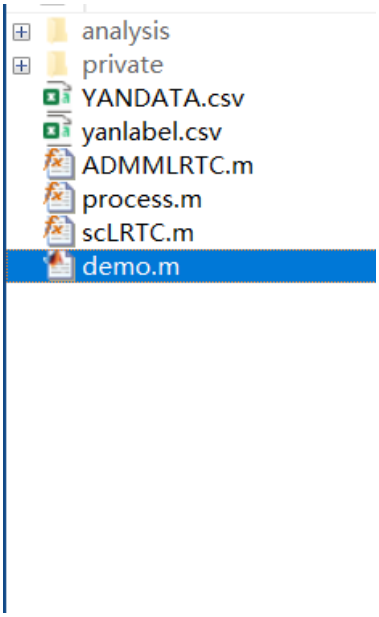
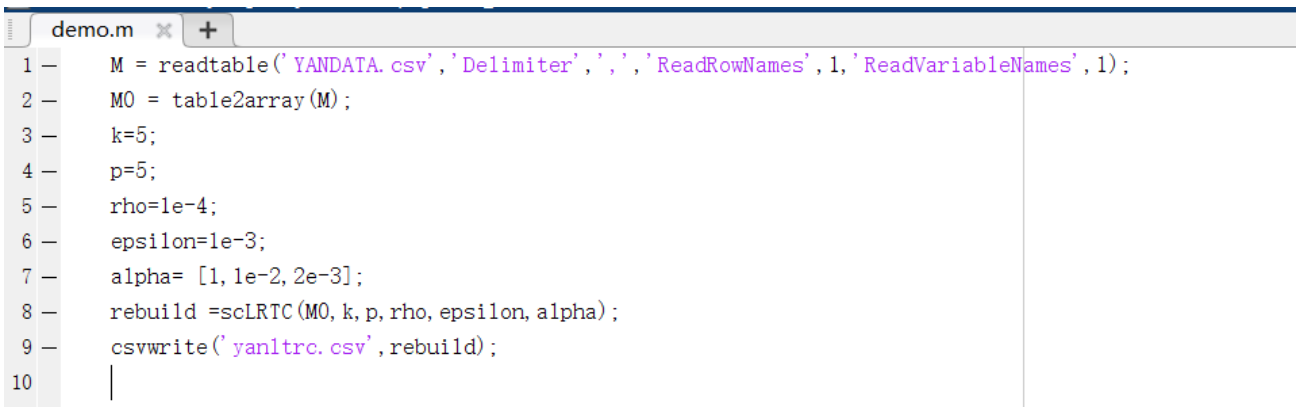
scLRTC detailed guide

Run demo

STEP 1. Download the source codes and unzip the MATLAB package. Change the current directory in MATLAB to the folder containing the scripts. Keep the current directory is scLRTC files directory .



STEP 2. Open demo.m and click green triangle arrow run button to get the result.



The program will run as shown below:

```
scLRTC: iterations = 40   difference=0.004080
scLRTC ends: total iterations = 48   difference=0.000991

scLRTC: iterations = 20   difference=0.033626
scLRTC: iterations = 40   difference=0.004859
scLRTC ends: total iterations = 49   difference=0.000629

scLRTC: iterations = 20   difference=0.033601
scLRTC: iterations = 40   difference=0.004866
scLRTC ends: total iterations = 49   difference=0.000628
```

When "complete!" appears, it means the program has finished running.

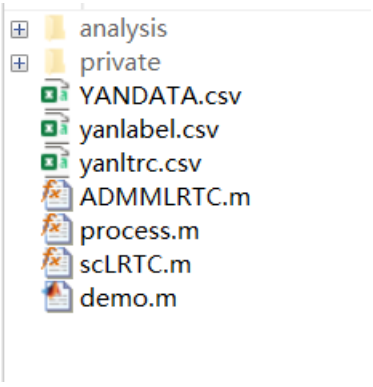
```
scLRTC: iterations = 20   difference=0.029099
scLRTC: iterations = 40   difference=0.004415
scLRTC ends: total iterations = 49   difference=0.000639

scLRTC: iterations = 20   difference=0.032028
scLRTC: iterations = 40   difference=0.002380
scLRTC ends: total iterations = 49   difference=0.000799

scLRTC: iterations = 20   difference=0.032006
scLRTC: iterations = 40   difference=0.002199
scLRTC ends: total iterations = 49   difference=0.000790

fx complete!>>
```

It will generate a csv file to save the data after imputation. In this demo, it called "yanltrc.csv". (Rows are genes and columns are cells)



We can perform the downstream analysis on the data of the yanlrtc.csv.
The downstream analysis needs two data. One is the gene expression matrix, the other is the cell label.

cluster analysis

SC3

For example, we used SC3 for cluster analysis. the path of the files, gene expression matrix file names, cell label file name should be modified (line 5,6,7). And the number of cluster (line 39,40) should adjust according to different data set size.

