Protocols for performing FOACC analysis of six scRNA-seq data sets

The protocol for human peripheral blood mononuclear cells (pbmc3k)

The protocol for *Caenorhabditis elegans* embryonic cells

The protocol for *Arabidopsis thaliana* roots

The protocol for *Arabidopsis thaliana* leaves

The protocol for rice roots

The protocol for maize leaves

Required software/package:

Required R packages:

```
(1) install BiocManager
install.packages("BiocManager")
# --- Please select a CRAN mirror for use in this session ---
# Secure CRAN mirrors
#    1: 0-Cloud [https]
# Selection: 1
(2) install Seurat
library("BiocManager")
BiocManager::install("Seurat")
(3) install SC3
BiocManager::install("SC3")
(4) install DIMMSC
BiocManager::install("devtools")
library("devtools")
install.packages("http://s3-us-west-2.amazonaws.com/10x.files/code/cellrangerRkit-1.1.0.tar.gz", repos=NULL)
# during intallation of 'cellrangerRkit-1.1.0.tar.gz', some errors may be presented, such as:
# ERROR: dependencies ‘bit64’, ‘Rmisc’, ‘rhdf5’ are not available for package ‘cellrangerRkit’
# To solve these error, install these related packages:
# BiocManager::install("bit64")
# BiocManager::install("Rmisc")
# BiocManager::install("rhdf5")
install_github("wt2015-github/DIMMSC")
(5) install dplyr
```

```
BiocManager::install("dplyr")
(6) install patchwork
BiocManager::install("patchwork")
(7) install SingleCellExperiment
BiocManager::install("SingleCellExperiment")
(8) install scater
BiocManager::install("scater")
(9) install Matrix
BiocManager::install("Matrix")
(10) install SeuratData
devtools::install_github('satijalab/seurat-data')
(11) install scales
BiocManager::install("scales")
(12) install phyclust
BiocManager::install("phyclust")
```

Required Perl scripts:

```
calculate_adjusted_rand_index_score.pl
calculate_normalized_mutual_information_score.pl
calculate_purity_score.pl
cell_cluster_change.pl
change_10x_chromium_lib_sequenced_data_format.pl
comparison_Seurat-SC3-DIMMSC.pl
comparison_Seurat-SC3-DIMMSC.type.select.step01.pl
comparison_Seurat-SC3-DIMMSC.type.select.step02.pl
generate_matrix.pl
generate_SC3-DIMMSC_Rscript.pl
ID_from_cell_list.pl
reformat_matrix.pl
select_cells_by_cellID.pl
transform_matrix.pl
```

Required perl modules:

```
(1) Data::Dumper
(2) FindBin
(3) Getopt::Long
(4) Math::Complex
```

The protocol for human peripheral blood mononuclear cells (pbmc3k; details in

"examples")

1. Obtaining data set

```
wget https://s3-us-west-2.amazonaws.com/10x.files/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz
```

2. Running Seurat analysis. The R scripts was obtained from Seurat web links (https://satijalab.org/seurat/articles/pbmc3k_tutorial.html).

```
library(Seurat)
library(dplyr)
library(patchwork)
library(scales)

pbmc.data <- Read10X(data.dir = ".")
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
pdf("pbmc.01.VlnPlot.pdf", height=3.5, width=8)
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
dev.off()

pdf("pbmc.01.FeatureScatter.pdf", height=3.5, width=8)
plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
dev.off()

pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)

# scale.factor of “10000” is inferred from “nCount_RNA” in “pbmc.01.VlnPlot.pdf”
pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize", scale.factor = 10000)
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(pbmc), 10)

pdf("pbmc.02.VariableFeaturePlot.pdf", height=3, width=10)
plot1 <- VariableFeaturePlot(pbmc)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
plot1 + plot2
dev.off()
```

```
all.genes <- rownames(pbmc)
pbmc <- ScaleData(pbmc, features = all.genes)
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))

pdf("pbmc.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
dev.off()

pdf("pbmc.04.DimPlot.pdf", height=6, width=6)
DimPlot(pbmc, reduction = "pca")
dev.off()

pdf("pbmc.05.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(pbmc, dims = 1:15, cells = 500, balanced = TRUE)
dev.off()

pbmc <- JackStraw(pbmc, num.replicate = 100)
pbmc <- ScoreJackStraw(pbmc, dims = 1:20)

pdf("pbmc.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(pbmc, dims = 1:15)
dev.off()

pdf("pbmc.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(pbmc)
dev.off()

# for “dims = 1:10”, the “10” is inferred from “pbmc.05.DimHeatmap.pdf”, “pbmc.06.JackStrawPlot.pdf”, and “pbmc.07.ElbowPlot.pdf” as shown in Seurat
pbmc <- FindNeighbors(pbmc, dims = 1:10)
pbmc <- FindClusters(pbmc, resolution = 0.5)

pbmc.UMAP <- RunUMAP(pbmc, dims = 1:10)
pbmc.tSNE <- RunTSNE(pbmc, dims = 1:10)

pdf("pbmc.08.DimPlot.UMAP.pdf", height=4, width=5)
DimPlot(pbmc.UMAP, reduction = "umap")
dev.off()

pdf("pbmc.09.DimPlot.tSNE.pdf", height=4, width=5)
DimPlot(pbmc.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(9) # 9, means 9 clusters detected by tSNE analysis
```

```
pdf("pbmc.step01.09.DimPlot.tSNE.select.pdf", height=3.2, width=4)
DimPlot(pbmc.tSNE, reduction = "tsne", cols=color_list)
dev.off()

write.table(WhichCells(pbmc.UMAP, idents="0"), "pbmc_UMAP_cell.0.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="1"), "pbmc_UMAP_cell.1.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="2"), "pbmc_UMAP_cell.2.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="3"), "pbmc_UMAP_cell.3.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="4"), "pbmc_UMAP_cell.4.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="5"), "pbmc_UMAP_cell.5.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="6"), "pbmc_UMAP_cell.6.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="7"), "pbmc_UMAP_cell.7.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="8"), "pbmc_UMAP_cell.8.txt", sep="\t", quote = FALSE)

pbmc.markers <- FindAllMarkers(pbmc.UMAP, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
pbmc.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_log2FC)

pdf("pbmc.10.VlnPlot.pdf", height=4, width=8)
VlnPlot(pbmc.tSNE, features = c("NKG7", "PF4"), slot = "counts", log = TRUE)
dev.off()

pdf("pbmc.11.FeaturePlot.pdf", height=7, width=8)
FeaturePlot(pbmc.tSNE, features = c("MS4A1", "GNLY", "CD3E", "CD14", "FCER1A", "FCGR3A", "LYZ",
"PPBP", "CD8A"))
dev.off()

# MS4A1 <-> ENSG00000156738, GNLY <-> ENSG00000115523, CD3E <-> ENSG00000198851
# CD14 <-> ENSG00000170458, FCER1A <-> ENSG00000179639, FCGR3A <-> ENSG00000203747
# LYZ <-> ENSG00000090382, PPBP <-> ENSG00000163736, CD8A <-> ENSG00000153563
# FeaturePlot(pbmc.tSNE, features = c("ENSG00000156738", "ENSG00000115523", "ENSG00000198851",
"ENSG00000170458", "ENSG00000179639", "ENSG00000203747", "ENSG00000090382",
"ENSG00000163736", "ENSG00000153563"))

top10 <- pbmc.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)
pdf("pbmc.12.DoHeatmap.pdf", height=8, width=8)
DoHeatmap(pbmc.UMAP, features = top10$gene) + NoLegend()
dev.off()

# new.cluster.ids <- c("Naive CD4 T", "CD14+ Mono", "Memory CD4 T", "B", "CD8 T", "FCGR3A+ Mono",
"NK", "DC", "Platelet")
# names(new.cluster.ids) <- levels(pbmc.UMAP)
# pbmc <- RenameIdents(pbmc.UMAP, new.cluster.ids)
# DimPlot(pbmc.UMAP, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()
```

```
saveRDS(pbmc, file = "pbmc3k_step01.rds")
```

3. Preparing data for running SC3 and DIMM-SC

```
#The content of "pbmc_UMAP_cell.list" file:
A01  /SeuratV4/ pbmc_UMAP_cell.0.txt
A02  /SeuratV4/ pbmc_UMAP_cell.1.txt
A03  /SeuratV4/ pbmc_UMAP_cell.2.txt
A04  /SeuratV4/ pbmc_UMAP_cell.3.txt
A05  /SeuratV4/ pbmc_UMAP_cell.4.txt
A06  /SeuratV4/ pbmc_UMAP_cell.5.txt
A07  /SeuratV4/ pbmc_UMAP_cell.6.txt
A08  /SeuratV4/ pbmc_UMAP_cell.7.txt
A09  /SeuratV4/ pbmc_UMAP_cell.8.txt

perl reformat_matrix.pl -matrix matrix.mtx -features genes.tsv -barcodes barcodes.tsv -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl ID_from_cell_list.pl -cell_list ../pbmc_UMAP_cell.list -output cell.id

perl select_cells_by_cellID.pl -reformat_byCellID reformat_MG200_MC3_byCellID.xls.gz -cell_ID_list cell.id -output cell. txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features genes.tsv step02

gzip step02_features.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byGeneID.xls -cluster 9 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The content of R scripts (run_SC3-DIMMSC.R) is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byGeneID.xls", header = TRUE, sep = "\t", quote = "", row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=9, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K9_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:9, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_9_clusters, "SC3_K9_cell_cluster.txt", quote=FALSE, sep="\t")
```

Note: running the script by command: “nohup R CMD BATCH run_SC3-DIMMSC.R &”.

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../pbmc_UMAP_cell.list -SC3_cell_cluster SC3_K9_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K9_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt

perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat-SC3-DIMMSC.select.cell.id

ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat-SC3-DIMMSC.select.cell.id step02_cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list step02_cell.id -output step02_cell.txt
```

```
perl generate_matrix.pl -reformat_matrix_byCellID step02_cell.txt -features ../step01/comparison_Seurat-SC3-DIMMSC/step02_features.tsv.gz -output_prefix step03

gzip step03_genes.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv

# in new folder step02:
ln -s ../step03_genes.tsv.gz features.tsv.gz
ln -s ../step03_matrix.mtx.gz matrix.mtx.gz
ln -s ../step03_barcodes.tsv barcodes.tsv
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(patchwork)
library(scales)

pbmc.data <- Read10X(data.dir = ".")
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
pdf("pbmc.step02.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
dev.off()

pdf("pbmc.step02.01.FeatureScatter.pdf", height=3.5, width=8)
plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
dev.off()

pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2000 & percent.mt < 5)
pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize", scale.factor = 8000)
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(pbmc), 10)

pdf("pbmc.step02.02.VariableFeaturePlot.pdf", height=3, width=10)
plot1 <- VariableFeaturePlot(pbmc)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
plot1 + plot2
dev.off()
```

```
all.genes <- rownames(pbmc)
pbmc <- ScaleData(pbmc, features = all.genes)
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))

pdf("pbmc.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
dev.off()

pdf("pbmc.step02.04.DimPlot.pdf", height=6, width=6)
DimPlot(pbmc, reduction = "pca")
dev.off()

pdf("pbmc.step02.05.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(pbmc, dims = 1:15, cells = 500, balanced = TRUE)
dev.off()

pbmc <- JackStraw(pbmc, num.replicate = 100)
pbmc <- ScoreJackStraw(pbmc, dims = 1:20)

pdf("pbmc.step02.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(pbmc, dims = 1:15)
dev.off()

pdf("pbmc.step02.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(pbmc)
dev.off()

pbmc <- FindNeighbors(pbmc, dims = 1:9)
pbmc <- FindClusters(pbmc, resolution = 0.5)

pbmc.UMAP <- RunUMAP(pbmc, dims = 1:9)
pbmc.tSNE <- RunTSNE(pbmc, dims = 1:9)

pdf("pbmc.step02.08.DimPlot.UMAP.pdf", height=4, width=5)
DimPlot(pbmc.UMAP, reduction = "umap")
dev.off()

pdf("pbmc.step02.09.DimPlot.tSNE.pdf", height=4, width=5)
DimPlot(pbmc.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(9)
new_identity_TSNE<-pbmc.tSNE
```

