scClassifier-Human-PBMC

Feng Zeng

2019/12/18

Basic Steps of scClassifier

scClassifier provides a supervised method for the identification of common and rare cell types. scClassifier is comprised of two components. The first component is a probabilistic bag-of-standard-cells model to characterize the transcriptome complexity associated with cell types. The second component is a cell network to tackle the interference of sequencing noises on cell type prediction.

Therefore, the basic steps of the use of scClassifier are as follows,

1. Select a set of genes that are informative;
2. Train the bag models and then predict cell types for single cells;
3. Use the network ensemble to tackle the disturbance of sequencing noises.

This quick tutorial is to walk through the above steps of scClassifier by applying it to analyze a simple PBMC scRNA-seq dataset.

Before going through the tutorial, users should download the PBMC scRNA-seq dataset and the reference database from Figshare, and put these two datasets in the tutorial directory.

Datasets can be retrieved here https://figshare.com/articles/scClassifier_s_tutorial_datasets/11407743.

Step Zero

First, we load the required packages, PBMC scRNA-seq dataset, and the reference database.

```r
# load packages
suppressPackageStartupMessages(
  suppressWarnings({
  library(dplyr)
  library(scClassifier)
  library(SingleCellExperiment)
}))


# load PBMC dataset
# The scRNA-seq data is a Seurat object.
load("pbmc3k.RData")

# load Immuno-Navigator database as reference
# The reference database is a ExpressionSet object.
load("immuno_navigator_human_expression.RData")
immuno.X <- exprs(immuno_navigator_human_expression)
immuno.X[immuno.X < 0] <- 0
immuno.y <- pData(immuno_navigator_human_expression)
```

Step One

This step is to select the informative genes. The procedures are as follows:

1. Collect highly variable genes;
2. Collect significant genes from PCA;
3. Combine genes collected at 1 and 2;
4. Identify differentially expressed genes out of genes collected at 3.

The above procedure is wrapped in SeuratDEGenes as a function of scClassifier.

```
genes.use <- SeuratDEGenes(pbmc3k@raw.data %>% as.matrix, min.diff.pct = 0.01)
genes.use <- intersect(genes.use, rownames(immuno.X))
```

Step Two

This step is to predict cell types for single cells. The training procedure is built in DirichletMultinomialClassifier.

```
z <- DirichletMultinomialClassifier(pbmc3k@data,
                                    immuno.X,
                                    immuno.y$type,
                                    genes.use = genes.use)
```

Step Three

This step is to tackle the disturbance of technical noises by cell network ensemble.

```
z.smooth <- DiffusionKernelAveraging(pbmc3k@data %>% as.matrix,
                                     t(z$score.matrix),
                                     genes.use = genes.use,
                                     tau = 15)
```

The classification accuracy is calculated as follows.

```
k <- z.smooth$celltype %>% as.character
message(cat("The accuracy is ", sum(k == pbmc3k@meta.data$type) / length(pbmc3k@cell.names)))
```

```
## The accuracy is  0.9613333

##
```