Then it will appear the SC3 clustering results and ARI after running the sc3.R file (line 40,41).
(You need install SC3, scater, mclust R package first, more detail see http://bioconductor.org/packages/release/bioc/vignettes/SC3/inst/doc/SC3.html#sc3-in-detail)

```
1  ## ----knitr-options, echo=FALSE, message=FALSE, warning=FALSE---------------
2  #rm(list = ls())
3  rm(list = ls())
4  gc()
5  setwd("D: /study/bioinformatics/impute/yan")
6  lpsdata<-read.table("YANlrtc.csv",header=T,row.names=1,sep=",",check.names=F)
7  class.label<- read.table("yanlabel.csv", header=T,sep=",",check.names=F)
8  class.label<-as.matrix(class.label)
9  class.label<- class.label[,2]
10 yan1<-lpsdata
11 label<-class.label
12
13 ## ----knitr-options, echo=FALSE, message=FALSE, warning=FALSE---------------
14 library(knitr)
15 opts_chunk$set(fig.align = 'center', fig.width = 6, fig.height = 5, dev = 'png')
16
17 ## ---- message=FALSE, warning=FALSE-----------------------------------------
18 library(SingleCellExperiment)
19 library(SC3)
20 library(scater)
21 library(mclust)
22 sce <- SingleCellExperiment(assays = list(counts = as.matrix(yan1),
23                                           logcounts = log2(as.matrix(yan1)+1)),
24                             colData = label)
25
26 # define feature names in feature_symbol column
27 rowData(sce)$feature_symbol <- rownames(sce)
28 # remove features with duplicated names
29 sce <- sce[!duplicated(rowData(sce)$feature_symbol), ]
30
31 # define spike-ins
32 #isSpike(sce, "ERCC") <- grepl("ERCC", rowData(sce)$feature_symbol)
33
34 ## --------------------------------------------------------------------------
35 #plotPCA(sce, colour_by = "cell_type1")
36
37 ## --------------------------------------------------------------------------
38
39 sce <- sc3(sce,gene_filter = FALSE, ks =9, biology = FALSE)
40 pre_label <-colData(sce)$sc3_9_clusters
41 ARI =adjustedRandIndex(pre_label, label)
42
```

TSNE+Kmeans

we used tsne+kmeans for cluster analysis. you should modify path (line 7) gene expression matrix file name (line 8), cell label file name (line 9) the result of ARI files names (line 23) cluster labels file names (line 24) as your path and file name. And the labelsum size (line 16), kmeans centers (line 19) should adjust according to different data set size.

Then it will appear the TSNE+kmeans clustering results and ARI after running the tesnkmeans.R file (line 40,41).
(You need install Rtsne, mclust R package first)

```
1
2  rm(list = ls())
3  gc()
4  library(devtools)
5  library(mclust)
6  library(Rtsne)
7  setwd("/Users/jianghuaijie/Desktop/study/bioinformatics/impute/uso")
8  lpsdata<-read.table("uso.csv",header=T,row.names=1,sep=",",check.names=F)
9  class.label<- read.table("usotruelabel.csv", header=T,sep=",",check.names=F)
10 class.label<-as.matrix(class.label)
11 class.label<-class.label[,2]
12 lpsdata =log2(lpsdata+1)
13 lpsdata=as.matrix(lpsdata)
14 arisum=array(0,dim=c(20,1))
15
16 labelsum <- matrix(1:12440,ncol=20)
17 for(i in 1:20){
18   jiangwei = Rtsne(t(as.matrix(lpsdata)),dim=2,10)$Y
19   temp = kmeans(t(as.matrix(lpsdata)), centers = 4)$cluster
20   arisum[i]=adjustedRandIndex(temp, class.label);
21   labelsum[,i]=temp
22 }
23 write.csv(arisum,'usorawari.csv')
24 write.csv(labelsum,'usorawlabel.csv')
```

Cell visualization

we used UMAP for Cell visualization you should modify path (line 6) gene expression matrix file name (line 7), cell label file name (line 8), the result of figure files names “umapraw” (line 41) as your path and file name. The Silhouette Coefficient is saved in “arawsc” (line 41). Then it will appear the figure of cell visualization and the Silhouette Coefficient after running the UMAP.R file (line 41).

