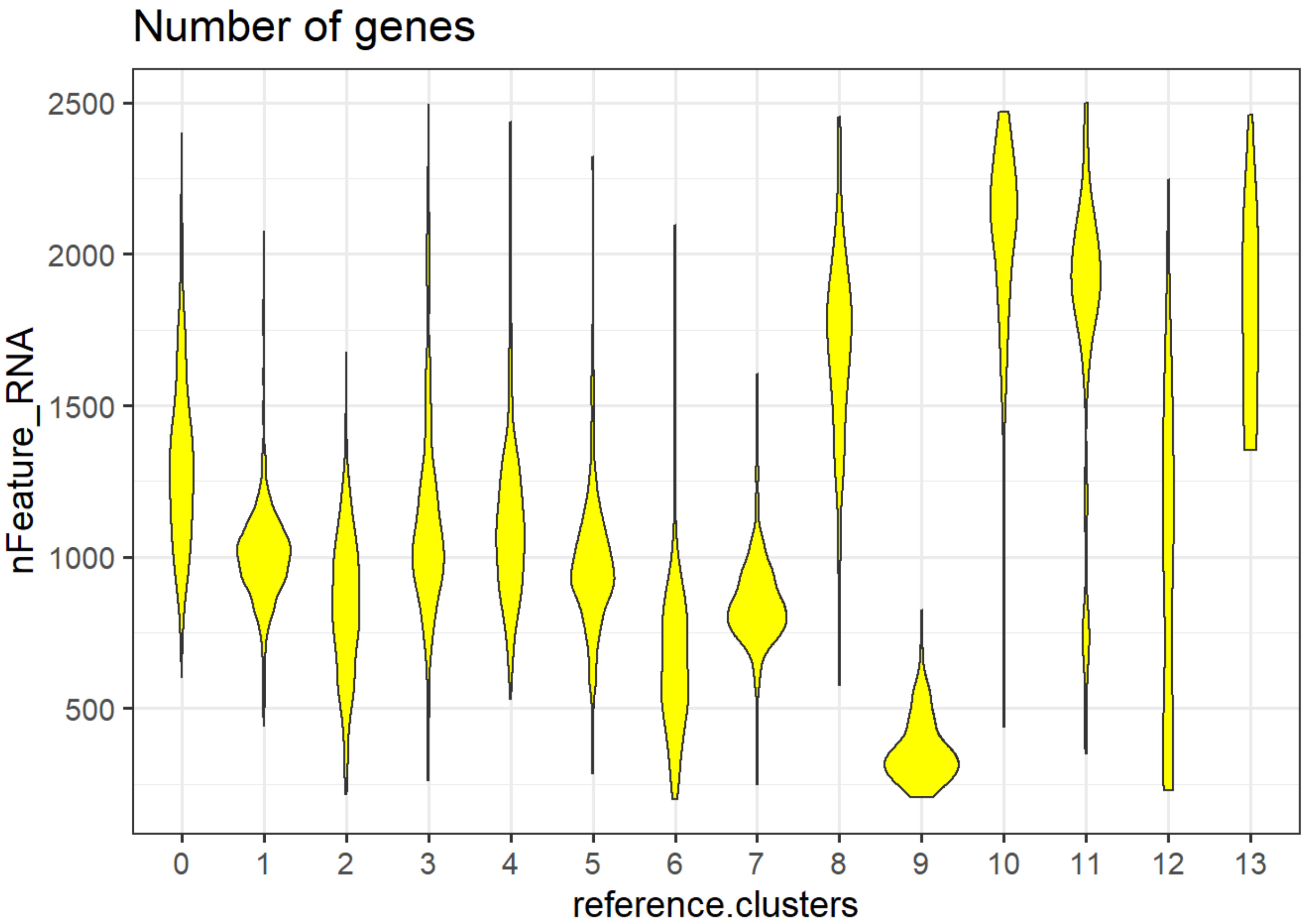


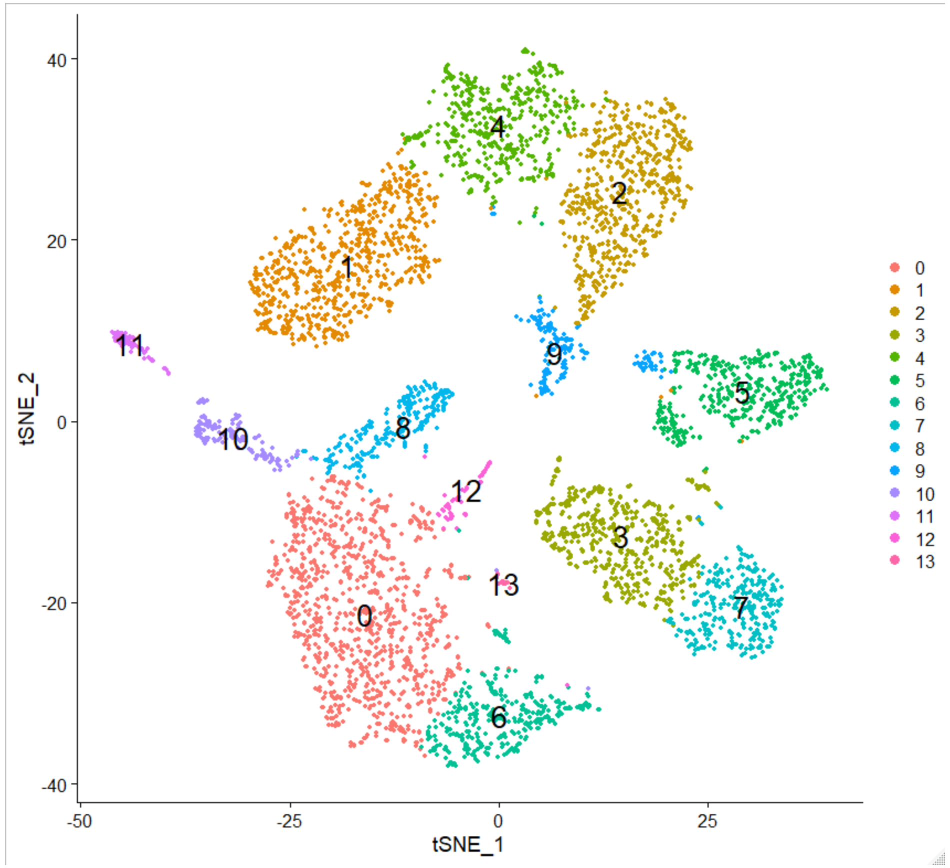
Cluster Properties – Reads per cell



