

Dimensionality reduction, differential expression analysis using single-cell RNA-seq data

Allen Zhang

Dimensionality reduction

Clustering

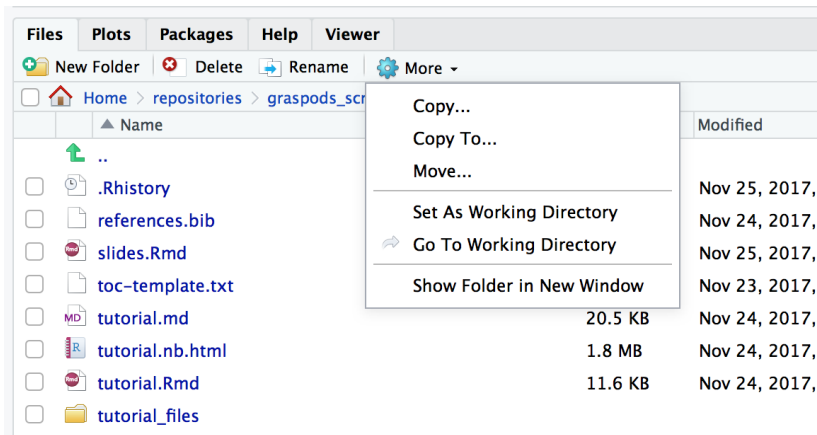
Differential expression

Install required packages

```
if (!require("pacman")) install.packages("pacman")  
  
pacman::p_load(tidyverse, stringr,  
               Rtsne, cluster, limma, edgeR, statmod)
```

- Make sure you have the latest version of RStudio (<https://www.rstudio.com/products/rstudio/download/>).

Navigate to working directory



Load the expression data

```
load("../data/pbmc3k_seurat_extracted.rda")
```

- `data_matrix_raw` contains the transcript (technically, UMI) counts for each gene, cell pair
- `data_matrix_scaled` contains the log-normalized and scaled version of the above
- `pbmc_metadata` contains metadata for each cell
- `variable_genes` contains a vector of genes that exhibit substantial expression variability across cells

```
dim(data_matrix_raw)
```

```
## [1] 13714 2700
```

```
dim(pbmc_metadata)
```

```
## [1] 2638 7
```

```
length(variable_genes)
```

```
## [1] 1838
```

```
dim(data_matrix_scaled)
```

```
## [1] 1838 2638
```

```
data_matrix_raw <-  
  as.matrix(data_matrix_raw[variable_genes,  
                           colnames(data_matrix_scaled)])
```

Dimensionality reduction

Dimensionality reduction

Goal:

- Reduce 1838-dimensional data into lower dimensions
- Why?
 - Visualization
 - Clustering

Dimensionality reduction

How?

- Principal component analysis (PCA)
- t-distributed stochastic neighbour embedding (t-SNE)

Principal component analysis (PCA)

- Expression data for each gene \Rightarrow sum of orthogonal components
- Allows us to write data in terms of these components

```
components <- 20
```

Running PCA

```
pca_results <-  
  irlba::irlba(A = t(data_matrix_scaled),  
               nv = components)
```

Results

```
dim(pca_results$u)
```

```
## [1] 2638 20
```

- `pca_results$u` is the **factor score** matrix: the values of each PC for each cell

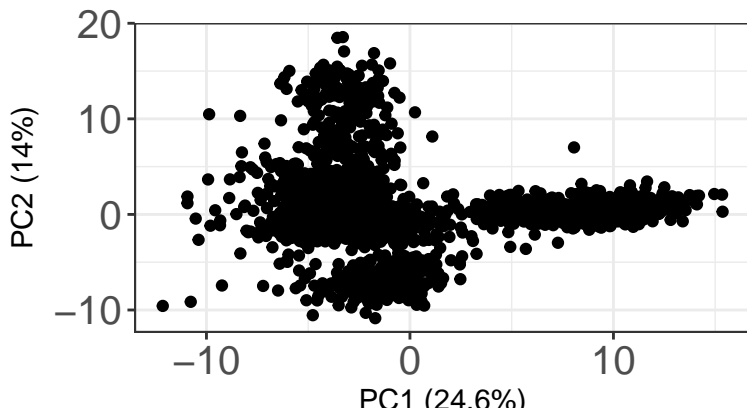
```
dim(pca_results$v)
```

```
## [1] 1838 20
```