```
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="0", "1"="2", "2"="3", "3"="1", "4"="5", "5"="6", "6"="4", "7"="7", "8"="8")
pdf("pbmc.step02.09.DimPlot.tSNE.select.pdf", height=3.2, width=4)
DimPlot(new_identity_TSNE, reduction = "tsne", cols=c(color_list[1], color_list[3], color_list[4], color_list[2], color_list[6], color_list[7], color_list[5], color_list[8], color_list[9]))
dev.off()
#            Step01  Step02  Step01     Step02
#      405 A01       A01     A01  A01
#      337 A03       A02     A03  A02
#      325 A04       A03     A04  A03
#      270 A02       A04     A02  A04
#      160 A06       A05     A06  A05
#      135 A07       A06     A07  A06
#      109 A05       A07     A05  A07
#       21 A08       A08     A08  A08
#       11 A09       A09     A09  A09

write.table(WhichCells(pbmc.UMAP, idents="0"), "pbmc_UMAP_cell.0.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="1"), "pbmc_UMAP_cell.1.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="2"), "pbmc_UMAP_cell.2.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="3"), "pbmc_UMAP_cell.3.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="4"), "pbmc_UMAP_cell.4.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="5"), "pbmc_UMAP_cell.5.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="6"), "pbmc_UMAP_cell.6.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="7"), "pbmc_UMAP_cell.7.txt", sep="\t", quote = FALSE)
write.table(WhichCells(pbmc.UMAP, idents="8"), "pbmc_UMAP_cell.8.txt", sep="\t", quote = FALSE)

pbmc.markers <- FindAllMarkers(pbmc.UMAP, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
pbmc.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_log2FC)

#pdf("pbmc.step02.10.VlnPlot.pdf", height=4, width=8)
#VlnPlot(pbmc.UMAP, features = c("NKG7", "PF4"), slot = "counts", log = TRUE)
#dev.off()

# NKG7 <-> ENSG00000105374
# PF4 <-> ENSG00000163737

pdf("pbmc.step02.10.VlnPlot.pdf", height=4, width=8)
VlnPlot(pbmc.tSNE, features = c("ENSG00000105374", "ENSG00000163737"), slot = "counts", log = TRUE)
dev.off()

#pdf("pbmc.step02.11.FeaturePlot.pdf", height=7, width=8)
#FeaturePlot(pbmc.tSNE, features = c("MS4A1", "GNLY", "CD3E", "CD14", "FCER1A", "FCGR3A", "LYZ", "PPBP", "CD8A"))
```

```
#dev.off()

# MS4A1 <-> ENSG00000156738, GNLY <-> ENSG00000115523, CD3E <-> ENSG00000198851
# CD14 <-> ENSG00000170458, FCER1A <-> ENSG00000179639, FCGR3A <-> ENSG00000203747
# LYZ <-> ENSG00000090382, PPBP <-> ENSG00000163736, CD8A <-> ENSG00000153563
pdf("pbmc.step02.11.FeaturePlot.pdf", height=7, width=8)
FeaturePlot(pbmc.tSNE, features = c("ENSG00000156738", "ENSG00000115523", "ENSG00000198851",
"ENSG00000170458", "ENSG00000179639", "ENSG00000203747", "ENSG00000090382",
"ENSG00000163736", "ENSG00000153563"))
dev.off()

top10 <- pbmc.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)
pdf("pbmc.step02.12.DoHeatmap.pdf", height=8, width=8)
DoHeatmap(pbmc.UMAP, features = top10$gene) + NoLegend()
dev.off()

# new.cluster.ids <- c("Naive CD4 T", "CD14+ Mono", "Memory CD4 T", "B", "CD8 T", "FCGR3A+ Mono",
"NK", "DC", "Platelet")
# names(new.cluster.ids) <- levels(pbmc.UMAP)
# pbmc <- RenameIdents(pbmc.UMAP, new.cluster.ids)
# DimPlot(pbmc.UMAP, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()

saveRDS(pbmc, file = "pbmc3k_step02.rds")
```

The protocol for *Caenorhabditis elegans* embryonic cells

1. Obtaining data set by running an R script:

```
library("devtools")
library("Seurat")
library("SeuratData")

devtools::install_github('satijalab/seurat-data')

InstallData("celegans.embryo")
# http://seurat.nygenome.org/src/contrib/celegans.embryo.SeuratData_0.1.0.tar.gz

write.table(GetAssayData(celegans.embryo), "Celegans.embryo.exp.byGene.txt", sep="\t", quote = FALSE)
```

Performing analysis using Bash and Perl scripts:

```
# contents in Celegans.embryo.exp.byGene.txt
```

```
# AAACCTGCAAGACGTG.300.1.1     AAACCTGGTGTGAATA.300.1.1         ...
# WBGene00010957    5       10     ...

# vi Celegans.embryo.exp.byGene.txt, by adding 'CellID'
# CellID     AAACCTGCAAGACGTG.300.1.1        AAACCTGGTGTGAATA.300.1.1         ...
# WBGene00010957    5       10     ...

perl transform_matrix.pl -matrix Celegans.embryo.exp.byGene.txt -output Celegans.embryo.exp.byCell.txt

less -S Celegans.embryo.exp.byGene.txt|awk '{print $1}'|sort -u | perl -ne 'if(!($_=~/CellID/)){print $_}'> gene.id

perl ../generate_matrix.pl -reformat_matrix_byCellID Celegans.embryo.exp.byCell.txt -features gene.id -output_prefix Celegans.embryo.data

gzip Celegans.embryo.data_features.tsv
gzip Celegans.embryo.data_matrix.mtx
gzip Celegans.embryo.data_barcodes.tsv
gzip Celegans.embryo.exp.byGene.txt
gzip Celegans.embryo.exp.byCell.txt

mkdir step01
cd step01
ln -s ../Celegans.embryo.data_features.tsv.gz features.tsv.gz
ln -s ../Celegans.embryo.data_matrix.mtx.gz matrix.mtx.gz
ln -s ../Celegans.embryo.data_barcodes.tsv.gz barcodes.tsv.gz
```

2. Running Seurat analysis. The content of R script was:

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

Cel.exp.filter <- Read10X(".")
Cel.exp.filter <- CreateSeuratObject(Cel.exp.filter, min.cells=3, min.features=200, project="10X_Cel")

pdf("Cel.step01.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(Cel.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

Cel.exp.filter <- subset(Cel.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 2300)
```

```
pdf("Cel.step01.01.VlnPlot.filter.pdf", height=6, width=8)
VlnPlot(Cel.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

Cel.exp.filter <- NormalizeData(Cel.exp.filter, normalization.method = "LogNormalize", scale.factor = 6000)
Cel.exp.filter <- FindVariableFeatures(Cel.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(Cel.exp.filter), 10)
pdf("Cel.step01.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(Cel.exp.filter) + LabelPoints(plot = VariableFeaturePlot(Cel.exp.filter), points = top10, repel
= TRUE)
dev.off()

Cel.all.genes <- rownames(Cel.exp.filter)
Cel.exp.filter <- ScaleData(Cel.exp.filter, features = Cel.all.genes)

Cel.exp.filter <- RunPCA(Cel.exp.filter, npcs = 50, features = VariableFeatures(object = Cel.exp.filter))

pdf("Cel.step01.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(Cel.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("Cel.step01.04.DimPlot.pdf", height=6, width=6)
DimPlot(Cel.exp.filter, reduction = "pca")
dev.off()

pdf("Cel.step01.05.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(Cel.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

Cel.exp.filter <- JackStraw(Cel.exp.filter, num.replicate = 100, dims = 50)
Cel.exp.filter <- ScoreJackStraw(Cel.exp.filter, dims = 1:30)

pdf("Cel.step01.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(Cel.exp.filter, dims = 1:30)
dev.off()

pdf("Cel.step01.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(Cel.exp.filter, ndims = 30)
dev.off()

Cel.exp.filter <- FindNeighbors(Cel.exp.filter, dims = 1:30)

Cel.exp.filter <- FindClusters(Cel.exp.filter, resolution = 0.5)
```

```
Cel.exp.filter.UMAP <- RunUMAP(Cel.exp.filter, dims = 1:30)
Cel.exp.filter.tSNE <- RunTSNE(Cel.exp.filter, dims = 1:30)

pdf("Cel.step01.08.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(Cel.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("Cel.step01.09.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(Cel.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(21)
pdf("Cel.step01.09.DimPlot.tSNE.select.pdf", height=5, width=6.5)
DimPlot(Cel.exp.filter.tSNE, reduction = "tsne", cols=color_list)
dev.off()

saveRDS(Cel.exp.filter, file = "Cel-Seurat4_step01.rds")

Cel.exp.filter.tSNE.markers <- FindAllMarkers(Cel.exp.filter.tSNE, logfc.threshold = 0.25, test.use = "roc", only.pos = TRUE)
Cel.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- Cel.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("Cel.step01.10.DoHeatmap.pdf", height=13, width=8)
DoHeatmap(Cel.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

write.table(WhichCells(Cel.exp.filter.tSNE, idents="0"), "Cel.step01.Cell-00.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="1"), "Cel.step01.Cell-01.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="2"), "Cel.step01.Cell-02.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="3"), "Cel.step01.Cell-03.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="4"), "Cel.step01.Cell-04.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="5"), "Cel.step01.Cell-05.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="6"), "Cel.step01.Cell-06.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="7"), "Cel.step01.Cell-07.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="8"), "Cel.step01.Cell-08.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="9"), "Cel.step01.Cell-09.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="10"), "Cel.step01.Cell-10.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="11"), "Cel.step01.Cell-11.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="12"), "Cel.step01.Cell-12.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="13"), "Cel.step01.Cell-13.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="14"), "Cel.step01.Cell-14.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="15"), "Cel.step01.Cell-15.ID.txt", sep="\t", quote = FALSE)
```

```
write.table(WhichCells(Cel.exp.filter.tSNE, idents="16"), "Cel.step01.Cell-16.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="17"), "Cel.step01.Cell-17.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="18"), "Cel.step01.Cell-18.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="19"), "Cel.step01.Cell-19.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="20"), "Cel.step01.Cell-20.ID.txt", sep="\t", quote = FALSE)
```

3. Preparing data for running SC3 and DIMM-SC

```
# The content of 'Cel.step01.Cell.list' file:
A01  /Cel.FOACC/step01/Cel.step01.Cell-00.ID.txt
A02  /Cel.FOACC/step01/Cel.step01.Cell-01.ID.txt
A03  /Cel.FOACC/step01/Cel.step01.Cell-02.ID.txt
A04  /Cel.FOACC/step01/Cel.step01.Cell-03.ID.txt
A05  /Cel.FOACC/step01/Cel.step01.Cell-04.ID.txt
A06  /Cel.FOACC/step01/Cel.step01.Cell-05.ID.txt
A07  /Cel.FOACC/step01/Cel.step01.Cell-06.ID.txt
A08  /Cel.FOACC/step01/Cel.step01.Cell-07.ID.txt
A09  /Cel.FOACC/step01/Cel.step01.Cell-08.ID.txt
A10  /Cel.FOACC/step01/Cel.step01.Cell-09.ID.txt
A11  /Cel.FOACC/step01/Cel.step01.Cell-10.ID.txt
A12  /Cel.FOACC/step01/Cel.step01.Cell-11.ID.txt
A13  /Cel.FOACC/step01/Cel.step01.Cell-12.ID.txt
A14  /Cel.FOACC/step01/Cel.step01.Cell-13.ID.txt
A15  /Cel.FOACC/step01/Cel.step01.Cell-14.ID.txt
A16  /Cel.FOACC/step01/Cel.step01.Cell-15.ID.txt
A17  /Cel.FOACC/step01/Cel.step01.Cell-16.ID.txt
A18  /Cel.FOACC/step01/Cel.step01.Cell-17.ID.txt
A19  /Cel.FOACC/step01/Cel.step01.Cell-18.ID.txt
A20  /Cel.FOACC/step01/Cel.step01.Cell-19.ID.txt
A21  /Cel.FOACC/step01/Cel.step01.Cell-20.ID.txt

perl reformat_matrix.pl -matrix matrix.mtx.gz -features features.tsv.gz -barcodes barcodes.tsv.gz -
min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl ID_from_cell_list.pl ../Cel.step01.Cell.list -output cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../reformat_MG200_MC3_byCellID.xls.gz -cell_ID_list
```

```
cell.id -output cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features ../features.tsv.gz -output_prefix step02

gzip step02_features.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byCellID.xls -cluster 21 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The R scripts is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byCellID.xls", header = TRUE, sep = "\t", quote = "", row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=21, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K21_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:21, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_21_clusters, "SC3_K21_cell_cluster.txt", quote=FALSE, sep="\t")
```

