BIMS 8702 Assignment 7: Dimensionality Reduction for Single-Cell RNA-seq

Due April 7, 2025

This assignment is to use Seurat (an R package for single-cell genomics data analysis) to perform dimensionality reduction and data visualization for a single-cell RNA-seq dataset for a human peripheral blood mononuclear cell (PBMC) sample.

1. Download the single-cell RNA-seq processed data (pre-processed with Cell Ranger) from this link:
https://cf.10xgenomics.com/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz

Unzip the tarball data package and local the data in your working directory.

2. Install the Seurat R package and dependent packages, and load the data:

```
> install.packages('Seurat')
```

Also install '`ggplot2`', '`splyr`', and '`patchwork`' packages if necessary

Load the packages and the downloaded dataset:

```
> load(dplyr)
> load(Seurat)
> load(patchwork)
> pbmc.data <- Read10X(data.dir =
"/yourpath/pbmc3k_filtered_gene_bc_matrices/hg19/")
> pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k",
min.cells = 3, min.features = 200)
```

3. Preprocess: normalize and scale data and identify high-variable genes
Data normalization:

```
> pbmc <- NormalizeData(pbmc)
```

You can adjust the normalization details by specify parameters, e.g.,

```
> pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize",
scale.factor = 10000)
```

Identify high-variable genes:

```
> pbmc <- FindVariableFeatures(pbmc, selection.method = "vst",
nfeatures = 2000)
```

Scale the data:

```
> all.genes <- rownames(pbmc)
> pbmc <- ScaleData(pbmc, features = all.genes)
```

4. Dimensionality Reduction using PCA:

```
> pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```

You can examine and visualize PCA results in a few different ways, e.g.,

```
> print(pbmc[["pca"]], dims = 1:5, nfeatures = 5)
```

Seurat provides several ways of visualizing both cells and features that define the PCA, try the plots below

```
> VizDimLoadings(pbmc, dims = 1:2, reduction = ”pca”)
> DimPlot(pbmc, reduction = ”pca”) + NoLegend()
> DimHeatmap(pbmc, dims = 1, cells = 500, balanced = TRUE)
```

You can use ‘Elbow plot’ to determine the number of dimsions (PCs) to be used for the following analyses.

```
> ElbowPlot(pbmc)
```

Please discuss how many PCs you want to use and explain your rationales.

Then choose the top 5, 10, 20, 100 PCs and complete the following analysis for each one.

5. Cell clustering:
Seurat first constructs a KNN graph based on the Euclidean distance in PCA space, and refine the edge weights between any two cells based on the shared overlap in their local neighborhoods (Jaccard similarity). This step is performed using the `FindNeighbors()` function, and takes as input the pre-defined dimensionality choice. e.g.,

```
> pbmc <- FindNeighbors(pbmc, dims = 1:5)
```

To cluster the cells, we next apply modularity optimization techniques such as the Louvain algorithm (default), to iteratively group cells together, with the goal of optimizing the standard modularity function. The `FindClusters()` function implements this procedure, and contains a resolution parameter that sets the ‘granularity’ of the downstream clustering, with increased values leading to a greater number of clusters. The clusters can be found using the `Idents()` function.

```
> pbmc <- FindClusters(pbmc, resolution = 0.5)
```

6. Data visualization
Please use both t-SNE and UMAP (sample command below) for visualization

```
> pbmc <- RunTsne(pbmc, dims = 1:5)
> pbmc <- RunUMAP(pbmc, dims = 1:5)
```

Then show t-SNE or UMAP visualization with clustering labels (colored)
# note that you can set `` `label = TRUE` `` or use the `LabelClusters` function to help label

```
> DimPlot(pbmc, reduction = ”umap”)
```

7. Discuss your findings.

8. (Optional) You can use the `Rtsne` package and test t-SNE results’ sensitivity to the perplexity parameter. Set different perplexity parameters (e.g., 20, 50, 100) and compare the results.