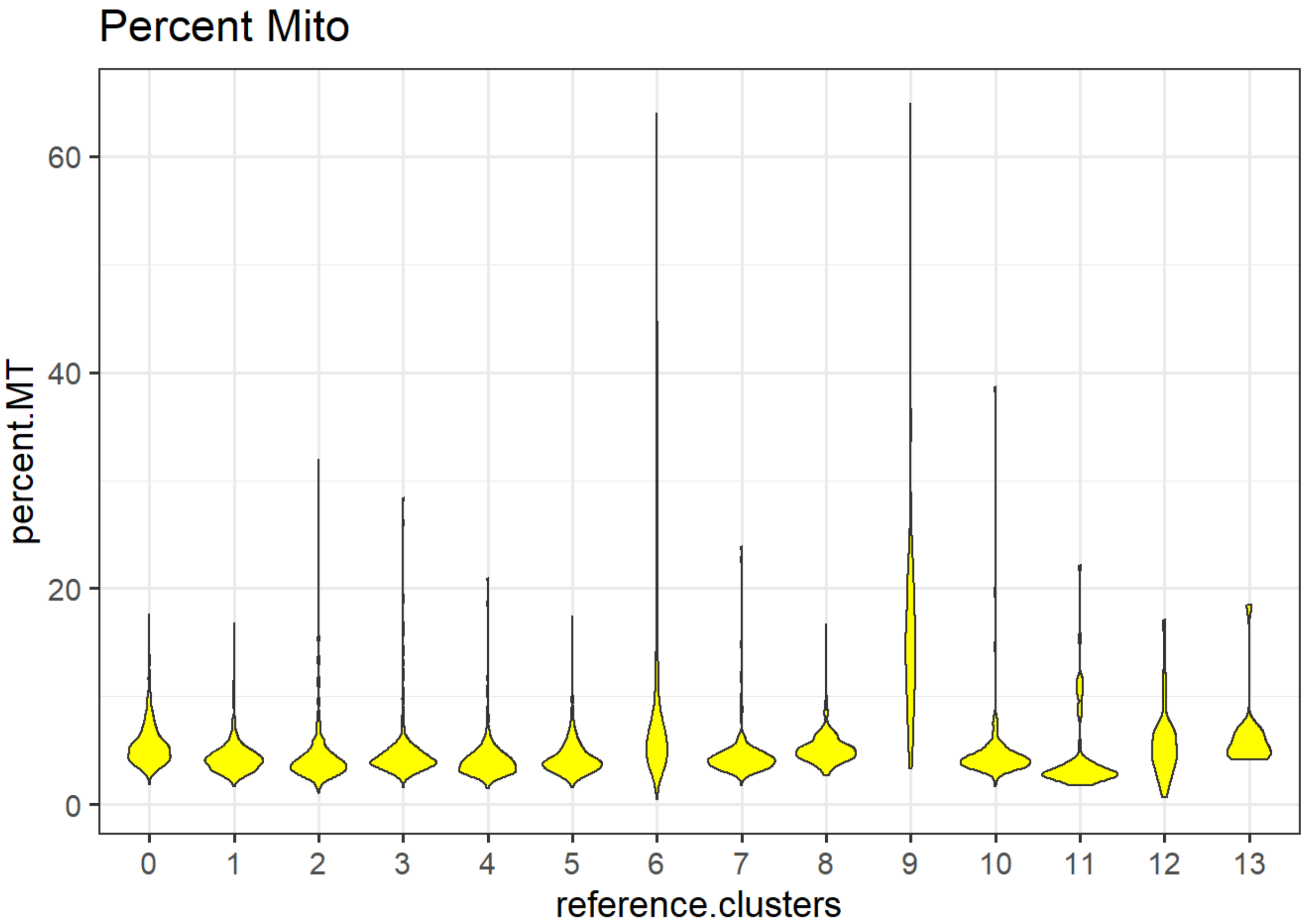


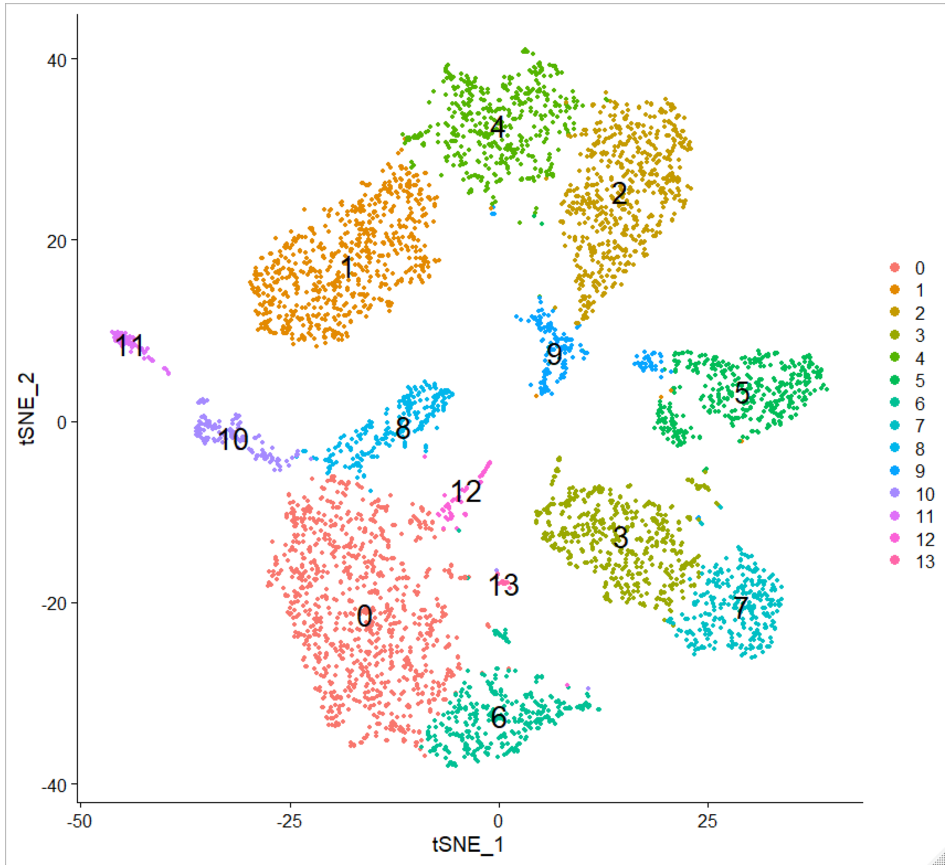
Cluster Properties – Genes per cell



