Example_scTypeGSEA

Description

This package is designed to assign cell type labels for each identified cluster in single-cell data. This package uses the "Seurat" R package to do data pre-processing and cell clustering. After clustering the cells, we use differential gene expression analysis to compute the fold-change in gene expression by comparing the cluster profile against all the other identified clusters together. Then we use the Gene Set Enrichment Analysis (GSEA) technique to compute the enrichment (NES and their associated P-value) of marker genes defined for a set of cell types. GSEA analysis provides us the most statistically relevant cell type for each cluster that is finally assigned to the group.

Installation

You can use following codes to install the package. To use some functions in "fgsea", your R version must be >= 3.6.0.

```
# You may need following codes to install dependent packages.
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("fgsea")
BiocManager::install("MAST")
install.packages("Seurat")

library(devtools)
install_github("cailab-tamu/scTypeGSEA")
```

And then you can use following code to load the package.

```
library(scTypeGSEA)
```

One Line example

Here we use a toy data set "pbmc_small" to show our main function "assignCellType". One can do quality control, data pre-process, cluster, get full changes, do GSEA and label the cell in one step.

```
pbmc_res <- assignCellType(obj = pbmc_small,
    min.cells = 1, min.features = 10,
    nfeatures = 100, npcs = 10, dims = 1:10,
    k.param = 5, resolution = 0.75,
    min.pct = 0.25, test.use = "MAST",
    minSize = 5)
```

```
## The input is not a Seurat object, next to transform gene express count matrix to Seurat object.
## The data has not been pre-processed, let's do it!

## Centering and scaling data matrix

## Computing nearest neighbor graph

## Computing SNN

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 80
## Number of edges: 2873
```

```
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.3564
## Number of communities: 2
## Elapsed time: 0 seconds
## Find marker genes for cluster 0
## Assuming data assay in position 1, with name et is log-transformed.
##
## Done!
## Combining coefficients and standard errors
## Calculating log-fold changes
## Calculating likelihood ratio tests
## Refitting on reduced model...
##
## Done!
## Find marker genes for cluster 1
## Assuming data assay in position 1, with name et is log-transformed.
##
## Done!
## Combining coefficients and standard errors
## Calculating log-fold changes
## Calculating likelihood ratio tests
## Refitting on reduced model...
##
## Done!
## Do GSEA for cluster 0
## Do GSEA for cluster 1
```

It will return a list with 2 slots. The first slot is a Seurat object.

```
pbmc_res$obj
```

```
## An object of class Seurat
## 230 features across 80 samples within 1 assay
## Active assay: RNA (230 features)
##  1 dimensional reduction calculated: pca
```

The second slot is a matrix contains cluster-ID, cell type and p-value for each cell.

```
head(pbmc_res$cell_mat)
```

```
##                ClusterID Cell Type             padj
## ATGCCAGAACGACT "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
## CATGGCCTGTGCAT "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
## GAACCTGATGAACC "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
## TGACTGGATTCTCA "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
## AGTCAGACTGCACA "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
## TCTGATACACGTGT "1"       "Gamma_delta_T_cells" "2.83772170020727e-10"
```

Step by step example

Next is to illustrate this pipeline step by step.

Load data and quality control

Here we will use the data set "pbmc" go through this pipeline. First is to load data set and do quality control. The input can be a Seurat object or count gene expression matrix.

```
library(scTypeGSEA)
pbmc <- scqc(pbmc_raw, min.cells = 3, min.features = 200, percent.mt = 5)
```

```
## The input is not a Seurat object, next to transform gene express count matrix to Seurat object.
```

```
pbmc
```

```
## An object of class Seurat
## 13714 features across 2643 samples within 1 assay
## Active assay: RNA (13714 features)
```

Data pre-process and do cluster

Next to do data pre-process and cluster. Here data pre-process includes normalization, feature selection and dimension reduction. Here we do "LogNormalize" normalization, "vst" feature selections to select 2000 features and do PCA to get the first 50 PCs. After data-process, we cluster the data.

```
set.seed(47)
pbmc <- doClustering(pbmc, normalization.method = "LogNormalize",
    scale.factor = 10000, selection.method = "vst",
    nfeatures = 2000, npcs = 50, dims = 1:50,
    k.param = 20, resolution = 0.5)
```

```
## The data has not been pre-processed, let's do it!
```

```
## Centering and scaling data matrix
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2643
## Number of edges: 201855
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8308
## Number of communities: 7
## Elapsed time: 0 seconds
```

```
head(pbmc@meta.data$seurat_clusters)
```

```
## [1] 0 3 0 4 2 0
## Levels: 0 1 2 3 4 5 6
```