- `pca_results$v` is the **loadings** matrix: how to convert PC values back to expression values for each gene

Visualization

```
ggplot(factor_scores, aes(x=PC1, y=PC2)) + geom_point() +  
  theme_bw() +  
  xlab(paste0("PC1 (", round(frac_vars[1],3)*100, "%)")) +  
  ylab(paste0("PC2 (", round(frac_vars[2],2)*100, "%)")) +  
  theme(axis.text = element_text(size=15))
```



t-distributed stochastic neighbour embedding (t-SNE)

- Nonlinear dimensionality reduction method

```
dims <- 2
```

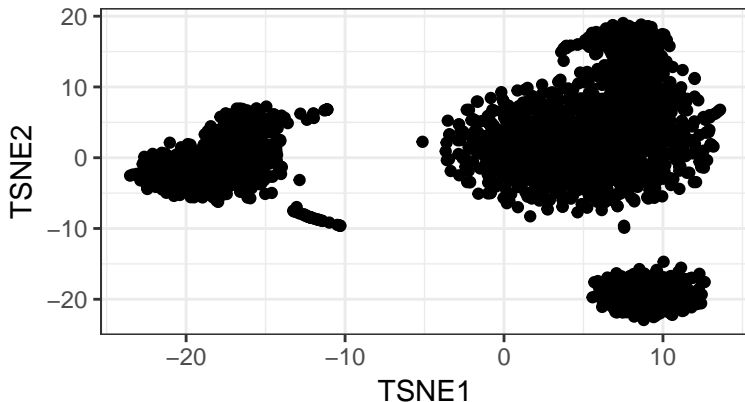
Running t-SNE

```
tsne_results <- Rtsne(X = t(data_matrix_scaled),  
                      dims = dims, initial_dims = 50)
```

```
tsne_results <-  
  readRDS("../intermediates/tsne_results.rds")
```

Plotting

```
ggplot(tsne_df, aes(x=TSNE1, y=TSNE2)) +  
  geom_point() + theme_bw() +  
  xlab("TSNE1") + ylab("TSNE2")
```



Clustering

k-medoids

Many ways to cluster, including:

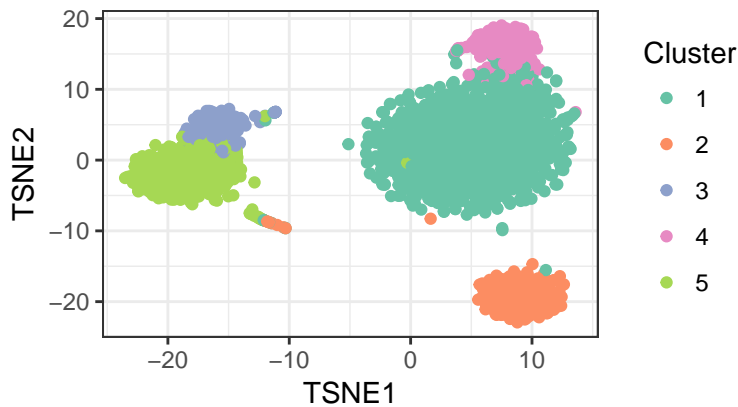
- k-means/k-medoids (k can be selected with silhouette, elbow method, etc.)
- dynamic tree cut method (Langfelder, Zhang, and Horvath 2008)

For speed we'll do partitioning around medoids (PAM) with 5 clusters, a greedy implementation of k-medoids.

```
pam_results <- pam(factor_scores, k = 5,  
                    metric = "euclidean",  
                    diss = FALSE)  
  
tsne_df$cluster <- pam_results$clustering
```

Plotting

```
ggplot(tsne_df, aes(x=TSNE1, y=TSNE2)) +  
  geom_point(aes(colour=factor(cluster))) +  
  theme_bw() + xlab("TSNE1") + ylab("TSNE2") +  
  guides(colour = guide_legend(title = "Cluster")) +  
  scale_colour_brewer(palette = "Set2")
```



Differential expression

What methods to use?