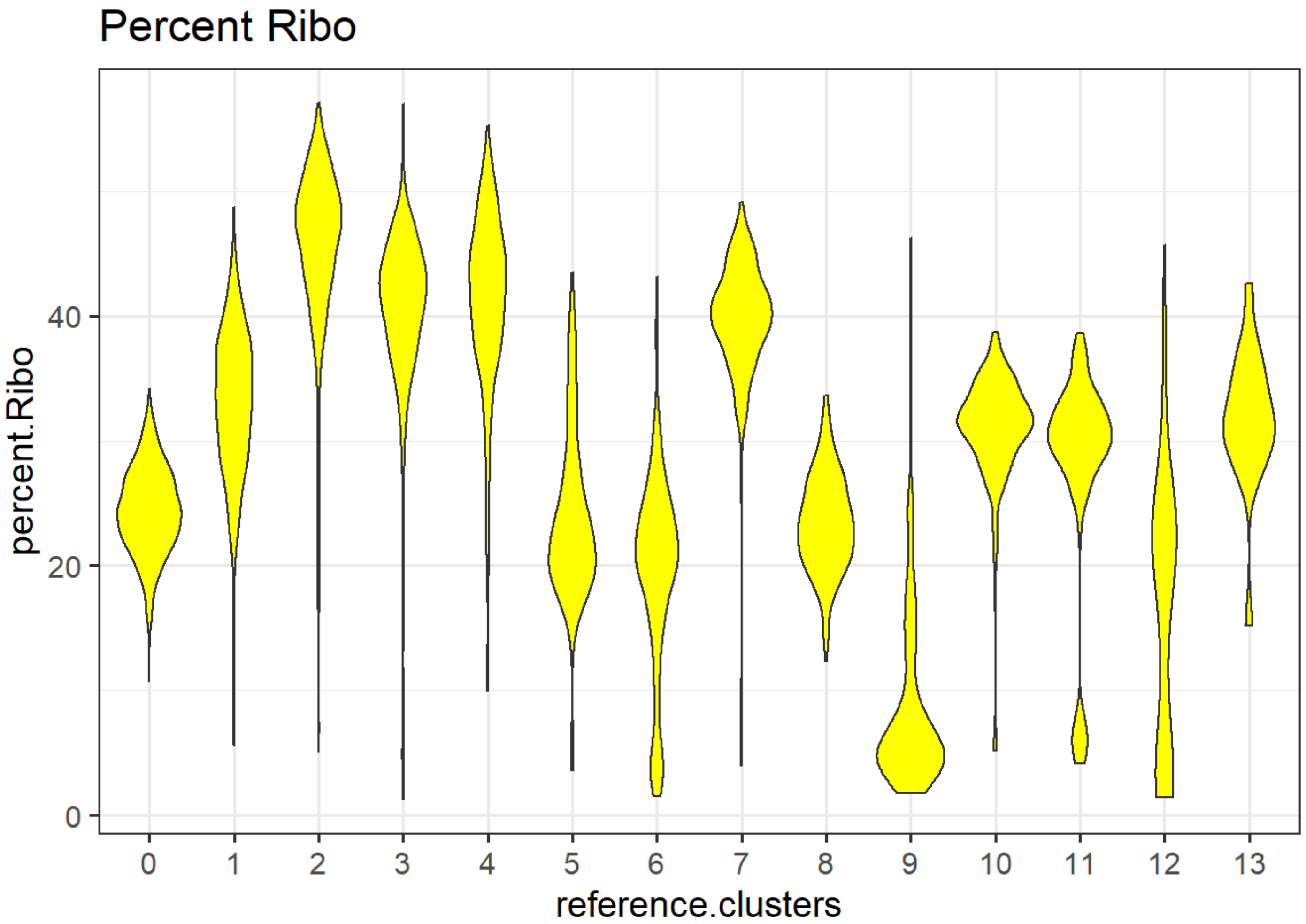


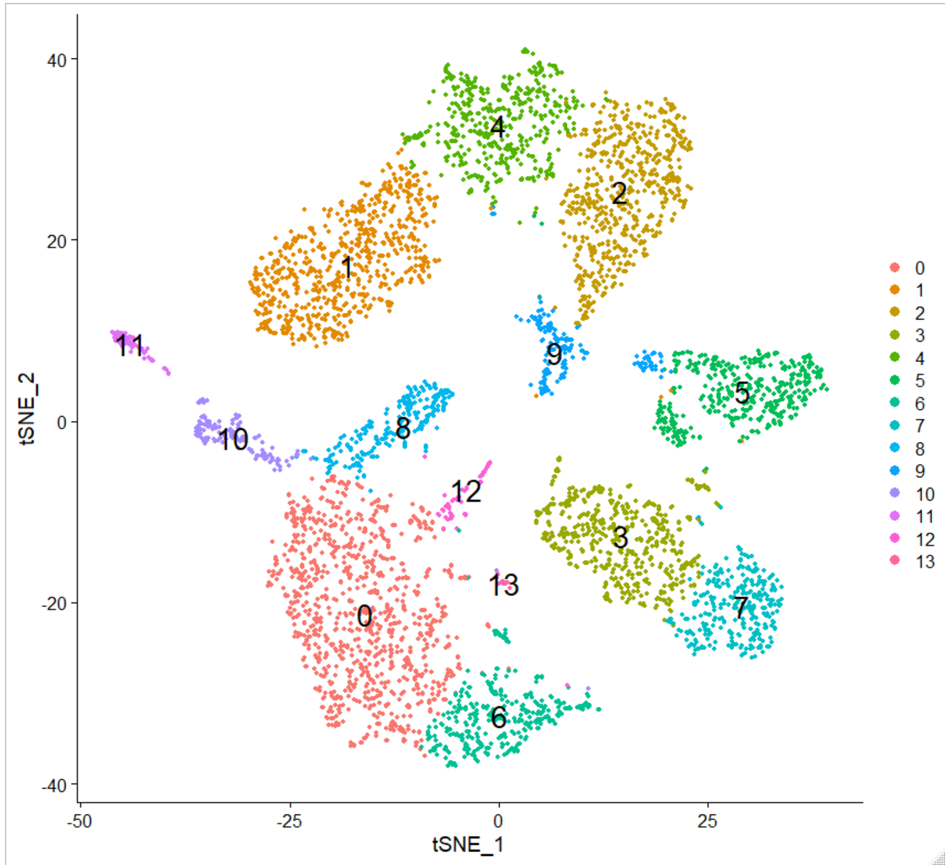
Cluster Properties – Amount of MT