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../Cel.step01.Cell.list -SC3_cell_cluster SC3_K21_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K21_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt
```

```
perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat_SC3_DIMMSC.type.select.cell.id

cd ../../step02
ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat_SC3_DIMMSC.type.select.cell.id select_cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list select_cell.id -output select_cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID select_cell.txt -features ../step01/comparison_Seurat-SC3-DIMMSC/step02_features.tsv.gz -output_prefix step03

gzip step03_features.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv
gzip select_cell.txt

in new folder:
ln -s step03_genes.tsv.gz features.tsv.gz
ln -s step03_matrix.mtx.gz matrix.mtx.gz
ln -s step03_barcodes.tsv barcodes.tsv
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

Cel.exp.filter <- Read10X(".")
Cel.exp.filter <- CreateSeuratObject(Cel.exp.filter, min.cells=3, min.features=200, project="10X_Cel")

pdf("Cel.step02.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(Cel.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

Cel.exp.filter <- subset(Cel.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 2300)

pdf("Cel.step02.01.VlnPlot.filter.pdf", height=6, width=8)
```

```
VlnPlot(Cel.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

Cel.exp.filter <- NormalizeData(Cel.exp.filter, normalization.method = "LogNormalize", scale.factor = 6000)
Cel.exp.filter <- FindVariableFeatures(Cel.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(Cel.exp.filter), 10)
pdf("Cel.step02.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(Cel.exp.filter) + LabelPoints(plot = VariableFeaturePlot(Cel.exp.filter), points = top10, repel
= TRUE)
dev.off()

Cel.all.genes <- rownames(Cel.exp.filter)
Cel.exp.filter <- ScaleData(Cel.exp.filter, features = Cel.all.genes)

Cel.exp.filter <- RunPCA(Cel.exp.filter, npcs = 50, features = VariableFeatures(object = Cel.exp.filter))

pdf("Cel.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(Cel.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("Cel.step02.04.DimPlot.pdf", height=6, width=6)
DimPlot(Cel.exp.filter, reduction = "pca")
dev.off()

pdf("Cel.step02.05.DimHeatmap.pdf", height=16, width=10)
DimHeatmap(Cel.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

Cel.exp.filter <- JackStraw(Cel.exp.filter, num.replicate = 100, dims = 50)
Cel.exp.filter <- ScoreJackStraw(Cel.exp.filter, dims = 1:30)

pdf("Cel.step02.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(Cel.exp.filter, dims = 1:30)
dev.off()

pdf("Cel.step02.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(Cel.exp.filter, ndims = 30)
dev.off()

Cel.exp.filter <- FindNeighbors(Cel.exp.filter, dims = 1:20)

Cel.exp.filter <- FindClusters(Cel.exp.filter, resolution = 0.5)
```

```
Cel.exp.filter.UMAP <- RunUMAP(Cel.exp.filter, dims = 1:20)
Cel.exp.filter.tSNE <- RunTSNE(Cel.exp.filter, dims = 1:20)

pdf("Cel.step02.08.DimPlot.UMAP.pdf", height=5, width=6)
DimPlot(Cel.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("Cel.step02.09.DimPlot.tSNE.pdf", height=5, width=6)
DimPlot(Cel.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(21) # 21, suggested by cluster numbers in step01
new_identity_TSNE<-Cel.exp.filter.tSNE
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="10+19", "1"="4", "2"="0", "3"="11", "4"="2",
"5"="8", "6"="17", "7"="14+15", "8"="13", "9"="1", "10"="18", "11"="6", "12"="9", "13"="5", "14"="7",
"15"="3", "16"="16", "17"="12", "18"="20")
pdf("Cel.step02.09.DimPlot.tSNE.select.pdf", height=5, width=6)
DimPlot(new_identity_TSNE, reduction = "tsne", cols=c(color_list[11], color_list[5], color_list[1], color_list[12],
color_list[3], color_list[9], color_list[18], color_list[15], color_list[14], color_list[2], color_list[19], color_list[7],
color_list[10], color_list[6], color_list[8], color_list[4], color_list[17], color_list[13], color_list[21]))
dev.off()
#       1 OriCluster       NewCluster OriCluster  NewCluster
#      24 A21      B19     A21  B19
#      48 A13      B18     A13  B18
#      48 A17      B17     A17  B17
#      56 A04      B16     A04  B16
#      56 A08      B15     A08  B15
#      58 A06      B14     A06  B14
#      58 A10      B13     A10  B13
#      62 A07      B12     A07  B12
#      63 A19      B11     A19  B11
#      66 A02      B10     A02  B10
#      69 A14      B09     A14  B09
#      47 A15      B08     A15  B08
#      25 A16      B08     A16  B08
#      76 A18      B07     A18  B07
#      87 A09      B06     A09  B06
#      92 A03      B05     A03  B05
#     111 A12      B04     A12  B04
#     123 A01      B03     A01  B03
#     124 A05      B02     A05  B02
#      94 A11      B01     A11  B01
#      41 A20      B01     A20  B01
```

```
saveRDS(Cel.exp.filter, file = "Cel-Seurat4_step02.rds")

Cel.exp.filter.tSNE.markers <- FindAllMarkers(Cel.exp.filter.tSNE, logfc.threshold = 0.25, test.use = "roc", only.pos = TRUE)
Cel.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- Cel.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("Cel.step02.10.DoHeatmap.pdf", height=13, width=8)
DoHeatmap(Cel.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

write.table(WhichCells(Cel.exp.filter.tSNE, idents="0"), "Cel.step02.Cell-00.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="1"), "Cel.step02.Cell-01.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="2"), "Cel.step02.Cell-02.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="3"), "Cel.step02.Cell-03.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="4"), "Cel.step02.Cell-04.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="5"), "Cel.step02.Cell-05.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="6"), "Cel.step02.Cell-06.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="7"), "Cel.step02.Cell-07.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="8"), "Cel.step02.Cell-08.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="9"), "Cel.step02.Cell-09.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="10"), "Cel.step02.Cell-10.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="11"), "Cel.step02.Cell-11.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="12"), "Cel.step02.Cell-12.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="13"), "Cel.step02.Cell-13.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="14"), "Cel.step02.Cell-14.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="15"), "Cel.step02.Cell-15.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="16"), "Cel.step02.Cell-16.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="17"), "Cel.step02.Cell-17.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(Cel.exp.filter.tSNE, idents="18"), "Cel.step02.Cell-18.ID.txt", sep="\t", quote = FALSE)
```

To identify the changed information of cell clusters from the analytical results in step01 and step02:

```
# cell cluster list in step02
B01   /Cel.FOACC/step02/Cel.step02.Cell-00.ID.txt
B02   /Cel.FOACC/step02/Cel.step02.Cell-01.ID.txt
B03   /Cel.FOACC/step02/Cel.step02.Cell-02.ID.txt
B04   /Cel.FOACC/step02/Cel.step02.Cell-03.ID.txt
B05   /Cel.FOACC/step02/Cel.step02.Cell-04.ID.txt
B06   /Cel.FOACC/step02/Cel.step02.Cell-05.ID.txt
B07   /Cel.FOACC/step02/Cel.step02.Cell-06.ID.txt
```

```
B08   /Cel.FOACC/step02/Cel.step02.Cell-07.ID.txt
B09   /Cel.FOACC/step02/Cel.step02.Cell-08.ID.txt
B10   /Cel.FOACC/step02/Cel.step02.Cell-09.ID.txt
B11   /Cel.FOACC/step02/Cel.step02.Cell-10.ID.txt
B12   /Cel.FOACC/step02/Cel.step02.Cell-11.ID.txt
B13   /Cel.FOACC/step02/Cel.step02.Cell-12.ID.txt
B14   /Cel.FOACC/step02/Cel.step02.Cell-13.ID.txt
B15   /Cel.FOACC/step02/Cel.step02.Cell-14.ID.txt
B16   /Cel.FOACC/step02/Cel.step02.Cell-15.ID.txt
B17   /Cel.FOACC/step02/Cel.step02.Cell-16.ID.txt
B18   /Cel.FOACC/step02/Cel.step02.Cell-17.ID.txt
B19   /Cel.FOACC/step02/Cel.step02.Cell-18.ID.txt

perl cell_cluster_change.pl -ori_cell_cluster_list ../step01/Cel.step01.Cell.list -new_cell_cluster_list
Cel.step02.Cell.list -output cell_cluster_change.txt

less cell_cluster_change.txt|awk '{print $2"\t"$3}'|sort | uniq -c | sort -d -k 3 -r |less -S

# the contents of cell_cluster_change.txt
      1 OriCluster      NewCluster
     24 A21     B19
     48 A13     B18
     48 A17     B17
      2 A14     B17
      2 A04     B17
      1 A07     B17
     56 A04     B16
     56 A08     B15
      1 A02     B15
     58 A06     B14
     58 A10     B13
     62 A07     B12
     63 A19     B11
     66 A02     B10
     69 A14     B09
     47 A15     B08
     25 A16     B08
      1 A09     B08
     76 A18     B07
     87 A09     B06
     92 A03     B05
    111 A12     B04
      1 A05     B03
    123 A01     B03
```

9	A18	B02
1	A19	B02
1	A11	B02
1	A01	B02
124	A05	B02
94	A11	B01
41	A20	B01
3	A02	B01
1	A01	B01

The protocol for *Arabidopsis thaliana* roots

1. Download the sequencing raw data (accession number: SRR8257100, SRR8257101) from NCBI.

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR825/SRR8257100/SRR8257100.sra
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR825/SRR8257101/SRR8257101.sra

# Download SRA Toolkit (https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software) and use 'fastq-dump' to split sra files:
fastq-dump --split-files SRR8257100.sra
fastq-dump --split-files SRR8257101.sra

perl change_10x_chromium_lib_sequenced_data_format.pl -read1fq SRR8257100_1.fastq.gz -read2fq SRR8257100_2.fastq.gz -output_prefix YSZT_1_L101
perl change_10x_chromium_lib_sequenced_data_format.pl -read1fq SRR8257101_1.fastq.gz -read2fq SRR8257101_2.fastq.gz -output_prefix YSZT_1_L102

# running Cell Ranger v2.1.1 on May 13, 2019
count --id=RWR1 --fastqs=/data/Arabidopsis_root_Ryu/WT_rep1/10x_format --transcriptome=/data/install/cell_ranger/cellranger-2.1.1/reference/Arabidopsis_thaliana_TAIR10_update/TAIR10 --localcores=20 --localmem=60 --jobmode=local --expect-cells=6000 --uiport=3600 --disable-ui

# obtain three files:
matrix.mtx.gz
genes.tsv.gz
barcodes.tsv.gz
```

```
# cd step01
# in step01 direction
ln -s ../matrix.mtx.gz matrix.mtx.gz
ln -s ../genes.tsv.gz features.tsv.gz
ln -s ../barcodes.tsv.gz barcodes.tsv.gz
```

2. Running Seurat analysis. The R scripts was:

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

AthRootSample.exp.filter <- Read10X(".")