- DE methods for bulk RNA-seq data as good as those for single-cell RNA-seq (Soneson and Robinson 2017)

Preparing the design matrix

```
cluster_label <- paste0("C", tsne_df$cluster)

design <- model.matrix(~0 + cluster_label)
colnames(design) <- paste0("C", 1:5)
```

Fitting a GLM

```
d <- DGEList(data_matrix_raw)
d <- calcNormFactors(d)

d <- estimateDisp(d, design)
fit <- glmFit(d, design, robust = TRUE)
```

```
fit <- readRDS("../intermediates/edgeR_glmfit.rds")
```

Comparing clusters 1 and 2

```
contrast.matrix <-  
  limma::makeContrasts(C1 - C2, levels = design)  
colnames(contrast.matrix) <- c("C1C2")
```

Performing the likelihood test:

```
results <- glmLRT(fit, coef = "C1 - C2",  
                  contrast = contrast.matrix)  
c1_c2_genes <-  
  edgeR::topTags(results, adjust.method = "BH",  
                  n = Inf, sort.by = "PValue")
```


Underexpressed in cluster 1 vs. 2

```
c1_c2_genes$table %>%
  subset(logFC < 0, select = c(logFC, FDR)) %>%
  head(5)
```

##		logFC	FDR
##	CD79A	-4.656100	0.000000e+00
##	CD79B	-3.810178	0.000000e+00
##	MS4A1	-3.680759	0.000000e+00
##	HLA-DRB1	-3.663803	0.000000e+00
##	HLA-DPA1	-3.564842	7.195417e-312

CD79A

From Wikipedia, the free encyclopedia

Cluster of differentiation CD79A also known as **B-cell antigen receptor complex-associated protein alpha chain** and **MB-1 membrane glycoprotein**, is a [protein](#) that in humans is encoded by the CD79A [gene](#).^[5]

The CD79a protein together with the related [CD79b](#) protein, forms a [dimer](#) associated with membrane-bound [immunoglobulin](#) in [B-cells](#), thus forming the B-cell antigen receptor (BCR).^[6] This occurs in a similar manner to the association of [CD3](#) with the [T-cell receptor](#), and enables the cell to respond to the presence of [antigens](#) on its surface.^[7]

It is associated with [agammaglobulinemia-3](#).^[8]

Overexpressed in cluster 1 vs. 2

```
c1_c2_genes$table %>%  
  subset(logFC > 0, select = c(logFC, FDR)) %>%  
  head(5)
```

##		logFC	FDR
##	IL32	3.693076	1.582355e-183
##	GIMAP7	2.776935	2.128554e-94
##	GIMAP4	2.432430	5.059254e-65
##	CD2	2.344373	1.014307e-55
##	GIMAP5	2.232089	4.864995e-43

CD2

From Wikipedia [Discussion about the content page \[ctrl-option-t\]](#)

CD2 (cluster of differentiation 2) is a [cell adhesion molecule](#) found on the surface of [T cells](#) and [natural killer \(NK\) cells](#). It has also been called T-cell surface antigen T11/Leu-5, LFA-2,^[5] LFA-3 receptor, erythrocyte receptor and rosette receptor.^[6]

Comparison to Seurat-annotated clusters

```
tsne_df$annotated_celltype <-  
  pbmc_metadata[tsne_df$cell,]$ClusterNames_0.6  
  
with(tsne_df, table(annotated_celltype, cluster))
```

##	cluster				
## annotated_celltype	1	2	3	4	5
## B cells	4	339	0	0	0
## CD14+ Monocytes	0	0	7	0	471
## CD4 T cells	1146	0	0	0	3
## CD8 T cells	228	1	1	78	0
## Dendritic cells	7	11	0	0	14
## FCGR3A+ Monocytes	0	0	156	0	2
## Megakaryocytes	1	0	13	0	1
## NK cells	2	0	0	153	0

Questions?

References

Langfelder, Peter, Bin Zhang, and Steve Horvath. 2008. "Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R." *Bioinformatics* 24 (5): 719–20.

doi:10.1093/bioinformatics/btm563.

Soneson, Charlotte, and Mark D. Robinson. 2017. "Bias, Robustness and Scalability in Differential Expression Analysis of Single-Cell RNA-Seq Data." *Doi.org*, May. Cold Spring Harbor Laboratory, 143289. doi:10.1101/143289.