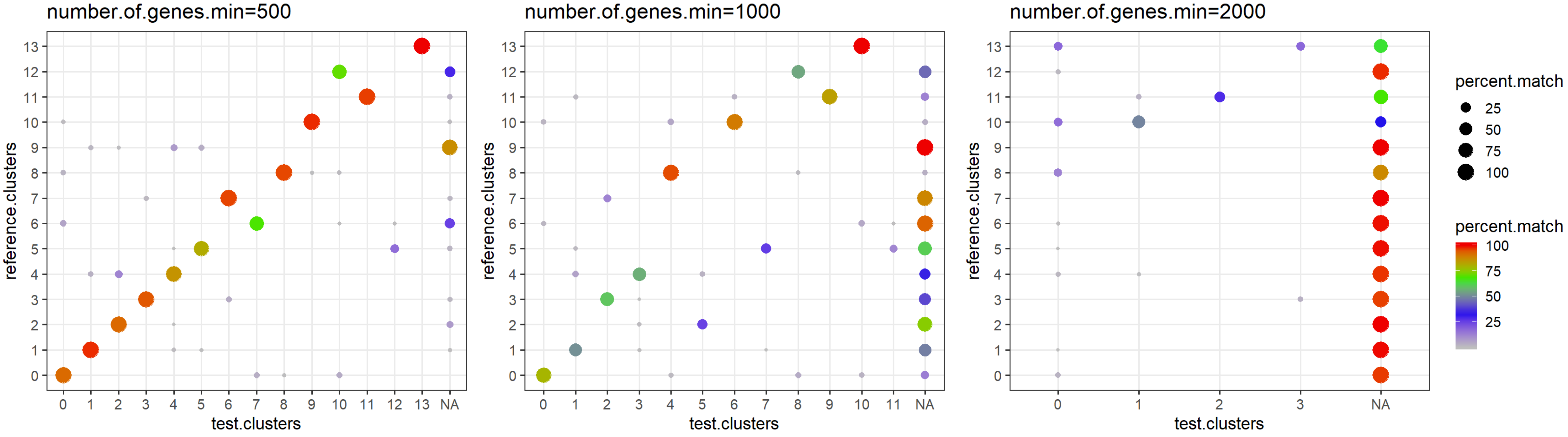


Cluster Properties – Amount of Ribosomal