If your data set have already been clustered, you can set "doit = TRUE" to do cluster using new parameters.

```
## don't run
pbmc <- check_cluster(pbmc, k.param = 30, resolution = 0.75, doit = TRUE)
```

Get fold changes

To get rank of gene list, we need to do gene differential expression analysis for each cluster with respect to all others. Here we use "wilcox" method.

```
cluster_list <- getFC(pbmc, min.pct = 0.25, test.use = "wilcox")
```

```
## Find marker genes for cluster 0
## Find marker genes for cluster 1
## Find marker genes for cluster 2
## Find marker genes for cluster 3
## Find marker genes for cluster 4
## Find marker genes for cluster 5
## Find marker genes for cluster 6
```

```
head(cluster_list[[1]])
```

```
##      IL7R      LDHB      CD3D      CD3E     NOSIP       LTB
## 1.2538187 1.1914868 1.1843738 1.0375362 0.9759263 0.9529190
```

Gene set enrichment analysis (GSEA)

After getting rank of gene list for each cluster, GSEA method will give us NSF. By NSF, we can decide the cell type for each cluster. Here we use "PanglaoDB" database.

```
cluster_celltype <- doGSEA(cluster_list = cluster_list, db = "PanglaoDB_list",
                                  minSize = 15, maxSize = 500)
```

```
## Do GSEA for cluster 0
## Do GSEA for cluster 1
## Do GSEA for cluster 2
## Do GSEA for cluster 3
## Do GSEA for cluster 4
## Do GSEA for cluster 5
## Do GSEA for cluster 6
```

```
head(cluster_celltype)
```

```
##          cell type         NES                padj
## Cluster0 "T_cells"         "2.56054175593529" "5.18028174072102e-06"
## Cluster1 "Neutrophils"     "2.29956534856229" "2.10866623817437e-06"
## Cluster2 "NK_cells"        "2.81922543534295" "9.96125097848532e-18"
## Cluster3 "B_cells_naive"   "2.87624750621789" "1.61417896374571e-13"
## Cluster4 "Dendritic_cells" "2.09699440161317" "5.56854459861125e-05"
## Cluster5 "Dendritic_cells" "2.58621936312799" "7.20051157046717e-10"
```

You can also use your own cell type database. Here "otherdb" can be a path to the new data base that hope to be used, which is list of cell types with their marker genes. The file must be 'rds' format.

```
## Don't run
cluster_celltype <- GSEA_analysis(cluster_list = cluster_list, otherdb = "path/to/your/rdsfile")
```

Label the cell type

The last step is to add label cell type to our Seurat object.

```
pbmc_res <- labelSeurat(pbmc, cluster_celltype)
pbmc <- pbmc_res$obj
head(pbmc@active.ident)
```

```
##  AAACATACAACCAC  AAACATTGAGCTAC  AAACATTGATCAGC  AAACCGTGCTTCCG  AAACCGTGTATGCG
##         T_cells   B_cells_naive         T_cells Dendritic_cells        NK_cells
##  AAACGCACTGGTAC
##         T_cells
## 6 Levels: T_cells Neutrophils NK_cells B_cells_naive ... Platelets
```