(You need install scater, cluster, ggplot2 R package first)

```
1  rm(list = ls())
2  gc()
3  library("scater")
4  library ("cluster")
5  library("ggplot2")
6  setwd("C:/Users/jianghuaijie/Desktop/study/bioinformatics/impute/yan")
7  lpsdata0<-read.table("YANDATA.csv",header=T,row.names=1,sep=",",check.names=F)
8  truelabel=read.table("yanlabel.csv",header=T,row.names=1,sep=",",check.names=F)
9  #set.seed(12345)
10 calcukateSC <- function(lpsdata,label,name1){
11   label<-as.matrix(label)
12   labelx <-as.factor(label)
13   sce <- SingleCellExperiment(assays = list(counts = as.matrix(lpsdata),
14                                             logcounts = log2(as.matrix(lpsdata)+1)),
15                               colData = label)
16   rowData(sce)$feature_symbol <- rownames(sce)
17   sce <- sce[!duplicated(rowData(sce)$feature_symbol), ]
18   tsnered<-runUMAP(sce)
19   tsnepc=tsnered@int_colData@listData[["reducedDims"]]@listData[["UMAP"]]
20   tsnepc1 =data.frame(var_x=tsnepc[,1],var_y=tsnepc[,2])
21   p<-ggplot(data=tsnepc1, aes(x=var_x, y=var_y,color=labelx)) + geom_point(size=2)+
22     theme(plot.title = element_text(hjust = 0.5))
23   p<-p+scale_x_discrete("")+scale_y_discrete("")
24   p<-p+ theme_set(theme_bw())
25   p<-p+theme(panel.grid.major=element_line(colour=NA))
26   p<-p+labs(fill="")
27   p<-p+theme(legend.position="none")
28   dir =paste(name1,".png")
29   ggsave(dir, plot = p, device = NULL, path = NULL,
30          scale = 1, width = NA, height = NA, units =c("in", "cm", "mm"),
31          dpi = 600, limitsize = TRUE)
32   dis <- dist(tsnepc1)^2
33   #library(fpc)
34   label=as.matrix(label)
35   sil <- silhouette (label, dis)
36   sil=as.matrix(sil)
37   avg = mean(sil[,3])
38   return (avg)
39 }
40
41 arawsc=calcukateSC(lpsdata0,truelabel,"umapraw")
```

Generate simulation data set

We used splatter to Generate simulation data set, you should modify the path (line 4) and file names (line 38, 39) as your path and file name. If you want to generate 4 data sets in the paper, you should modify the parameters (line 29) as {-0.4, -0.35, -0.3, -0.25} respectively.

Then it will appear gene expression matrix files (line 38) and cell label files (line 39) after running the simulation.R file (line 41).

(You need install splatter scater ggplot2 R package first, more detail see https://bioconductor.org/packages/devel/bioc/vignettes/splatter/inst/doc/splatter.html)

```
1  ## ---- include = FALSE----------------------------------------------------
2  rm(list = ls())
3  gc()
4  setwd("/Users/mac/Downloads/")
5  knitr::opts_chunk$set(
6    collapse = TRUE,
7    comment = "#>"
8  )
9  ## ----setup---------------------------------------------------------------
10 library("splatter")
11 library("scater")
12 library("ggplot2")
13 # three groups
14 ## ----nGenes--------------------------------------------------------------
15 # Set the number of genes to 1000
16 params = newSplatParams()
17 params = setParams(params, list(batchCells = 500,
18                                 nGenes =1000,
19                                 group.prob = c(0.30, 0.3,0.4),
20                                 de.prob = c(0.05, 0.08, 0.01),
21                                 de.facLoc = 0.5,
22                                 de.facScale = 0.8)
23 )
24 # Set up the vector of dropout.mid
25 #dropout_mid = c(4, 5, 5.5)
26 # determine if it is a good parameter
27 # Generate the simulation data using Splatter package
28 sim = splatSimulateGroups(params,
29                           dropout.shape =c(-0.05,-0.05,-0.05),
30                           dropout.mid = c(0,0,0),
31                           dropout.type = "group",
32                           )
33 sim <- normalize(sim)
34 plotPCA(sim, colour_by = "Group")
35 X <- assays(sim)$count
36 X.log <- log10(X+ 1)
37 simlabel<-sim$Group
38 write.csv(X.log,file = "sim.csv",row.names = T)
39 write.csv(simlabel,file = "simlabel.csv",row.names = T)
40
```

Differentially expressed gene detection

We use seurat-encapsulated MAST for differential expression gene testing.you should modify path (line 4), gene expression matrix file names (line 5),cell label file names (line6) as your path and file name. The result of DE gene files and notDE gene files will save in the two csv files (line 19 20). you can modify the file name as your files names.