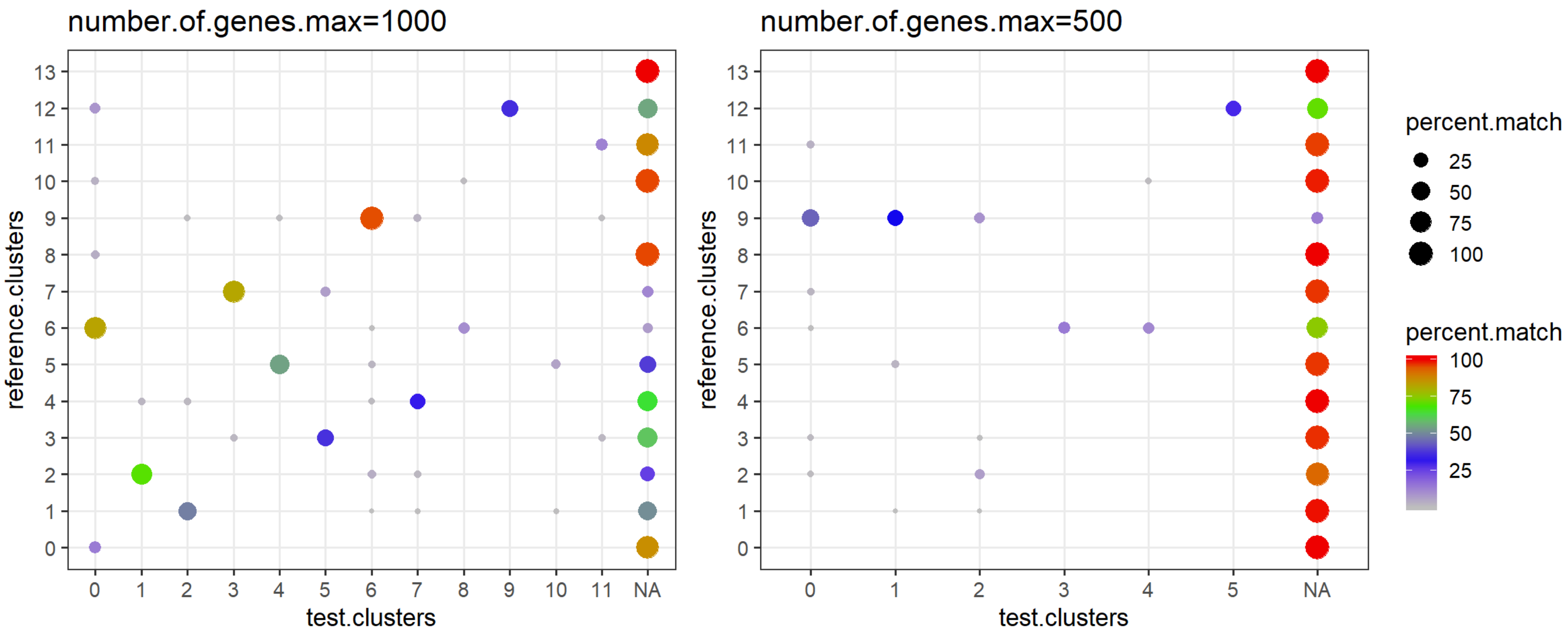




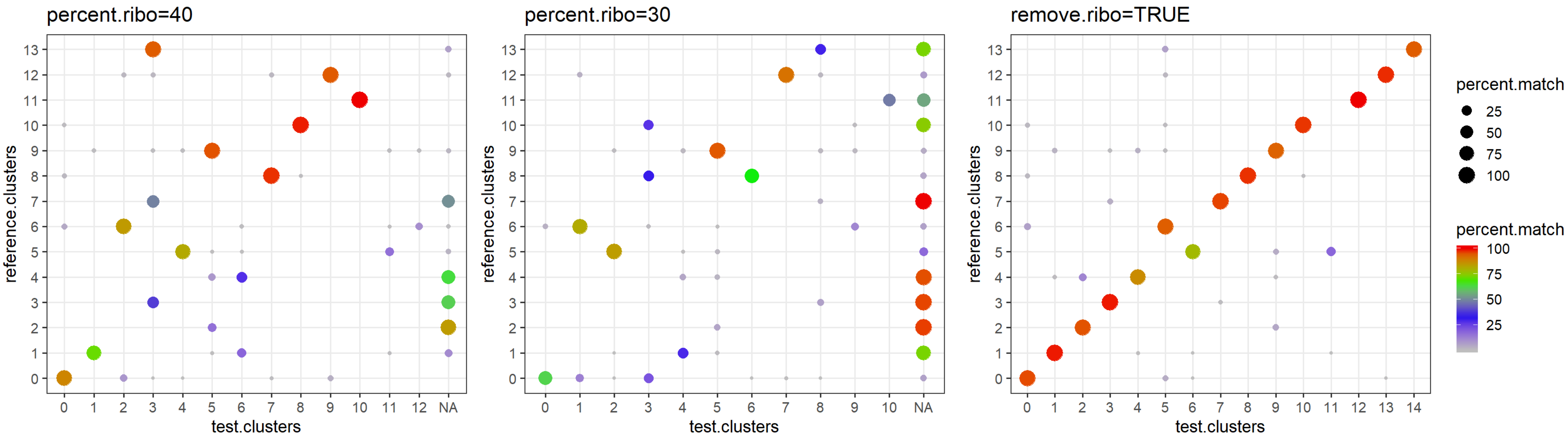
Varying Parameters – Min Genes per Cell (200)