AthRootSample.exp.filter <- CreateSeuratObject(AthRootSample.exp.filter, min.cells=3, min.features=200, project="AthRootSample")

AthRootSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(AthRootSample.exp.filter, pattern = "^ATMG")
AthRootSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(AthRootSample.exp.filter, pattern = "^ATCG")

pdf("AthRoot.step01.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(AthRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthRootSample.exp.filter <- subset(AthRootSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 9600 & percent.mt < 0.07 & percent.chl < 0.1)

pdf("AthRoot.step01.01.VlnPlot.filter.pdf", height=6, width=8)
VlnPlot(AthRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthRootSample.exp.filter <- NormalizeData(AthRootSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 150000)

AthRootSample.exp.filter <- FindVariableFeatures(AthRootSample.exp.filter, selection.method = "vst", nfeatures = 2000)
```

```
top10 <- head(VariableFeatures(AthRootSample.exp.filter), 10)

pdf("AthRoot.step01.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(AthRootSample.exp.filter) + LabelPoints(plot =
VariableFeaturePlot(AthRootSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(AthRootSample.exp.filter)
AthRootSample.exp.filter <- ScaleData(AthRootSample.exp.filter, features = all.genes)

AthRootSample.exp.filter <- RunPCA(AthRootSample.exp.filter, npcs = 50, features = VariableFeatures(object =
AthRootSample.exp.filter))

pdf("AthRoot.step01.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(AthRootSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("AthRoot.step01.04.DimPlot.pdf", height=6, width=6)
DimPlot(AthRootSample.exp.filter, reduction = "pca")
dev.off()

pdf("AthRoot.step01.05.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(AthRootSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

AthRootSample.exp.filter <- JackStraw(AthRootSample.exp.filter, num.replicate = 100, dims = 50)
AthRootSample.exp.filter <- ScoreJackStraw(AthRootSample.exp.filter, dims = 1:30)

pdf("AthRoot.step01.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(AthRootSample.exp.filter, dims = 1:30)
dev.off()

pdf("AthRoot.step01.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(AthRootSample.exp.filter, ndims = 30)
dev.off()

AthRootSample.exp.filter <- FindNeighbors(AthRootSample.exp.filter, dims = 1:30)

AthRootSample.exp.filter <- FindClusters(AthRootSample.exp.filter, resolution = 0.5)

AthRootSample.exp.filter.UMAP <- RunUMAP(AthRootSample.exp.filter, dims = 1:30)
AthRootSample.exp.filter.tSNE <- RunTSNE(AthRootSample.exp.filter, dims = 1:30)

pdf("AthRoot.step01.08.DimPlot.UMAP.pdf", height=3.5, width=5)
```

```
DimPlot(AthRootSample.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("AthRoot.step01.09.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(AthRootSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(14)
pdf("AthRoot.step01.09.DimPlot.tSNE.select.pdf", height=4, width=4.9)
DimPlot(AthRootSample.exp.filter.tSNE, reduction = "tsne", cols=color_list)
dev.off()

AthRootSample.exp.filter.tSNE.markers <- FindAllMarkers(AthRootSample.exp.filter.tSNE, logfc.threshold = 0.25, test.use = "roc", only.pos = TRUE)
AthRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- AthRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)

pdf("AthRoot.step01.10.DoHeatmap.pdf", height=8, width=8)
DoHeatmap(AthRootSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(AthRootSample.exp.filter, file = "AthRoot_step01.rds")

write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="0"), "AthRoot.step01.Ath01_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="1"), "AthRoot.step01.Ath02_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="2"), "AthRoot.step01.Ath03_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="3"), "AthRoot.step01.Ath04_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="4"), "AthRoot.step01.Ath05_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="5"), "AthRoot.step01.Ath06_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="6"), "AthRoot.step01.Ath07_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="7"), "AthRoot.step01.Ath08_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="8"), "AthRoot.step01.Ath09_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="9"), "AthRoot.step01.Ath10_ID.txt", sep="\t", quote = FALSE)
```

```
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="10"), "AthRoot.step01.Ath11_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="11"), "AthRoot.step01.Ath12_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="12"), "AthRoot.step01.Ath13_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="13"), "AthRoot.step01.Ath14_ID.txt", sep="\t", quote = FALSE)

pdf("AthRoot.step01.11.FeaturePlot.select.pdf", height=6.3, width=10)
FeaturePlot(AthRootSample.exp.filter.tSNE, features = c("AT4G36710","AT1G07640","AT1G68810","AT2G37090","AT5G57620","AT3G11550","AT2G34910","AT1G27740","AT2G37260","AT5G18840","AT5G17520","AT1G28290"))
dev.off()
```

3. Preparing data for running SC3 and DIMM-SC

```
#The content of "cell.list" file:
A01  /Ath_roots_Ryu/step01/AthRoot.step01.Ath01_ID.txt
A02  /Ath_roots_Ryu/step01/AthRoot.step01.Ath02_ID.txt
A03  /Ath_roots_Ryu/step01/AthRoot.step01.Ath03_ID.txt
A04  /Ath_roots_Ryu/step01/AthRoot.step01.Ath04_ID.txt
A05  /Ath_roots_Ryu/step01/AthRoot.step01.Ath05_ID.txt
A06  /Ath_roots_Ryu/step01/AthRoot.step01.Ath06_ID.txt
A07  /Ath_roots_Ryu/step01/AthRoot.step01.Ath07_ID.txt
A08  /Ath_roots_Ryu/step01/AthRoot.step01.Ath08_ID.txt
A09  /Ath_roots_Ryu/step01/AthRoot.step01.Ath09_ID.txt
A10  /Ath_roots_Ryu/step01/AthRoot.step01.Ath10_ID.txt
A11  /Ath_roots_Ryu/step01/AthRoot.step01.Ath11_ID.txt
A12  /Ath_roots_Ryu/step01/AthRoot.step01.Ath12_ID.txt
A13  /Ath_roots_Ryu/step01/AthRoot.step01.Ath13_ID.txt
A14  /Ath_roots_Ryu/step01/AthRoot.step01.Ath14_ID.txt

perl reformat_matrix.pl -matrix matrix.mtx.gz -features features.tsv.gz -barcodes barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl ID_from_cell_list.pl -cell_list ../AthRoot.step01.Ath.list    -output cell.id
```

```
perl select_cells_by_cellID.pl -reformat_byCellID ../reformat_MG200_MC3_byCellID.xls.gz -cell_ID_list cell.id -output cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features ../features.tsv.gz -output_prefix step02

gzip step02_features.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byGeneID.xls -cluster 14 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The R scripts is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byGeneID.xls", header = TRUE, sep = "\t", quote = "", row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=14, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K14_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:14, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_14_clusters, "SC3_K14_cell_cluster.txt", quote=FALSE, sep="\t")
```

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../AthRoot.step01.Ath.list -
```

```
SC3_cell_cluster SC3_K14_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K14_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt

perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat_SC3_DIMMSC.type.select.cell.id

# assess the clustering by purity, ARI and NMI scores
perl calculate_purity_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.purity.txt
perl calculate_adjusted_rand_index_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.ARI.txt
perl calculate_normalized_mutual_information_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.NMI.txt

cd ../../step02/
ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat_SC3_DIMMSC.type.select.cell.id selected.cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list selected.cell.id -output selected.cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID selected.cell.txt -features ../step01/comparison_Seurat-SC3-DIMMSC/step02_features.tsv.gz -output_prefix step03

gzip selected.cell.txt
gzip step03_genes.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv

ln -s ../step03_genes.tsv.gz features.tsv.gz
ln -s ../step03_matrix.mtx.gz matrix.mtx.gz
ln -s ../step03_barcodes.tsv barcodes.tsv
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

AthRootSample.exp.filter <- Read10X(".")
```

```
AthRootSample.exp.filter <- CreateSeuratObject(AthRootSample.exp.filter, min.cells=3, min.features=200, project="AthRootSample")

AthRootSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(AthRootSample.exp.filter, pattern = "^ATMG")
AthRootSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(AthRootSample.exp.filter, pattern = "^ATCG")

pdf("AthRoot.step02.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(AthRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthRootSample.exp.filter <- subset(AthRootSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 9600 & percent.mt < 0.07 & percent.chl < 0.1)

pdf("AthRoot.step02.01.VlnPlot.filter.pdf", height=6, width=8)
VlnPlot(AthRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthRootSample.exp.filter <- NormalizeData(AthRootSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 150000)

AthRootSample.exp.filter <- FindVariableFeatures(AthRootSample.exp.filter, selection.method = "vst", nfeatures = 2000)
top10 <- head(VariableFeatures(AthRootSample.exp.filter), 10)

pdf("AthRoot.step02.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(AthRootSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(AthRootSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(AthRootSample.exp.filter)
AthRootSample.exp.filter <- ScaleData(AthRootSample.exp.filter, features = all.genes)

AthRootSample.exp.filter <- RunPCA(AthRootSample.exp.filter, npcs = 50, features = VariableFeatures(object = AthRootSample.exp.filter))

pdf("AthRoot.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(AthRootSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()
```

```
pdf("AthRoot.step02.04.DimPlot.pdf", height=6, width=6)
DimPlot(AthRootSample.exp.filter, reduction = "pca")
dev.off()

pdf("AthRoot.step02.05.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(AthRootSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

AthRootSample.exp.filter <- JackStraw(AthRootSample.exp.filter, num.replicate = 100, dims = 50)
AthRootSample.exp.filter <- ScoreJackStraw(AthRootSample.exp.filter, dims = 1:30)

pdf("AthRoot.step02.06.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(AthRootSample.exp.filter, dims = 1:30)
dev.off()

pdf("AthRoot.step02.07.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(AthRootSample.exp.filter, ndims = 30)
dev.off()

AthRootSample.exp.filter <- FindNeighbors(AthRootSample.exp.filter, dims = 1:30)

AthRootSample.exp.filter <- FindClusters(AthRootSample.exp.filter, resolution = 0.6)

AthRootSample.exp.filter.UMAP <- RunUMAP(AthRootSample.exp.filter, dims = 1:30)
AthRootSample.exp.filter.tSNE <- RunTSNE(AthRootSample.exp.filter, dims = 1:30)

pdf("AthRoot.step02.08.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(AthRootSample.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("AthRoot.step02.09.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(AthRootSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(14)
new_identity_TSNE<-AthRootSample.exp.filter.tSNE
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="3+6", "1"="1", "2"="0", "3"="4", "4"="2",
"5"="7", "6"="5", "7"="9", "8"="10", "9"="11", "10"="12", "11"="8", "12"="13")
pdf("AthRoot.step02.09.DimPlot.tSNE.select.pdf", height=4, width=4.9)
DimPlot(new_identity_TSNE, reduction = "tsne",
cols=c(color_list[4],color_list[2],color_list[1],color_list[5],color_list[3],color_list[8],color_list[6],color_list[10],
color_list[11],color_list[12],color_list[13],color_list[9],color_list[14]))
dev.off()
```

```
#         1 OriCluster        NewCluster OriCluster  NewCluster
#        19 A14         B13     A14  B13
#        63 A09         B12     A09  B12
#       136 A13         B11     A13  B11
#       143 A12         B10     A12  B10
#       153 A11         B09     A11  B09
#       155 A10         B08     A10  B08
#       199 A06         B07     A06  B07
#       227 A08         B06     A08  B06
#       249 A03         B05     A03  B05
#       271 A05         B04     A05  B04
#       351 A01         B03     A01  B03
#       357 A02         B02     A02  B02
#       304 A04         B01     A04  B01
#       108 A07         B01     A07  B01

AthRootSample.exp.filter.tSNE.markers <- FindAllMarkers(AthRootSample.exp.filter.tSNE, logfc.threshold = 0.25, test.use = "roc", only.pos = TRUE)
AthRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- AthRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)

pdf("AthRoot.step02.10.DoHeatmap.pdf", height=8, width=8)
DoHeatmap(AthRootSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(AthRootSample.exp.filter, file = "AthRoot_step02.rds")

write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="0"), "AthRoot.step02.Ath01_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="1"), "AthRoot.step02.Ath02_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="2"), "AthRoot.step02.Ath03_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="3"), "AthRoot.step02.Ath04_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="4"), "AthRoot.step02.Ath05_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="5"), "AthRoot.step02.Ath06_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="6"), "AthRoot.step02.Ath07_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="7"), "AthRoot.step02.Ath08_ID.txt", sep="\t", quote = FALSE)
```

```
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="8"), "AthRoot.step02.Ath09_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="9"), "AthRoot.step02.Ath10_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="10"), "AthRoot.step02.Ath11_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="11"), "AthRoot.step02.Ath12_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthRootSample.exp.filter.tSNE, idents="12"), "AthRoot.step02.Ath13_ID.txt", sep="\t", quote = FALSE)

pdf("AthRoot.step02.11.FeaturePlot.select.pdf", height=6.3, width=10)
FeaturePlot(AthRootSample.exp.filter.tSNE, features = c("AT4G36710","AT1G07640","AT1G68810","AT2G37090","AT5G57620","AT3G11550","AT2G34910","AT1G27740","AT2G37260","AT5G18840","AT5G17520","AT1G28290"))
dev.off()
```

