PBMCs Example

```r
library(org.Hs.eg.db)
library(Seurat)
source('cell_type_identification2.R')
```

We start by reading in our training data, which is downloaded from 10X Genomics: CD4, CD8, CD14, and NK cells. For the purposes of this example, we will withhold 100 cells of each to serve as test data. After loading in the data, we convert the gene symbols to Ensembl IDs (required).

```r
set.seed(6619)

### Preparation for symbol->ENSEMBL conversion
Hs_symbol <- org.Hs.egSYMBOL
mapped_Hs_genes.symbol <- mappedkeys(Hs_symbol)
Hs_symbol.df <- as.data.frame(Hs_symbol[mapped_Hs_genes.symbol])
Hs_ensembl <- org.Hs.egENSEMBL
mapped_Hs_genes.ensembl <- mappedkeys(Hs_ensembl)
Hs_ensembl.df <- as.data.frame(Hs_ensembl[mapped_Hs_genes.ensembl])
Hs_mapping <- merge(Hs_symbol.df,Hs_ensembl.df)

## CD4: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/cd4_t_helper?
cd4_facs.data <- Read10X(data.dir = "../cd4_singlecell/")
cd4_facs.data <- as.matrix(cd4_facs.data)
rownames(cd4_facs.data) <- Hs_mapping$ensembl_id[match(rownames(cd4_facs.data),Hs_mapping$symbol)]
cd4_facs.data <- cd4_facs.data[!is.na(rownames(cd4_facs.data)),]
cd4_facs.data <- na.omit(cd4_facs.data)
cd4.test <- cd4_facs.data[,1:100] # CD4 withheld cells
cd4_facs.data <- cd4_facs.data[,101:11213] # CD4 training cells

## CD8: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/cytotoxic_t
cd8_facs.data <- Read10X(data.dir = '../filtered_matrices_cd8/hg19/')
cd8_facs.data <- as.matrix(cd8_facs.data)
rownames(cd8_facs.data) <- Hs_mapping$ensembl_id[match(rownames(cd8_facs.data),Hs_mapping$symbol)]
cd8_facs.data <- cd8_facs.data[!is.na(rownames(cd8_facs.data)),]
cd8.test <- cd8_facs.data[,1:100] # CD8 withheld cells
cd8_facs.data <- cd8_facs.data[,101:10209] # CD8 training cells

## CD14: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/cd14_monocytes
cd14_facs.data <- Read10X(data.dir = '../filtered_matrices_cd14/hg19/')
cd14_facs.data <- as.matrix(cd14_facs.data)
rownames(cd14_facs.data) <- Hs_mapping$ensembl_id[match(rownames(cd14_facs.data),Hs_mapping$symbol)]
cd14_facs.data <- cd14_facs.data[!is.na(rownames(cd14_facs.data)),]
cd14.test <- cd14_facs.data[,1:100] # CD14 withheld cells
cd14_facs.data <- cd14_facs.data[,101:2612] # CD14 training cells

## NK: https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/cd56_nk
nk_facs.data <- Read10X(data.dir = '../filtered_matrices_nk/hg19/')
nk_facs.data <- as.matrix(nk_facs.data)
```

```r
rownames(nk_facs.data) <- Hs_mapping$ensembl_id[match(rownames(nk_facs.data),Hs_mapping$symbol)]
nk_facs.data <- nk_facs.data[!is.na(rownames(nk_facs.data)),]
nk.test <- nk_facs.data[,1:100] # NK withheld cells
nk_facs.data <- nk_facs.data[,101:8385] # NK training cells
```

Next, we fit our model to these training data. This can be done in two ways. We could call the `trainReference` function individually on the vector of total gene counts for each cell-type, or we could combine the training data into a single matrix and call the `trainAllReference` function on this matrix, alongside the vector of cell-type labels. There is no real difference between the two (`trainAllReference` will simply split the matrix into the different cell-types according to your provided labels and then call `trainReference` on each one), so whichever is easiest to use will depend on whether your different cell-types are already in separate matrices, like here, or not. We show both ways:

```r
## Manually train each cell-type individually
cd4.d <- trainReference(rowSums(cd4_facs.data))
cd8.d <- trainReference(rowSums(cd8_facs.data))
cd14.d <- trainReference(rowSums(cd14_facs.data))
nk.d <- trainReference(rowSums(nk_facs.data))
pbmcs.d <- list(CD4=cd4.d,CD14=cd14.d,CD8=cd8.d,NK=nk.d)

## Or input a single matrix with a vector of labels
common_pbmcs_genes <- Reduce(intersect,list(rownames(cd4_facs.data),rownames(cd14_facs.data),
                                            rownames(cd8_facs.data),rownames(nk_facs.data)))
pbmcs_reference <- as.matrix(cbind(cd4_facs.data[common_pbmcs_genes,],
                                   cd14_facs.data[common_pbmcs_genes,],
                                   cd8_facs.data[common_pbmcs_genes,],
                                   nk_facs.data[common_pbmcs_genes,]))
pbmcs_reference_labels <- c(rep('CD4',dim(cd4_facs.data)[2]),rep('CD14',dim(cd14_facs.data)[2]),
                            rep('CD8',dim(cd8_facs.data)[2]),rep('NK',dim(nk_facs.data)[2]))
pbmcs.d <- trainAllReference(pbmcs_reference,pbmcs_reference_labels)
```

The list `pbmcs.d` contains one table for each inputted cell-type, which indicates each gene's empirical rate, probability of belonging to the off-low latent state, and probability of belonging to the off-high latent state. If we wanted to summarize these more succinctly, we could sum the two probabilities to result in an overall probability of that gene being off in a given cell-type. This could then be used beyond the cell-type annotation context to study gene expression within a single cell-type, to compare genes across cell-types, and to identify markers.

As an example, we show below how we can identify genes with a low probability of being off in CD4 cells, but a high probability of being off in the others (i.e. potential markers for this cell-type):

```r
## Compute overall probability of being off
prob.off <- sapply(pbmcs.d,function(x) x[,2]+x[,3])
colnames(prob.off) <- names(pbmcs.d)
rownames(prob.off) <- rownames(pbmcs.d[[1]])

## Identify genes unlikely to be off in CD4, but likely to be off in the others
head(prob.off[which(prob.off[,1]<0.1&prob.off[,2]>0.75&prob.off[,3]>0.75&prob.off[,4]>0.75),])
```

```
##                          CD4      CD14       CD8        NK
## ENSG00000114737 3.105377e-02 0.9977716 0.7917704 0.9744741
## ENSG00000164530 1.931694e-02 0.9999889 0.9439893 0.9998522
## ENSG00000154016 2.238031e-06 1.0000000 0.8609776 1.0000000
```

Finally, we can use the original object `pbmcs.d` to annotate our withheld test cells, which should be placed in a single matrix (required):

```r
## Put withheld test cells in one matrix
pbmcs_withheld <- as.matrix(cbind(cd4.test[common_pbmcs_genes,],cd14.test[common_pbmcs_genes,],
                                    cd8.test[common_pbmcs_genes,],nk.test[common_pbmcs_genes,]))
true_labels <- c(rep('CD4',100),rep('CD14',100),rep('CD8',100),rep('NK',100))

## Annotate!
annotation <- classifyTarget(pbmcs_withheld,pbmcs.d)
table(annotation,true_labels)
```

```
##             true_labels
## annotation CD14 CD4 CD8 NK
##       CD14   99   0   0  0
##       CD4     0  97   5  0
##       CD8     1   3  95  1
##       NK      0   0   0 99
```