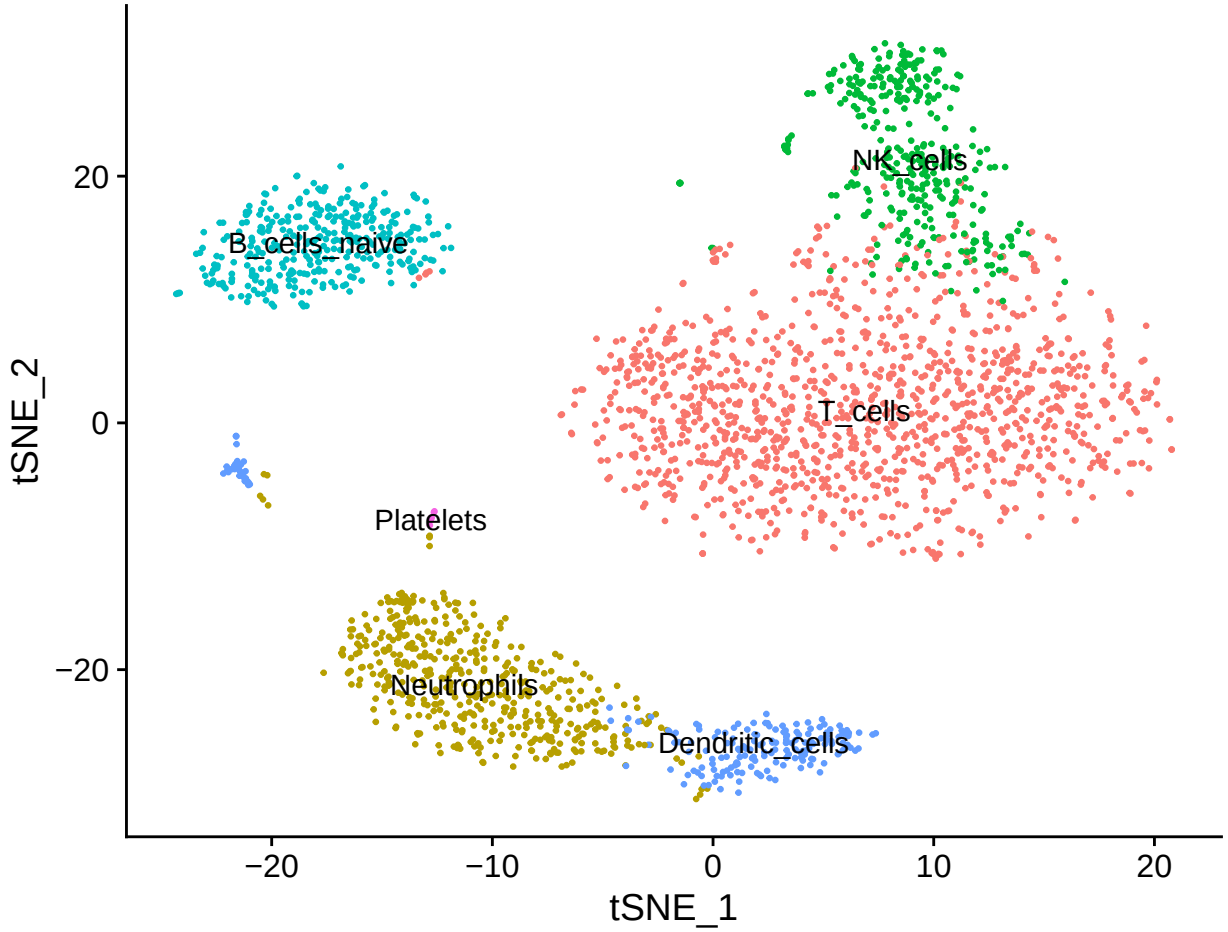
We can also check cell types for all cells.

```
head(pbmc_res$cell_mat)
```

```
##                ClusterID Cell Type         padj
## AAACATACAACCAC "1"       "T_cells"         "5.18028174072102e-06"
## AAACATTGAGCTAC "4"       "B_cells_naive"   "1.61417896374571e-13"
## AAACATTGATCAGC "1"       "T_cells"         "5.18028174072102e-06"
## AAACCGTGCTTCCG "5"       "Dendritic_cells" "5.56854459861125e-05"
## AAACCGTGTATGCG "3"       "NK_cells"        "9.96125097848532e-18"
## AAACGCACTGGTAC "1"       "T_cells"         "5.18028174072102e-06"
```

Plot the data with cell type

```
library(Seurat)
pbmc <- pbmc_res$obj
pbmc <- Seurat::RunTSNE(pbmc, dims = 1:30)
Seurat::DimPlot(pbmc, reduction = "tsne", label = TRUE, pt.size = 0.5) + NoLegend()
```



For ATAC sequence data

To deal with single cell ATAC sequence data, you may need following codes to install extra dependent packages.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("GenomeInfoDb")
BiocManager::install("ensembldb")
BiocManager::install("EnsDb.Hsapiens.v75")
BiocManager::install("GenomicRanges")

library(devtools)
install_github("timoast/signac")
```

To assign cell type for ATAC data, one can use following code to create gene acticity matrix and the go back to the above pipeline.

```
## don't run
dta_raw <- atac2rna(peaks = peaks, metadata = metadata, fragmentpath = fragment.path, qualitycontrol = 
```

Here "atac2rna" function is to create gene acticity matrix. This function provide two methods to acchieve this. The first one is based on "CreateGeneActivityMatrix" function in "Seurat" package, which is to take in a peak matrix and an annotation file (gtf), and collapse the peak matrix to a gene activity matrix. It makes the simplifying assumption that all counts in the gene body plus X kb up and or downstream should be attributed to that gene. This method is much faster than the second one. The second method is based on "FeatureMatrix" function in "Signac" package, which constructd a feature times cell matrix from a genomic fragments file. Using "?atac2rna" check details for parameters.