To identify the changed information of cell clusters from the analytical results in step01 and step02:

```
# cell cluster list in step02
B01  /Ath_roots_Ryu/step02/AthRoot.step02.Ath01_ID.txt
B02  /Ath_roots_Ryu/step02/AthRoot.step02.Ath02_ID.txt
B03  /Ath_roots_Ryu/step02/AthRoot.step02.Ath03_ID.txt
B04  /Ath_roots_Ryu/step02/AthRoot.step02.Ath04_ID.txt
B05  /Ath_roots_Ryu/step02/AthRoot.step02.Ath05_ID.txt
B06  /Ath_roots_Ryu/step02/AthRoot.step02.Ath06_ID.txt
B07  /Ath_roots_Ryu/step02/AthRoot.step02.Ath07_ID.txt
B08  /Ath_roots_Ryu/step02/AthRoot.step02.Ath08_ID.txt
B09  /Ath_roots_Ryu/step02/AthRoot.step02.Ath09_ID.txt
B10  /Ath_roots_Ryu/step02/AthRoot.step02.Ath10_ID.txt
B11  /Ath_roots_Ryu/step02/AthRoot.step02.Ath11_ID.txt
B12  /Ath_roots_Ryu/step02/AthRoot.step02.Ath12_ID.txt
B13  /Ath_roots_Ryu/step02/AthRoot.step02.Ath13_ID.txt

perl /10t/linrm/single-cell/bin_singlecell/FOACC_v1/cell_cluster_change.pl -ori_cell_cluster_list ../step01/AthRoot.step01.Ath.list -new_cell_cluster_list AthRoot.step02.Ath.list -output cell_cluster_change.txt

less cell_cluster_change.txt|awk '{print $2"\t"$3}'|sort | uniq -c | sort -d -k 3 -r |less -S

# the contents of cell_cluster_change.txt
```

	OriCluster	NewCluster
19	A14	B13
63	A09	B12
136	A13	B11
2	A03	B10
143	A12	B10
153	A11	B09
155	A10	B08
199	A06	B07
227	A08	B06
1	A05	B06
249	A03	B05
8	A09	B04
271	A05	B04
1	A10	B04
351	A01	B03
357	A02	B02
304	A04	B01
108	A07	B01

The protocol for *Arabidopsis thaliana* leaves

1. Download the sequencing scRNA-seq data from NCBI.

```
# https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE161332
wget https://ftp.ncbi.nlm.nih.gov/geo/series/GSE161nnn/GSE161332/suppl/GSE161332_barcodes.tsv.gz
wget https://ftp.ncbi.nlm.nih.gov/geo/series/GSE161nnn/GSE161332/suppl/GSE161332_features.tsv.gz
wget https://ftp.ncbi.nlm.nih.gov/geo/series/GSE161nnn/GSE161332/suppl/GSE161332_matrix.mtx.gz
```

2. Running Seurat analysis. The R scripts was:

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

AthLeafSample.exp.filter <- Read10X(".")

AthLeafSample.exp.filter <- CreateSeuratObject(AthLeafSample.exp.filter, min.cells=3, min.features=200,
```

```
project="AthLeafSample")

AthLeafSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(AthLeafSample.exp.filter, pattern = "^ATMG")
AthLeafSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(AthLeafSample.exp.filter, pattern = "^ATCG")

pdf("AthLeaf.step01.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(AthLeafSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthLeafSample.exp.filter <- subset(AthLeafSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 8100 & percent.mt < 1 & percent.chl < 75)

pdf("AthLeaf.step01.01.VlnPlot.filter.pdf", height=6, width=8)
VlnPlot(AthLeafSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthLeafSample.exp.filter <- NormalizeData(AthLeafSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 125000)

AthLeafSample.exp.filter <- FindVariableFeatures(AthLeafSample.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(AthLeafSample.exp.filter), 10)

pdf("AthLeaf.step01.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(AthLeafSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(AthLeafSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(AthLeafSample.exp.filter)
AthLeafSample.exp.filter <- ScaleData(AthLeafSample.exp.filter, features = all.genes)

AthLeafSample.exp.filter <- RunPCA(AthLeafSample.exp.filter, npcs = 50, features = VariableFeatures(object = AthLeafSample.exp.filter))

pdf("AthLeaf.step01.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(AthLeafSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("AthLeaf.step01.04.DimHeatmap.pdf", height=16, width=8)
```

```
DimHeatmap(AthLeafSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

AthLeafSample.exp.filter <- JackStraw(AthLeafSample.exp.filter, num.replicate = 100, dims = 50)
AthLeafSample.exp.filter <- ScoreJackStraw(AthLeafSample.exp.filter, dims = 1:30)

pdf("AthLeaf.step01.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(AthLeafSample.exp.filter, dims = 1:30)
dev.off()

pdf("AthLeaf.step01.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(AthLeafSample.exp.filter, ndims = 30)
dev.off()

AthLeafSample.exp.filter <- FindNeighbors(AthLeafSample.exp.filter, dims = 1:30)

AthLeafSample.exp.filter <- FindClusters(AthLeafSample.exp.filter, resolution = 0.5)

AthLeafSample.exp.filter.UMAP <- RunUMAP(AthLeafSample.exp.filter, dims = 1:30)

AthLeafSample.exp.filter.tSNE <- RunTSNE(AthLeafSample.exp.filter, dims = 1:30)

pdf("AthLeaf.step01.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(AthLeafSample.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("AthLeaf.step01.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(AthLeafSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(13)
pdf("AthLeaf.step01.08.DimPlot.tSNE.select.pdf", height=3.2, width=4)
DimPlot(AthLeafSample.exp.filter.tSNE, reduction = "tsne", cols=color_list)
dev.off()

AthLeafSample.exp.filter.tSNE.markers <- FindAllMarkers(AthLeafSample.exp.filter.tSNE, logfc.threshold =
0.25, test.use = "roc", only.pos = TRUE)
AthLeafSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- AthLeafSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("AthLeaf.step01.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(AthLeafSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()
```

```
saveRDS(AthLeafSample.exp.filter, file = "AthLeaf_step01.rds")

write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="0"), "Cells.step01.Ath01_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="1"), "Cells.step01.Ath02_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="2"), "Cells.step01.Ath03_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="3"), "Cells.step01.Ath04_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="4"), "Cells.step01.Ath05_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="5"), "Cells.step01.Ath06_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="6"), "Cells.step01.Ath07_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="7"), "Cells.step01.Ath08_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="8"), "Cells.step01.Ath09_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="9"), "Cells.step01.Ath10_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="10"), "Cells.step01.Ath11_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="11"), "Cells.step01.Ath12_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="12"), "Cells.step01.Ath13_ID.txt", sep="\t", quote = FALSE)

pdf("AthLeaf.step01.10.FeaturePlot.select.pdf", height=6.3, width=8)
FeaturePlot(AthLeafSample.exp.filter.tSNE, features = c("AT2G22330", "AT1G07640", "AT4G19840", "AT5G41920", "AT2G26250", "AT3G24140", "AT5G59870", "AT2G45190", "AT2G40100"))
dev.off()
```

3. Preparing data for running SC3 and DIMM-SC

```
# The content of “Cells.step01.Ath.list” file:
A01  /Ath_leaf_Kim/step01/Cells.step01.Ath01_ID.txt
A02  /Ath_leaf_Kim/step01/Cells.step01.Ath02_ID.txt
A03  /Ath_leaf_Kim/step01/Cells.step01.Ath03_ID.txt
A04  /Ath_leaf_Kim/step01/Cells.step01.Ath04_ID.txt
A05  /Ath_leaf_Kim/step01/Cells.step01.Ath05_ID.txt
A06  /Ath_leaf_Kim/step01/Cells.step01.Ath06_ID.txt
```

```
A07  /Ath_leaf_Kim/step01/Cells.step01.Ath07_ID.txt
A08  /Ath_leaf_Kim/step01/Cells.step01.Ath08_ID.txt
A09  /Ath_leaf_Kim/step01/Cells.step01.Ath09_ID.txt
A10  /Ath_leaf_Kim/step01/Cells.step01.Ath10_ID.txt
A11  /Ath_leaf_Kim/step01/Cells.step01.Ath11_ID.txt
A12  /Ath_leaf_Kim/step01/Cells.step01.Ath12_ID.txt
A13  /Ath_leaf_Kim/step01/Cells.step01.Ath13_ID.txt

perl reformat_matrix.pl -matrix matrix.mtx.gz -features features.tsv.gz -barcodes barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl ID_from_cell_list.pl -cell_list ../Cells.step01.Ath.list -output cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../reformat_MG200_MC3_byCellID.xls -cell_ID_list cell.id -output cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features ../features.tsv.gz -output_prefix step02

gzip step02_features.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byGeneID.xls -cluster 13 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The R scripts is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byGeneID.xls", header = TRUE, sep = "\t", quote = "",
```

```
row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=13, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K13_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:13, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_13_clusters, "SC3_K13_cell_cluster.txt", quote=FALSE, sep="\t")
```

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../Cells.step01.Ath.list -SC3_cell_cluster SC3_K13_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K13_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt

perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat-SC3-DIMMSC.type.select.cell.id

cd ../../step02/

ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat-SC3-DIMMSC.type.select.cell.id select.cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list select.cell.id -output select.cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID select.cell.txt -features ../step01/comparison_Seurat-SC3-DIMMSC/step02_features.tsv.gz -output_prefix step03

gzip step03_features.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv

ln -s step03_features.tsv.gz features.tsv.gz
ln -s step03_matrix.mtx.gz matrix.mtx.gz
ln -s step03_barcodes.tsv.gz barcodes.tsv.gz
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

AthLeafSample.exp.filter <- Read10X(".")

AthLeafSample.exp.filter <- CreateSeuratObject(AthLeafSample.exp.filter, min.cells=3, min.features=200, project="AthLeafSample")

AthLeafSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(AthLeafSample.exp.filter, pattern = "^ATMG")
AthLeafSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(AthLeafSample.exp.filter, pattern = "^ATCG")

pdf("AthLeaf.step02.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(AthLeafSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthLeafSample.exp.filter <- subset(AthLeafSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 8100 & percent.mt < 1 & percent.chl < 75)

pdf("AthLeaf.step02.01.VlnPlot.filter.pdf", height=6, width=8)
VlnPlot(AthLeafSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

AthLeafSample.exp.filter <- NormalizeData(AthLeafSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 125000)

AthLeafSample.exp.filter <- FindVariableFeatures(AthLeafSample.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(AthLeafSample.exp.filter), 10)

pdf("AthLeaf.step02.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(AthLeafSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(AthLeafSample.exp.filter), points = top10, repel = TRUE)
dev.off()
```

```
all.genes <- rownames(AthLeafSample.exp.filter)
AthLeafSample.exp.filter <- ScaleData(AthLeafSample.exp.filter, features = all.genes)

AthLeafSample.exp.filter <- RunPCA(AthLeafSample.exp.filter, npcs = 50, features = VariableFeatures(object =
AthLeafSample.exp.filter))

pdf("AthLeaf.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(AthLeafSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("AthLeaf.step02.04.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(AthLeafSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

AthLeafSample.exp.filter <- JackStraw(AthLeafSample.exp.filter, num.replicate = 100, dims = 50)
AthLeafSample.exp.filter <- ScoreJackStraw(AthLeafSample.exp.filter, dims = 1:30)

pdf("AthLeaf.step02.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(AthLeafSample.exp.filter, dims = 1:30)
dev.off()

pdf("AthLeaf.step02.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(AthLeafSample.exp.filter, ndims = 30)
dev.off()

AthLeafSample.exp.filter <- FindNeighbors(AthLeafSample.exp.filter, dims = 1:30)

AthLeafSample.exp.filter <- FindClusters(AthLeafSample.exp.filter, resolution = 0.5)

AthLeafSample.exp.filter.UMAP <- RunUMAP(AthLeafSample.exp.filter, dims = 1:30)

AthLeafSample.exp.filter.tSNE <- RunTSNE(AthLeafSample.exp.filter, dims = 1:30)

pdf("AthLeaf.step02.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(AthLeafSample.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("AthLeaf.step02.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(AthLeafSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(13)
new_identity_TSNE<-AthLeafSample.exp.filter.tSNE
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="0+11", "1"="1+3", "2"="5", "3"="4", "4"="6",
```

```
"5"="2", "6"="8", "7"="1", "8"="7+9", "9"="10", "10"="12")
pdf("AthLeaf.step02.08.DimPlot.tSNE.select.pdf", height=3.2, width=4.3)
DimPlot(new_identity_TSNE, reduction = "tsne", cols=c(color_list[1], color_list[2], color_list[6], color_list[5],
color_list[7], color_list[3], color_list[9], color_list[2], color_list[8], color_list[11], color_list[13]))
dev.off()

#         1 OriCluster         NewCluster OriCluster  NewCluster
#        42 A13         B11     A13  B11
#        45 A11         B10     A11  B10
#        34 A08         B09     A08  B09
#        17 A10         B09     A10  B09
#        56 A02         B08     A02  B08
#       107 A09         B07     A09  B07
#       115 A03         B06     A03  B06
#       127 A07         B05     A07  B05
#       112 A05         B04     A05  B04
#       239 A06         B03     A06  B03
#       144 A02         B02     A02  B02
#       125 A04         B02     A04  B02
#       250 A01         B01     A01  B01
#        23 A12         B01     A12  B01

AthLeafSample.exp.filter.tSNE.markers <- FindAllMarkers(AthLeafSample.exp.filter.tSNE, logfc.threshold =
0.25, test.use = "roc", only.pos = TRUE)
AthLeafSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- AthLeafSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("AthLeaf.step02.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(AthLeafSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(AthLeafSample.exp.filter, file = "AthLeaf_step02.rds")

write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="0"), "Cells.step02.Ath01_ID.txt", sep="\t",
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="1"), "Cells.step02.Ath02_ID.txt", sep="\t",
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="2"), "Cells.step02.Ath03_ID.txt", sep="\t",
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="3"), "Cells.step02.Ath04_ID.txt", sep="\t",
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="4"), "Cells.step02.Ath05_ID.txt", sep="\t",
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="5"), "Cells.step02.Ath06_ID.txt", sep="\t",
```

```
quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="6"), "Cells.step02.Ath07_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="7"), "Cells.step02.Ath08_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="8"), "Cells.step02.Ath09_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="9"), "Cells.step02.Ath10_ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(AthLeafSample.exp.filter.tSNE, idents="10"), "Cells.step02.Ath11_ID.txt", sep="\t", quote = FALSE)

pdf("AthLeaf.step02.10.FeaturePlot.select.pdf", height=6.3, width=8)
FeaturePlot(AthLeafSample.exp.filter.tSNE, features = c("AT2G22330", "AT1G07640", "AT4G19840", "AT5G41920", "AT2G26250", "AT3G24140", "AT5G59870", "AT2G45190", "AT2G40100"))
dev.off()
```