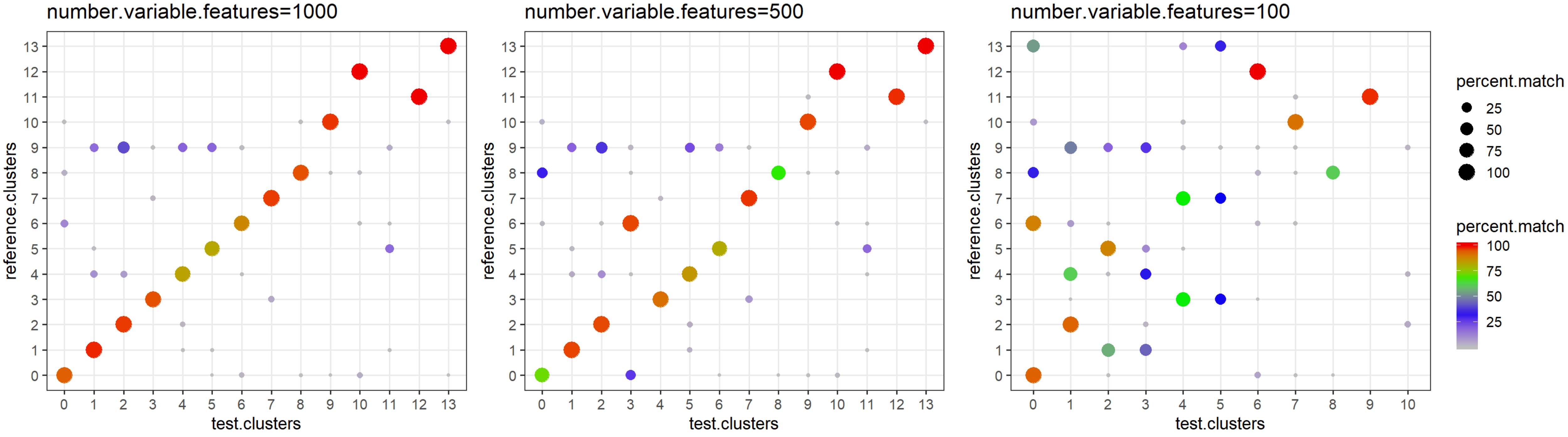
Varying Parameters – Max Genes per Cell (2500)



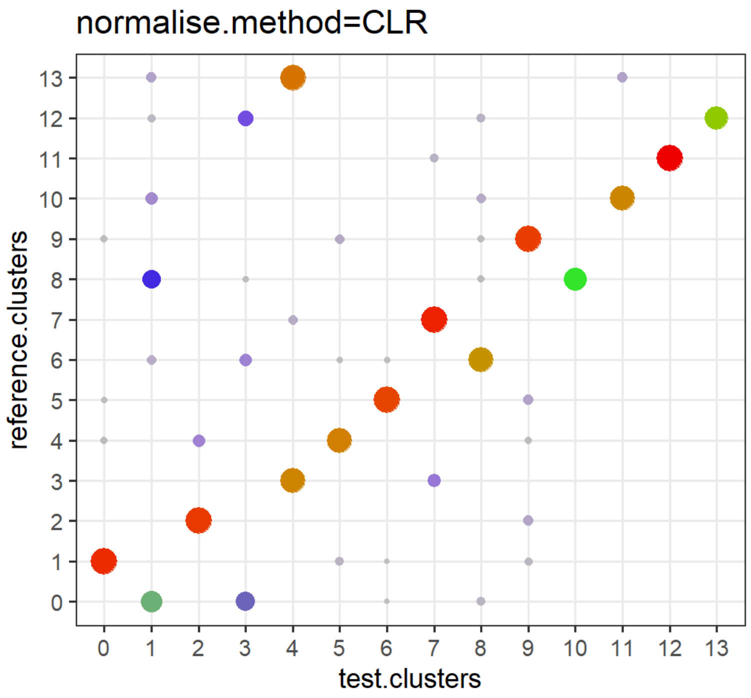
Varying Parameters – Ribo Removal (100)