Then it will appear DE gene files (line 19) and notDE gene files (line 20) After running the DEMAST.R file.

(You need install seurat package first, more detail see https://satijalab.org/seurat/articles/de_vignette.html)

```
1
2  rm(list = ls())
3  gc()
4  setwd("C:/Users/jianghuaijie/Desktop/study/bioinformatics/impute/simdif")
5  lpsdata<-read.table("sim_full.csv",header=T,row.names=1,sep=",",check.names=F)
6  class.label<- read.table("sim_label.csv", header=T,sep=",",check.names=F)
7  class.label<-as.matrix(class.label)
8  label<- class.label[,2]
9  library(Seurat)
10 pbmc <-CreateSeuratObject(counts = lpsdata,project = "simdata",min.cells = 3)
11
12 Idents(pbmc)<-label
13 gene1 <- FindMarkers(pbmc, ident.1 = "Group2", ident.2 = "Group3",test.use = "MAST")
14 sigDE <- rownames(gene1)[gene1['p_val_adj'] < 0.01]
15
16 genename<-rownames(lpsdata)
17 notDE<-setdiff(genename,sigDE1)
18
19 write.csv(sigDE,"fullsiggene23.csv")
20 write.csv(notDE,"fullnotsiggene23.csv")
```

cell trajectory inference

In the TSCAN.R files, there two functions we defined. One is used to calculate KRCS and POS score, the other is used to draw the figure of cell trajectory inference.

The input of two function is the gene expression matrix and the cell label.

(You need install TSCAN ggplot2 R package first, more detail see http://www.bioconductor.org/packages/release/bioc/vignettes/TSCAN/inst/doc/TSCAN.pdf)

```
library(TSCAN)
my.TSCAN =  function(count, cellLabels){
  colnames(count) = c(1:ncol(count))
  procdata <- TSCAN::preprocess(count)
  lpsmclust <- TSCAN::exprmclust(procdata)

  lpsorder <- TSCAN::TSCANorder(lpsmclust, orderonly=F)

  Pseudotime = lpsorder$Pseudotime[match(colnames(count),lpsorder$sample_name)]
  cor.kendall = cor(Pseudotime, as.numeric(cellLabels), method = "kendall", use = "complete.obs")
  subpopulation <- data.frame(cell = colnames(count), sub = as.numeric(cellLabels)-1)
  POS <- orderscore(subpopulation, lpsorder)[1]
  out = list(cor.kendall=abs(cor.kendall), POS=abs(POS))
  out
  #Pseudotime
}
plotmclust2 <- function (mclustobj,cellLabels, x = 1, y = 2, MSTorder = NULL, show_tree = T,
                         show_cell_names = F, cell_name_size = 3, markerexpr = NULL)
{
  color_by = 'cellLabels' # color_by = "State"
  lib_info_with_pseudo <- data.frame(State = mclustobj$clusterid,
                                     sample_name = names(mclustobj$clusterid),
                                     cellLabels = cellLabels)
  lib_info_with_pseudo$State <- factor(lib_info_with_pseudo$State)
  S_matrix <- mclustobj$pcareduceres
  pca_space_df <- data.frame(S_matrix[, c(x, y)])
  colnames(pca_space_df) <- c("pca_dim_1", "pca_dim_2")
  pca_space_df$sample_name <- row.names(pca_space_df)
  edge_df <- merge(pca_space_df, lib_info_with_pseudo, by.x = "sample_name",
                   by.y = "sample_name")
  edge_df$markerexpr <- markerexpr[edge_df$sample_name]
  if (!is.null(markerexpr)) {
    g <- ggplot(data = edge_df, aes(x = pca_dim_1, y = pca_dim_2,
                                    size = markerexpr))
    g <- g + geom_point(aes_string(color = color_by), na.rm = TRUE)
  }
  else {
    g <- ggplot(data = edge_df, aes(x = pca_dim_1, y = pca_dim_2))
    g <- g + geom_point(aes_string(color = color_by), na.rm = TRUE,
                        size = 3)
```