To identify the changed information of cell clusters from the analytical results in step01 and step02:

```
# cell cluster list in step02
B01  /Ath_leaf_Kim/step02/Cells.step02.Ath01_ID.txt
B02  /Ath_leaf_Kim/step02/Cells.step02.Ath02_ID.txt
B03  /Ath_leaf_Kim/step02/Cells.step02.Ath03_ID.txt
B04  /Ath_leaf_Kim/step02/Cells.step02.Ath04_ID.txt
B05  /Ath_leaf_Kim/step02/Cells.step02.Ath05_ID.txt
B06  /Ath_leaf_Kim/step02/Cells.step02.Ath06_ID.txt
B07  /Ath_leaf_Kim/step02/Cells.step02.Ath07_ID.txt
B08  /Ath_leaf_Kim/step02/Cells.step02.Ath08_ID.txt
B09  /Ath_leaf_Kim/step02/Cells.step02.Ath09_ID.txt
B10  /Ath_leaf_Kim/step02/Cells.step02.Ath10_ID.txt
B11  /Ath_leaf_Kim/step02/Cells.step02.Ath11_ID.txt

perl cell_cluster_change.pl -ori_cell_cluster_list ../step01/Cells.step01.Ath.list -new_cell_cluster_list Cells.step02.Ath.list -output cell_cluster_change.txt

less cell_cluster_change.txt|awk '{print $2"\t"$3}'|sort | uniq -c | sort -d -k 3 -r |less -S

# the contents of cell_cluster_change.txt
      1 OriCluster        NewCluster
     42 A13       B11
     45 A11       B10
```

```
 34 A08  B09
 17 A10  B09
 56 A02  B08
  2 A07  B08
  2 A04  B08
107 A09  B07
  3 A02  B06
  1 A08  B06
  1 A01  B06
115 A03  B06
  1 A11  B05
127 A07  B05
  8 A04  B04
  4 A10  B04
  4 A02  B04
  3 A03  B04
112 A05  B04
  2 A07  B03
239 A06  B03
  2 A08  B02
  1 A10  B02
144 A02  B02
125 A04  B02
  3 A07  B01
250 A01  B01
 23 A12  B01
 10 A08  B01
```

The protocol for rice roots

1. Download the sequencing scRNA-seq data from NCBI.

```
# https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4363200
wget https://ftp.ncbi.nlm.nih.gov/geo/samples/GSM4363nnn/GSM4363200/suppl/GSM4363200_9311_matrix.mtx.gz
wget https://ftp.ncbi.nlm.nih.gov/geo/samples/GSM4363nnn/GSM4363200/suppl/GSM4363200_9311_barcodes.tsv.gz
wget https://ftp.ncbi.nlm.nih.gov/geo/samples/GSM4363nnn/GSM4363200/suppl/GSM4363200_9311_genes.tsv.gz
```

2. Running Seurat analysis. The R scripts was:

```
library(Seurat)
```

```
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

RiceRootSample.exp.filter <- Read10X(".")

RiceRootSample.exp.filter <- CreateSeuratObject(RiceRootSample.exp.filter, min.cells=3, min.features=200, project="RiceRootSample")

pdf("RiceRoot.step01.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(RiceRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

RiceRootSample.exp.filter <- subset(RiceRootSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 5000)

pdf("RiceRoot.step01.01.VlnPlot.filter.pdf", height=6, width=4)
VlnPlot(RiceRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

RiceRootSample.exp.filter <- NormalizeData(RiceRootSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 40000)

RiceRootSample.exp.filter <- FindVariableFeatures(RiceRootSample.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(RiceRootSample.exp.filter), 10)

pdf("RiceRoot.step01.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(RiceRootSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(RiceRootSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(RiceRootSample.exp.filter)
RiceRootSample.exp.filter <- ScaleData(RiceRootSample.exp.filter, features = all.genes)

RiceRootSample.exp.filter <- RunPCA(RiceRootSample.exp.filter, npcs = 50, features = VariableFeatures(object = RiceRootSample.exp.filter))

pdf("RiceRoot.step01.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(RiceRootSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()
```

```
pdf("RiceRoot.step01.04.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(RiceRootSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

RiceRootSample.exp.filter <- JackStraw(RiceRootSample.exp.filter, num.replicate = 100, dims = 50)
RiceRootSample.exp.filter <- ScoreJackStraw(RiceRootSample.exp.filter, dims = 1:30)

pdf("RiceRoot.step01.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(RiceRootSample.exp.filter, dims = 1:30)
dev.off()

pdf("RiceRoot.step01.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(RiceRootSample.exp.filter, ndims = 30)
dev.off()

RiceRootSample.exp.filter <- FindNeighbors(RiceRootSample.exp.filter, dims = 1:30)

RiceRootSample.exp.filter <- FindClusters(RiceRootSample.exp.filter, resolution = 0.5)

RiceRootSample.exp.filter.UMAP <- RunUMAP(RiceRootSample.exp.filter, dims = 1:30)

RiceRootSample.exp.filter.tSNE <- RunTSNE(RiceRootSample.exp.filter, dims = 1:30)

pdf("RiceRoot.step01.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(RiceRootSample.exp.filter.UMAP, reduction = "umap")
dev.off()

pdf("RiceRoot.step01.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(RiceRootSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(14)
pdf("RiceRoot.step01.08.DimPlot.tSNE.select.pdf", height=3.9, width=4.8)
DimPlot(RiceRootSample.exp.filter.tSNE, reduction = "tsne", cols=color_list)
dev.off()

RiceRootSample.exp.filter.tSNE.markers <- FindAllMarkers(RiceRootSample.exp.filter.tSNE, logfc.threshold =
0.25, test.use = "roc", only.pos = TRUE)
RiceRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- RiceRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("RiceRoot.step01.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(RiceRootSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
```

```
dev.off()

saveRDS(RiceRootSample.exp.filter, file = "RiceRoot_step01.rds")

write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="0"), "Cells.step01.rice01.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="1"), "Cells.step01.rice02.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="2"), "Cells.step01.rice03.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="3"), "Cells.step01.rice04.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="4"), "Cells.step01.rice05.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="5"), "Cells.step01.rice06.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="6"), "Cells.step01.rice07.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="7"), "Cells.step01.rice08.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="8"), "Cells.step01.rice09.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="9"), "Cells.step01.rice10.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="10"), "Cells.step01.rice11.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="11"), "Cells.step01.rice12.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="12"), "Cells.step01.rice13.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="13"), "Cells.step01.rice14.ID.txt", sep="\t", quote = FALSE)

pdf("RiceRoot.step01.10.FeaturePlot.select.pdf", height=5.5, width=9)
FeaturePlot(RiceRootSample.exp.filter.tSNE, features = c("LOC-Os03g25280", "LOC-Os03g12290", "LOC-Os04g46810", "LOC-Os08g03450", "LOC-Os03g61470", "LOC-Os03g37490", "LOC-Os07g07860", "LOC-Os06g38960", "LOC-Os01g73700", "LOC-Os01g73980", "LOC-Os10g42750"))
dev.off()
```

3. Preparing data for running SC3 and DIMM-SC

```
# The content of “Cells.step01.rice.list” file:
A01  /Rice_roots_Liu/step01/Cells.step01.rice01.ID.txt
```

```
A02  /Rice_roots_Liu/step01/Cells.step01.rice02.ID.txt
A03  /Rice_roots_Liu/step01/Cells.step01.rice03.ID.txt
A04  /Rice_roots_Liu/step01/Cells.step01.rice04.ID.txt
A05  /Rice_roots_Liu/step01/Cells.step01.rice05.ID.txt
A06  /Rice_roots_Liu/step01/Cells.step01.rice06.ID.txt
A07  /Rice_roots_Liu/step01/Cells.step01.rice07.ID.txt
A08  /Rice_roots_Liu/step01/Cells.step01.rice08.ID.txt
A09  /Rice_roots_Liu/step01/Cells.step01.rice09.ID.txt
A10  /Rice_roots_Liu/step01/Cells.step01.rice10.ID.txt
A11  /Rice_roots_Liu/step01/Cells.step01.rice11.ID.txt
A12  /Rice_roots_Liu/step01/Cells.step01.rice12.ID.txt
A13  /Rice_roots_Liu/step01/Cells.step01.rice13.ID.txt
A14  /Rice_roots_Liu/step01/Cells.step01.rice14.ID.txt

perl reformat_matrix.pl -matrix matrix.mtx.gz -features features.tsv.gz -barcodes barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl /10t/linrm/single-cell/bin_singlecell/FOACC_v1/ID_from_cell_list.pl -cell_list ../Cells.step01.rice.list -output cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../reformat_MG200_MC3_byCellID.xls -cell_ID_list cell.id -output cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features ../features.tsv.gz -output_prefix step02

gzip step02_genes.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

# 14 clusters, which is suggested by Seurat analysis
perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byGeneID.xls -cluster 14 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The R scripts is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byGeneID.xls", header = TRUE, sep = "\t", quote = "", row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=14, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K14_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:14, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_14_clusters, "SC3_K14_cell_cluster.txt", quote=FALSE, sep="\t")
```

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../Cells.step01.rice.list -SC3_cell_cluster SC3_K14_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K14_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt

perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat-SC3-DIMMSC.type.select.cell.id

perl calculate_purity_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.purity.txt

perl calculate_adjusted_rand_index_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.ARI.txt

perl calculate_normalized_mutual_information_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.NMI.txt

cd ../../step02/
ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat-SC3-DIMMSC.type.select.cell.id
```

```
select.cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list select.cell.id -output select.cell.txt

perl generate_ori_matrix.pl select_cell.txt step02_genes.tsv.gz step03

gzip step03_features.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv

ln -s step03_features.tsv.gz features.tsv.gz
ln -s step03_matrix.mtx.gz matrix.mtx.gz
ln -s step03_barcodes.tsv.gz barcodes.tsv.gz
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

RiceRootSample.exp.filter <- Read10X(".")