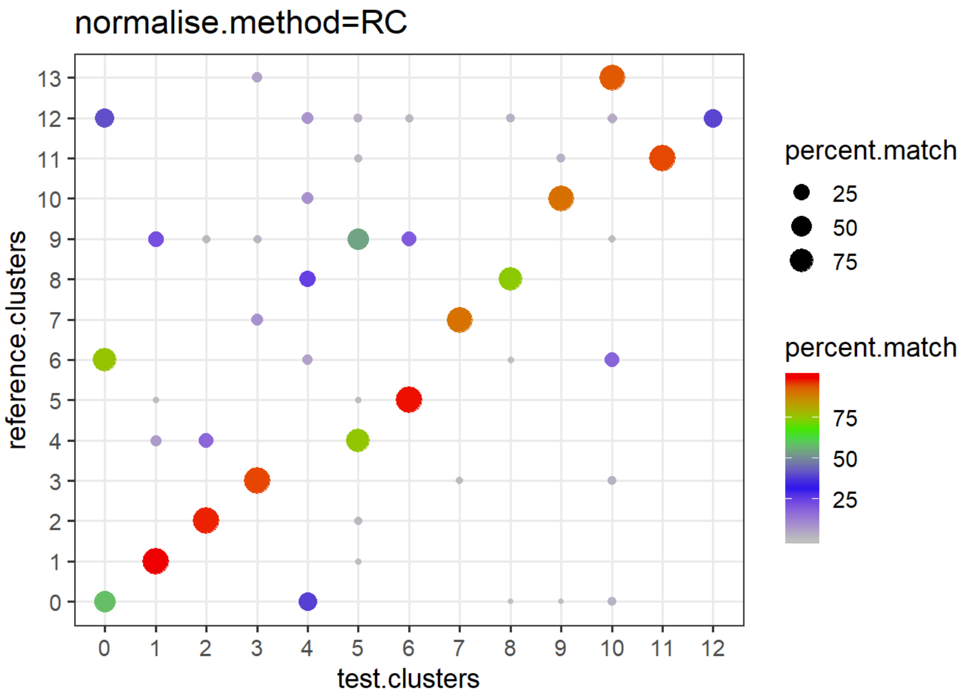
Varying Parameters – Variable Features (2000)



Varying Parameters – Normalisation (Log Norm)

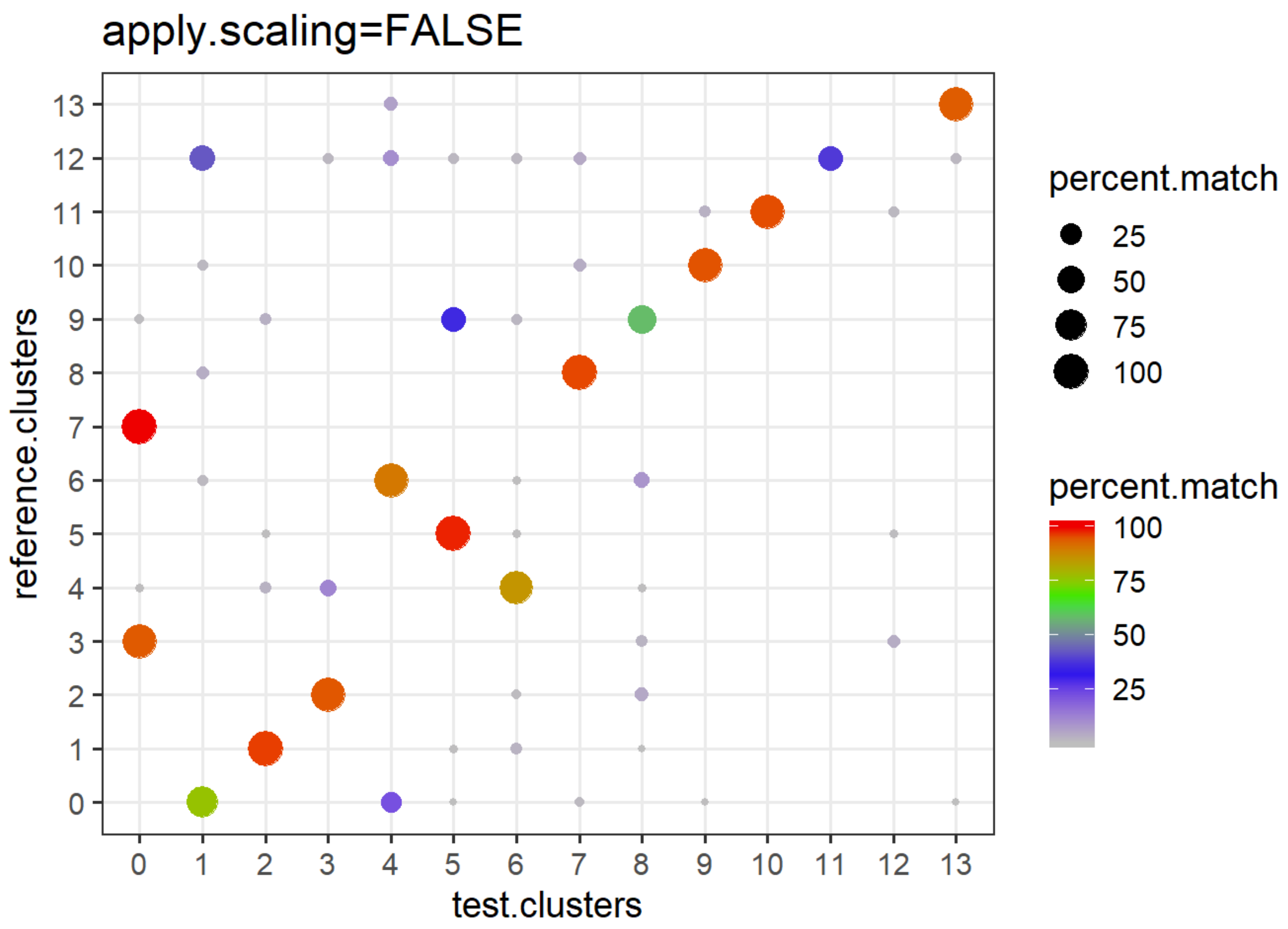


Centred Log Ratio
Similar to Size Factors

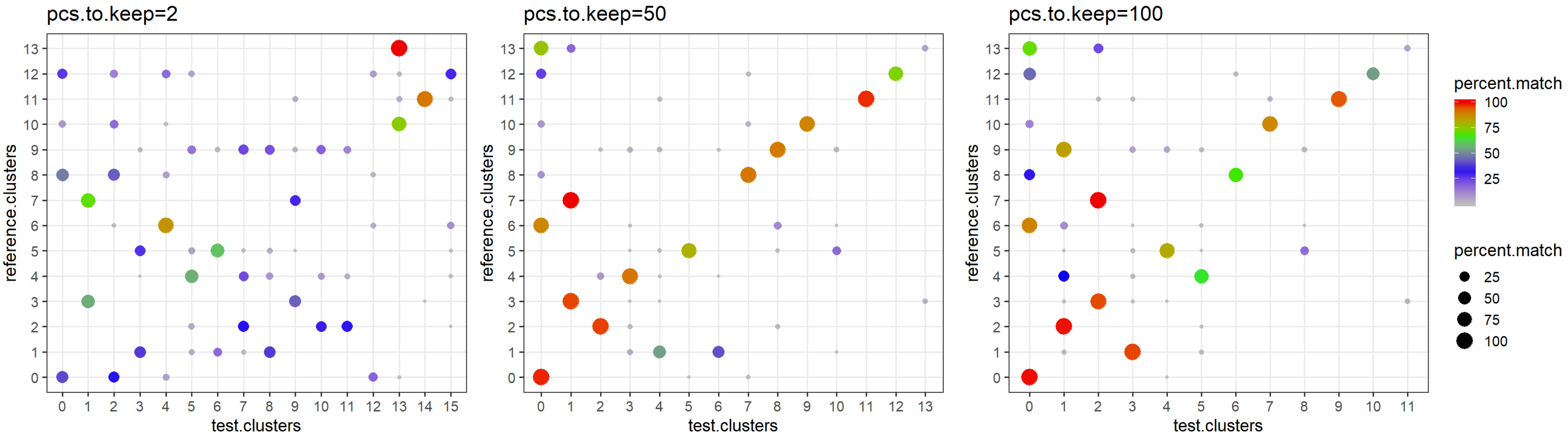


Relative counts
Linear Globally Normalised

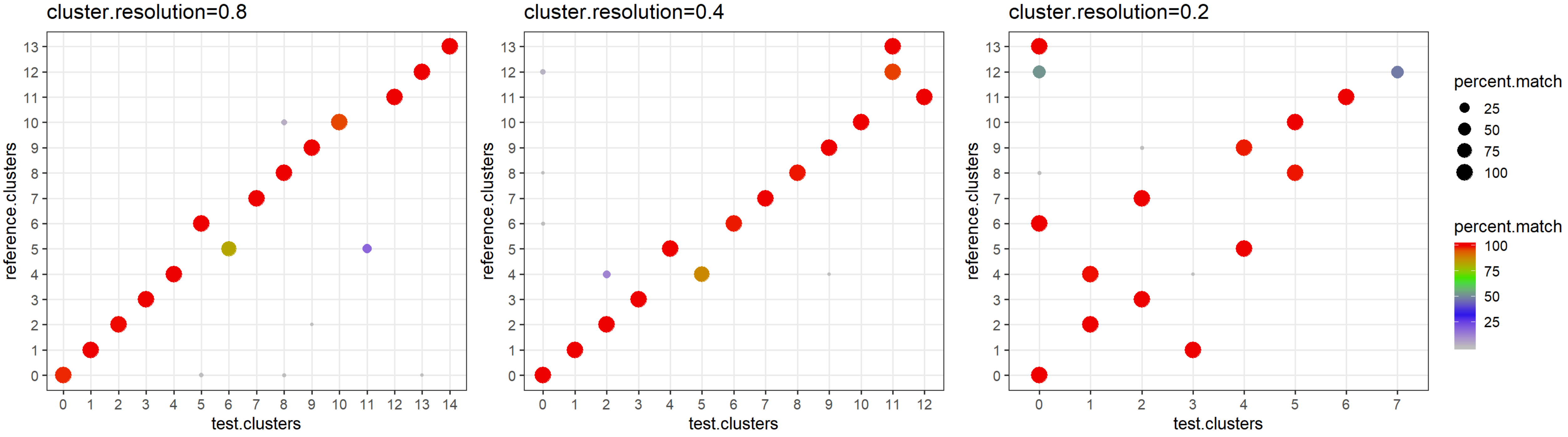
Varying Parameters – Apply Scaling (TRUE)



Varying Parameters – PCs to keep (10)



Varying Parameters – Cluster Resolution (0.5)


cluster.resolution=0.8
cluster.resolution=0.4
cluster.resolution=0.2
reference.clusters
test.clusters
percent.match
percent.match