RiceRootSample.exp.filter <- CreateSeuratObject(RiceRootSample.exp.filter, min.cells=3, min.features=200, project="RiceRootSample")

pdf("RiceRoot.step02.01.VlnPlot.pdf", height=6, width=4)
VlnPlot(RiceRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

RiceRootSample.exp.filter <- subset(RiceRootSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 5000)

pdf("RiceRoot.step02.01.VlnPlot.filter.pdf", height=6, width=4)
VlnPlot(RiceRootSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
dev.off()

RiceRootSample.exp.filter <- NormalizeData(RiceRootSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 30000)

RiceRootSample.exp.filter <- FindVariableFeatures(RiceRootSample.exp.filter, selection.method = "vst",
```

```
nfeatures = 2000)

top10 <- head(VariableFeatures(RiceRootSample.exp.filter), 10)

pdf("RiceRoot.step02.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(RiceRootSample.exp.filter) + LabelPoints(plot =
VariableFeaturePlot(RiceRootSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(RiceRootSample.exp.filter)
RiceRootSample.exp.filter <- ScaleData(RiceRootSample.exp.filter, features = all.genes)

RiceRootSample.exp.filter <- RunPCA(RiceRootSample.exp.filter, npcs = 50, features = VariableFeatures(object
= RiceRootSample.exp.filter))

pdf("RiceRoot.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(RiceRootSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("RiceRoot.step02.04.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(RiceRootSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

RiceRootSample.exp.filter <- JackStraw(RiceRootSample.exp.filter, num.replicate = 100, dims = 50)
RiceRootSample.exp.filter <- ScoreJackStraw(RiceRootSample.exp.filter, dims = 1:30)

pdf("RiceRoot.step02.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(RiceRootSample.exp.filter, dims = 1:30)
dev.off()

pdf("RiceRoot.step02.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(RiceRootSample.exp.filter, ndims = 30)
dev.off()

RiceRootSample.exp.filter <- FindNeighbors(RiceRootSample.exp.filter, dims = 1:30)

RiceRootSample.exp.filter <- FindClusters(RiceRootSample.exp.filter, resolution = 0.5)

RiceRootSample.exp.filter.UMAP <- RunUMAP(RiceRootSample.exp.filter, dims = 1:30)

RiceRootSample.exp.filter.tSNE <- RunTSNE(RiceRootSample.exp.filter, dims = 1:30)

pdf("RiceRoot.step02.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(RiceRootSample.exp.filter.UMAP, reduction = "umap")
```

```
dev.off()

pdf("RiceRoot.step02.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(RiceRootSample.exp.filter.tSNE, reduction = "tsne")
dev.off()

color_list<-hue_pal()(14)
new_identity_TSNE<-RiceRootSample.exp.filter.tSNE
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="0+3", "1"="2", "2"="1", "3"="5", "4"="4", "5"="6", "6"="8", "7"="7+13", "8"="9", "9"="12", "10"="10", "11"="11")
pdf("RiceRoot.step02.08.DimPlot.tSNE.select.pdf", height=3.9, width=4.8)
DimPlot(new_identity_TSNE, reduction = "tsne", cols=c(color_list[1], color_list[3], color_list[2], color_list[6], color_list[5], color_list[7], color_list[9], color_list[8], color_list[10], color_list[13], color_list[11], color_list[12]))
dev.off()

#         1 OriCluster         NewCluster OriCluster  NewCluster
#        17 A12        B12     A12  B12
#        26 A11        B11     A11  B11
#        58 A13        B10     A13  B10
#       140 A10        B09     A10  B09
#         8 A14        B08     A14  B08
#       144 A08        B08     A08  B08
#       154 A09        B07     A09  B07
#       228 A07        B06     A07  B06
#       287 A05        B05     A05  B05
#       299 A06        B04     A06  B04
#       398 A02        B03     A02  B03
#       591 A03        B02     A03  B02
#       565 A01        B01     A01  B01
#       138 A04        B01     A04  B01

RiceRootSample.exp.filter.tSNE.markers <- FindAllMarkers(RiceRootSample.exp.filter.tSNE, logfc.threshold = 0.25, test.use = "roc", only.pos = TRUE)
RiceRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- RiceRootSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)
pdf("RiceRoot.step02.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(RiceRootSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(RiceRootSample.exp.filter, file = "RiceRoot_step02.rds")

write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="0"), "Cells.step02.rice01.ID.txt", sep="\t", quote = FALSE)
```

```
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="1"), "Cells.step02.rice02.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="2"), "Cells.step02.rice03.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="3"), "Cells.step02.rice04.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="4"), "Cells.step02.rice05.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="5"), "Cells.step02.rice06.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="6"), "Cells.step02.rice07.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="7"), "Cells.step02.rice08.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="8"), "Cells.step02.rice09.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="9"), "Cells.step02.rice10.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="10"), "Cells.step02.rice11.ID.txt", sep="\t", quote = FALSE)
write.table(WhichCells(RiceRootSample.exp.filter.tSNE, idents="11"), "Cells.step02.rice12.ID.txt", sep="\t", quote = FALSE)

pdf("RiceRoot.step02.10.FeaturePlot.select.pdf", height=5.5, width=9)
FeaturePlot(RiceRootSample.exp.filter.tSNE, features = c("LOC-Os03g25280", "LOC-Os03g12290", "LOC-Os04g46810", "LOC-Os08g03450", "LOC-Os03g61470", "LOC-Os03g37490", "LOC-Os07g07860", "LOC-Os06g38960", "LOC-Os01g73700", "LOC-Os01g73980", "LOC-Os10g42750"))
dev.off()
```

To identify the changed information of cell clusters from the analytical results in step01 and step02:

```
# cell cluster list in step02, ‘Cells.step02.rice.list’
B01   /Rice_roots_Liu/step02/Cells.step02.rice01.ID.txt
B02   /Rice_roots_Liu/step02/Cells.step02.rice02.ID.txt
B03   /Rice_roots_Liu/step02/Cells.step02.rice03.ID.txt
B04   /Rice_roots_Liu/step02/Cells.step02.rice04.ID.txt
B05   /Rice_roots_Liu/step02/Cells.step02.rice05.ID.txt
B06   /Rice_roots_Liu/step02/Cells.step02.rice06.ID.txt
B07   /Rice_roots_Liu/step02/Cells.step02.rice07.ID.txt
B08   /Rice_roots_Liu/step02/Cells.step02.rice08.ID.txt
B09   /Rice_roots_Liu/step02/Cells.step02.rice09.ID.txt
```

```
B10   /Rice_roots_Liu/step02/Cells.step02.rice10.ID.txt
B11   /Rice_roots_Liu/step02/Cells.step02.rice11.ID.txt
B12   /Rice_roots_Liu/step02/Cells.step02.rice12.ID.txt

perl cell_cluster_change.pl -ori_cell_cluster_list ../step01/Cells.step01.rice.list -new_cell_cluster_list Cells.step02.rice.list -output cell_cluster_change.txt

less cell_cluster_change.txt|awk '{print $2"\t"$3}'|sort | uniq -c | sort -d -k 3 -r |less -S

      1 OriCluster      NewCluster
     17 A12     B12
     26 A11     B11
     58 A13     B10
      1 A10     B10
    140 A10     B09
      8 A14     B08
      1 A01     B08
    144 A08     B08
      1 A10     B07
    154 A09     B07
      3 A11     B06
      2 A02     B06
    228 A07     B06
      2 A06     B05
    287 A05     B05
      5 A02     B04
    299 A06     B04
      1 A09     B04
    398 A02     B03
      2 A07     B03
    591 A03     B02
      3 A05     B02
      1 A09     B02
    565 A01     B01
      3 A02     B01
      1 A08     B01
    138 A04     B01
```

The protocol for maize leaves

1. Download the sequencing scRNA-seq data from NCBI.

```
# https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE157757
GSM4774629_filtered_feature_bc_matrix_1_barcodes.tsv.gz
GSM4774629_filtered_feature_bc_matrix_1_features.tsv.gz
GSM4774629_filtered_feature_bc_matrix_1_matrix.mtx.gz
```

2. Running Seurat analysis. The R scripts was:

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

MaizeSample.exp.filter <- Read10X(".")

MaizeSample.exp.filter <- CreateSeuratObject(MaizeSample.exp.filter, min.cells=3, min.features=200, project="MaizeSample")

MaizeSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(MaizeSample.exp.filter, pattern = "^ZeamMp")
MaizeSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(MaizeSample.exp.filter, pattern = "^ZemaCp")

pdf("MaizeLeaf.step01.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(MaizeSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

MaizeSample.exp.filter <- subset(MaizeSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 6700 & percent.mt < 2.5 & percent.chl < 20)

pdf("MaizeLeaf.step01.01.VlnPlot.filter.pdf", height=6, width=4)
VlnPlot(MaizeSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

MaizeSample.exp.filter <- NormalizeData(MaizeSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 50000)

MaizeSample.exp.filter <- FindVariableFeatures(MaizeSample.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(MaizeSample.exp.filter), 10)

pdf("MaizeLeaf.step01.02.VariableFeaturePlot.pdf", height=5, width=12)
```

```
VariableFeaturePlot(MaizeSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(MaizeSample.exp.filter),
points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(MaizeSample.exp.filter)
MaizeSample.exp.filter <- ScaleData(MaizeSample.exp.filter, features = all.genes)

MaizeSample.exp.filter <- RunPCA(MaizeSample.exp.filter, npcs = 50, features = VariableFeatures(object =
MaizeSample.exp.filter))

pdf("MaizeLeaf.step01.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(MaizeSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("MaizeLeaf.step01.04.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(MaizeSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

MaizeSample.exp.filter <- JackStraw(MaizeSample.exp.filter, num.replicate = 100, dims = 50)
MaizeSample.exp.filter <- ScoreJackStraw(MaizeSample.exp.filter, dims = 1:30)

pdf("MaizeLeaf.step01.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(MaizeSample.exp.filter, dims = 1:30)
dev.off()

pdf("MaizeLeaf.step01.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(MaizeSample.exp.filter, ndims = 30)
dev.off()

MaizeSample.exp.filter <- FindNeighbors(MaizeSample.exp.filter, dims = 1:30)

MaizeSample.exp.filter <- FindClusters(MaizeSample.exp.filter, resolution = 0.5)

MaizeSample.exp.filter.UMAP <- RunUMAP(MaizeSample.exp.filter, dims = 1:30)

MaizeSample.exp.filter.tSNE <- RunTSNE(MaizeSample.exp.filter, dims = 1:30)

pdf("MaizeLeaf.step01.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(MaizeSample.exp.filter.UMAP, reduction = "umap")
#DimPlot(MaizeSample.exp.filter.UMAP, reduction = "umap", label = TRUE)
dev.off()

pdf("MaizeLeaf.step01.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(MaizeSample.exp.filter.tSNE, reduction = "tsne")
```

```
#DimPlot(MaizeSample.exp.filter.tSNE, reduction = "tsne", label = TRUE)
dev.off()

color_list<-hue_pal()(10)
pdf("MaizeLeaf.step01.08.DimPlot.tSNE.select.pdf", height=3, width=3.7)
DimPlot(MaizeSample.exp.filter.tSNE, reduction = "tsne", cols=color_list)
dev.off()

MaizeSample.exp.filter.tSNE.markers <- FindAllMarkers(MaizeSample.exp.filter.tSNE, logfc.threshold = 0.25,
test.use = "roc", only.pos = TRUE)
MaizeSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- MaizeSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)

pdf("MaizeLeaf.step01.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(MaizeSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(MaizeSample.exp.filter, file = "MaizeLeaf_step01.rds")

write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="0"), "Cells.step01.maize01.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="1"), "Cells.step01.maize02.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="2"), "Cells.step01.maize03.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="3"), "Cells.step01.maize04.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="4"), "Cells.step01.maize05.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="5"), "Cells.step01.maize06.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="6"), "Cells.step01.maize07.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="7"), "Cells.step01.maize08.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="8"), "Cells.step01.maize09.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="9"), "Cells.step01.maize10.ID.txt", sep="\t", quote
= FALSE)
```

3. Preparing data for running SC3 and DIMM-SC

```
#The content of "Cells.step01.maize.list" file:
A01  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize01.ID.txt
A02  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize02.ID.txt
A03  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize03.ID.txt
A04  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize04.ID.txt
A05  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize05.ID.txt
A06  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize06.ID.txt
A07  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize07.ID.txt
A08  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize08.ID.txt
A09  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize09.ID.txt
A10  /Maize_leaf_Bezrutczyk/step01/Cells.step01.maize10.ID.txt

perl reformat_matrix.pl -matrix matrix.mtx.gz -features features.tsv.gz -barcodes barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat

gzip reformat_MG200_MC3_byGeneID.xls
gzip reformat_MG200_MC3_byCellID.xls

mkdir comparison_Seurat-SC3-DIMMSC
cd comparison_Seurat-SC3-DIMMSC

perl /10t/linrm/single-cell/bin_singlecell/FOACC_v1/ID_from_cell_list.pl -cell_list ../Cells.step01.maize.list -output cell.id

perl select_cells_by_cellID.pl -reformat_byCellID ../reformat_MG200_MC3_byCellID.xls -cell_ID_list cell.id -output cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID cell.txt -features ../features.tsv.gz -output_prefix step02

gzip step02_features.tsv
gzip step02_matrix.mtx
gzip step02_barcodes.tsv

perl reformat_matrix.pl -matrix step02_matrix.mtx.gz -features step02_features.tsv.gz -barcodes step02_barcodes.tsv.gz -min_expressed_genes_one_cell 200 -min_cells_expressed_one_gene 3 -output_prefix reformat_step02

# 10 clusters, inferred from Seurat analysis
perl generate_SC3-DIMMSC_Rscript.pl -reformat_byGeneID reformat_step02_MG200_MC3_byGeneID.xls -cluster 10 -output run_SC3-DIMMSC.R
```

4. Running SC3 and DIMM-SC analyses. The R scripts is:

```
library(DIMMSC)
library(SingleCellExperiment)
library(SC3)
library(scater)

data=read.table("reformat_step02_MG200_MC3_byGeneID.xls", header = TRUE, sep = "\t", quote = "", row.names = 1, as.is = TRUE)
data=as.matrix(data)

data_cluster<-DIMMSC(data=data, K=10, method_cluster_initial="kmeans", method_alpha_initial="Ronning", maxiter=200, tol=1e-4, lik.tol=1e-2)

write.table(data_cluster$mem, "DIMMSC_K10_cell_cluster.txt", quote=FALSE, sep="\t")

sce <- SingleCellExperiment(assays = list(counts = data, logcounts = log2(data + 1)))
rowData(sce)$feature_symbol <- rownames(sce)
sce.run <- sc3(sce, ks = 2:10, biology = TRUE, n_cores = 1)

write.table(sce.run$sc3_10_clusters, "SC3_K10_cell_cluster.txt", quote=FALSE, sep="\t")
```

5. select cells

```
perl comparison_Seurat-SC3-DIMMSC.pl -cell_ID_list cell.id -seurat_cell_list ../Cells.step01.maize.list -SC3_cell_cluster SC3_K10_cell_cluster.txt -DIMMSC_cell_cluster DIMMSC_K10_cell_cluster.txt -output comparison_Seurat-SC3-DIMMSC.txt

perl comparison_Seurat-SC3-DIMMSC.type.select.step01.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -comparison_Seurat_SC3_DIMMSC_type comparison_Seurat-SC3-DIMMSC.txt.type.txt -output comparison_Seurat-SC3-DIMMSC.type.select.cell.id

perl calculate_purity_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.purity.txt

perl calculate_adjusted_rand_index_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.ARI.txt

perl calculate_normalized_mutual_information_score.pl -comparison_Seurat_SC3_DIMMSC comparison_Seurat-SC3-DIMMSC.txt -output comparison_Seurat-SC3-DIMMSC.NMI.txt

cd ../../step02/
ln -s ../step01/comparison_Seurat-SC3-DIMMSC/comparison_Seurat-SC3-DIMMSC.type.select.cell.id select.cell.id
```

```
perl select_cells_by_cellID.pl -reformat_byCellID ../step01/comparison_Seurat-SC3-DIMMSC/reformat_step02_MG200_MC3_byCellID.xls -cell_ID_list select.cell.id -output select.cell.txt

perl generate_matrix.pl -reformat_matrix_byCellID select.cell.txt -features ../step01/comparison_Seurat-SC3-DIMMSC/step02_features.tsv.gz -output_prefix step03

gzip select.cell.txt
gzip step03_genes.tsv
gzip step03_matrix.mtx
gzip step03_barcodes.tsv

ln -s step03_features.tsv.gz features.tsv.gz
ln -s step03_matrix.mtx.gz matrix.mtx.gz
ln -s step03_barcodes.tsv.gz barcodes.tsv.gz
```

6. Using Seurat to display clustering

```
library(Seurat)
library(dplyr)
library(Matrix)
library(patchwork)
library(scales)

MaizeSample.exp.filter <- Read10X(".")

MaizeSample.exp.filter <- CreateSeuratObject(MaizeSample.exp.filter, min.cells=3, min.features=200, project="MaizeSample")

MaizeSample.exp.filter[["percent.mt"]] <- PercentageFeatureSet(MaizeSample.exp.filter, pattern = "^ZeamMp")
MaizeSample.exp.filter[["percent.chl"]] <- PercentageFeatureSet(MaizeSample.exp.filter, pattern = "^ZemaCp")

pdf("MaizeLeaf.step02.01.VlnPlot.pdf", height=6, width=8)
VlnPlot(MaizeSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()

MaizeSample.exp.filter <- subset(MaizeSample.exp.filter, subset = nFeature_RNA > 200 & nFeature_RNA < 6700 & percent.mt < 2.5 & percent.chl < 20)

pdf("MaizeLeaf.step02.01.VlnPlot.filter.pdf", height=6, width=4)
VlnPlot(MaizeSample.exp.filter, features = c("nFeature_RNA", "nCount_RNA", "percent.mt", "percent.chl"), ncol = 4)
dev.off()
```

```
MaizeSample.exp.filter <- NormalizeData(MaizeSample.exp.filter, normalization.method = "LogNormalize", scale.factor = 50000)

MaizeSample.exp.filter <- FindVariableFeatures(MaizeSample.exp.filter, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(MaizeSample.exp.filter), 10)

pdf("MaizeLeaf.step02.02.VariableFeaturePlot.pdf", height=5, width=12)
VariableFeaturePlot(MaizeSample.exp.filter) + LabelPoints(plot = VariableFeaturePlot(MaizeSample.exp.filter), points = top10, repel = TRUE)
dev.off()

all.genes <- rownames(MaizeSample.exp.filter)
MaizeSample.exp.filter <- ScaleData(MaizeSample.exp.filter, features = all.genes)

MaizeSample.exp.filter <- RunPCA(MaizeSample.exp.filter, npcs = 50, features = VariableFeatures(object = MaizeSample.exp.filter))

pdf("MaizeLeaf.step02.03.VizDimLoadings.pdf", height=6, width=8)
VizDimLoadings(MaizeSample.exp.filter, dims = 1:2, reduction = "pca")
dev.off()

pdf("MaizeLeaf.step02.04.DimHeatmap.pdf", height=16, width=8)
DimHeatmap(MaizeSample.exp.filter, dims = 1:30, cells = 500, balanced = TRUE)
dev.off()

MaizeSample.exp.filter <- JackStraw(MaizeSample.exp.filter, num.replicate = 100, dims = 50)
MaizeSample.exp.filter <- ScoreJackStraw(MaizeSample.exp.filter, dims = 1:30)

pdf("MaizeLeaf.step02.05.JackStrawPlot.pdf", height=6, width=8)
JackStrawPlot(MaizeSample.exp.filter, dims = 1:30)
dev.off()

pdf("MaizeLeaf.step02.06.ElbowPlot.pdf", height=5, width=5)
ElbowPlot(MaizeSample.exp.filter, ndims = 30)
dev.off()

MaizeSample.exp.filter <- FindNeighbors(MaizeSample.exp.filter, dims = 1:16)

MaizeSample.exp.filter <- FindClusters(MaizeSample.exp.filter, resolution = 0.5)

MaizeSample.exp.filter.UMAP <- RunUMAP(MaizeSample.exp.filter, dims = 1:16)
```

```
MaizeSample.exp.filter.tSNE <- RunTSNE(MaizeSample.exp.filter, dims = 1:16)

pdf("MaizeLeaf.step02.07.DimPlot.UMAP.pdf", height=3.5, width=5)
DimPlot(MaizeSample.exp.filter.UMAP, reduction = "umap")
#DimPlot(MaizeSample.exp.filter.UMAP, reduction = "umap", label = TRUE)
dev.off()

pdf("MaizeLeaf.step02.08.DimPlot.tSNE.pdf", height=3.5, width=5)
DimPlot(MaizeSample.exp.filter.tSNE, reduction = "tsne")
#DimPlot(MaizeSample.exp.filter.tSNE, reduction = "tsne", label = TRUE)
dev.off()

color_list<-hue_pal()(10)
new_identity_TSNE<-MaizeSample.exp.filter.tSNE
new_identity_TSNE<-RenameIdents(new_identity_TSNE, "0"="2", "1"="0", "2"="3", "3"="1", "4"="5",
"5"="4", "6"="6", "7"="7")
pdf("MaizeLeaf.step02.08.DimPlot.tSNE.select.pdf", height=3, width=3.7)
DimPlot(new_identity_TSNE, reduction = "tsne", cols=c(color_list[3], color_list[1], color_list[4], color_list[2],
color_list[6], color_list[5], color_list[7], color_list[8]))
dev.off()
#       1 OriCluster       NewCluster OriCluster  NewCluster
#      58 A08      B8      A08  B8
#     108 A07      B7      A07  B7
#     121 A05      B6      A05  B6
#     184 A06      B5      A06  B5
#     265 A02      B4      A02  B4
#     302 A04      B3      A04  B3
#     409 A01      B2      A01  B2
#     426 A03      B1      A03  B1

MaizeSample.exp.filter.tSNE.markers <- FindAllMarkers(MaizeSample.exp.filter.tSNE, logfc.threshold = 0.25,
test.use = "roc", only.pos = TRUE)
MaizeSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 2, wt = myAUC)

top10 <- MaizeSample.exp.filter.tSNE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = myAUC)

pdf("MaizeLeaf.step02.09.DoHeatmap.pdf", height=8.5, width=8.5)
DoHeatmap(MaizeSample.exp.filter.tSNE, features = top10$gene) + NoLegend()
dev.off()

saveRDS(MaizeSample.exp.filter, file = "MaizeLeaf_step02.rds")

write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="0"), "Cells.step02.maize01.ID.txt", sep="\t", quote
```

```
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="1"), "Cells.step02.maize02.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="2"), "Cells.step02.maize03.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="3"), "Cells.step02.maize04.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="4"), "Cells.step02.maize05.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="5"), "Cells.step02.maize06.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="6"), "Cells.step02.maize07.ID.txt", sep="\t", quote
= FALSE)
write.table(WhichCells(MaizeSample.exp.filter.tSNE, idents="7"), "Cells.step02.maize08.ID.txt", sep="\t", quote
= FALSE)
```

To identify the changed information of cell clusters from the analytical results in step01 and step02:

```
# cell cluster list in step02, 'Cells.step02.maize.list'
B1    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize01.ID.txt
B2    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize02.ID.txt
B3    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize03.ID.txt
B4    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize04.ID.txt
B5    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize05.ID.txt
B6    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize06.ID.txt
B7    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize07.ID.txt
B8    /Maize_leaf_Bezrutczyk/step02/Cells.step02.maize08.ID.txt

perl cell_cluster_change.pl -ori_cell_cluster_list ../step01/Cells.step01.maize.list -new_cell_cluster_list
Cells.step02.maize.list -output cell_cluster_change.txt

less cell_cluster_change.txt|awk '{print $2"\t"$3}'|sort | uniq -c | sort -d -k 3 -r |less -S

      1 OriCluster      NewCluster
     58 A08     B8
      1 A01     B8
      6 A02     B7
    108 A07     B7
    121 A05     B6
      2 A09     B5
      1 A07     B5
```

1	A04	B5
184	A06	B5
265	A02	B4
302	A04	B3
2	A09	B3
1	A01	B3
5	A10	B2
5	A02	B2
409	A01	B2
1	A08	B2
4	A05	B1
426	A03	B1