MAR 30, 2023

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.36wgqjk53vk5/v1

Protocol Citation: María José Pérez J., michela.deleidi 2023. Single cell analysis of iPSC-derived midbrain organoids. **protocols.io** https://dx.doi.org/10.17504/protocols.io.36wgqjk53vk5/v1

Protocol status: Working
We use this protocol and it's working

Created: Mar 28, 2023

Last Modified: Mar 30, 2023

PROTOCOL integer ID: 79542

Keywords: Data preparation, Quality control, Data normalization, Data integration

Single cell analysis of iPSC-derived midbrain organoids

In 1 collection

María José Pérez J.[1], michela.deleidi[1]

[1]German Center for Neurodegenerative Diseases (DZNE), Tübingen, 72076 Germany

Federico Bertoli

ABSTRACT

The following script was used for analysis of gene corrected (GC) versus GBA1 mutant (MUT) midbrain organoids. The purpose was to combine, filter, integrate, and identify clusters and differentially expressed genes sets.

ATTACHMENTS

404-875.docx

GUIDELINES

This script is based on the Seurat tutorials

- https://satijalab.org/seurat/articles/integration_introduction.html
- https://ucdavis-bioinformatics-training.github.io/2019-single-cell-RNA-sequencing-Workshop-UCD_UCSF/scrnaseq_analysis/scRNA_Workshop-PART1.html

Part 1. Data preparation

1

```
# Install and load the libraries
library(Seurat)
library(patchwork)
library(ggplot2)
library(cowplot)
library(magrittr)
library(tidyverse)

# Load and combine 10x Runs
setwd("D:.../analysis_results")

GC1.data <-Read10X_h5("1GC_filtered_feature_bc_matrix.h5")
GC1 <-CreateSeuratObject(counts = GC1.data)
GC1

MUT1.data <-Read10X_h5("1MUT_filtered_feature_bc_matrix.h5")
MUT1 <-CreateSeuratObject(counts = MUT1.data)
MUT1

GC2.data <-Read10X_h5("2GC_filtered_feature_bc_matrix.h5")
GC2 <-CreateSeuratObject(counts = GC2.data)
GC2

MUT2.data <-Read10X_h5("2MUT_filtered_feature_bc_matrix.h5")
MUT2 <-CreateSeuratObject(counts = MUT2.data)
MUT2

GC3.data <-Read10X_h5("3GC_filtered_feature_bc_matrix.h5")
GC3 <-CreateSeuratObject(counts = GC3.data)
GC3

MUT3.data <-Read10X_h5("3MUT_filtered_feature_bc_matrix.h5")
MUT3 <-CreateSeuratObject(counts = MUT3.data)
MUT3

# Merge more than two objects
MUT.combined <-Reduce(function(x,y) merge (x,y, all=T), list (GC1,
MUT1, GC2, MUT2, GC3, MUT3))
```

Part 2. Quality control

2

```
# Add number of genes per UMI for each cell to metadata
MUT.combined$log10GenesPerUMI <-log10(MUT.combined$nFeature_RNA)
/log10(MUT.combined$nCount_RNA)

# Compute percent mito ratio
MUT.combined$mitoRatio <-PercentageFeatureSet(object =
MUT.combined, pattern ="^MT-")
MUT.combined$mitoRatio <- MUT.combined@meta.data$mitoRatio /100

# Create .RData object
save(MUT.combined, file ="Seurat_Object.RData")

# Filter out low quality reads using selected thresholds
filtered_seurat <-subset(x =MUT.combined,subset= (nCount_RNA >=500)
&(nFeature_RNA >=250) &(log10GenesPerUMI >0.80) & (mitoRatio
<0.20))

# Extract counts
counts <-GetAssayData(object =filtered_seurat, slot ="counts")

# Output a logical vector for every gene on whether the more than
zero counts per cell
nonzero <- counts >0

# Sums all TRUE values and returns TRUE if more than 10 TRUE values
per gene
keep_genes <- Matrix::rowSums(nonzero) >=10

# Only keeping those genes expressed in more than 10 cells
filtered_counts <-counts[keep_genes, ]

# Reassign to filtered Seurat object
filtered_seurat <-CreateSeuratObject(filtered_counts, meta.data
=filtered_seurat@meta.data)

# Save.RData object
save(filtered_seurat, file="seurat_filtered.RData")
```

Part 3. Data preparation and normalization

3

```
#Cell cycle scoring. Normalize the counts
seurat_phase <-NormalizeData(filtered_seurat)

# Load cell cycle markers
load("cycle.rda")

# Score cells for cell cycle
seurat_phase <-CellCycleScoring(seurat_phase,g2m.features =
g2m_genes,s.features =s_genes)

# View cell cycle scores and phases assigned to cells
View(seurat_phase@meta.data)

# Identify the most variable genes
seurat_phase <-FindVariableFeatures(seurat_phase,selection.method
="vst",nfeatures =2000, verbose =FALSE)

# Scale the counts
seurat_phase <-ScaleData(seurat_phase)

# Perform PCA
seurat_phase <-RunPCA(seurat_phase)

# Plot the PCA colored by cell cycle phase
DimPlot(seurat_phase,reduction ="pca", group.by="Phase", split.by
="Phase")

# Split seurat object by condition to perform cell cycle scoring
and SCT on all samples
split_seurat <-SplitObject(filtered_seurat, split.by ="orig.ident")

for (i in1:length(split_seurat)) { split_seurat[[i]] <-
NormalizeData(split_seurat[[i]], verbose =TRUE)
split_seurat[[i]] <-CellCycleScoring(split_seurat[[i]],
g2m.features=g2m_genes, s.features=s_genes)
split_seurat[[i]] <-SCTransform(split_seurat[[i]], vars.to.regress
=c("mitoRatio", "S.Score", "G2M.Score")) }
```

Part 4. Data integration and visualization (directly after Pa...

4

```
# Select the most variable features to use for integration
integ_features <-SelectIntegrationFeatures(object.list
```

```
=split_seurat,nfeatures =3000)

# Prepare the SCT list object for integration
split_seurat <-PrepSCTIntegration(object.list
=split_seurat,anchor.features =integ_features)

# Find anchors
integ_anchors <-FindIntegrationAnchors(object.list
=split_seurat,normalization.method ="SCT", anchor.features
=integ_features)

# Integrate across conditions
organoids.combined.sct <-IntegrateData(anchorset
=integ_anchors,normalization.method ="SCT")

# Save integrated seurat object
saveRDS(organoids.combined.sct, "integrated_seurat.rds")

## Run the standard workflow for visualization and clustering
DefaultAssay(organoids.combined.sct) <-"integrated"

organoids.combined.sct <-RunPCA(organoids.combined.sct, verbose
=FALSE)
PCAPlot(organoids.combined.sct,split.by ="orig.ident")
organoids.combined.sct <-RunUMAP(organoids.combined.sct, reduction
="pca", dims =1:40)

# Plot UMAP
DimPlot(organoids.combined.sct, split.by ="orig.ident")

DimPlot(organoids.combined.sct, group.by ="orig.ident")

# Elbow plot
ElbowPlot(object =organoids.combined.sct,ndims =40)

#Find neighbors for cluster analysis
organoids.combined.sct <-FindNeighbors(organoids.combined.sct,
reduction ="pca", dims =1:40)
organoids.combined.sct <-FindClusters(organoids.combined.sct,
resolution =c(0.6))

# Assign identity of clusters
Idents(object =organoids.combined.sct) <-"integrated_snn_res.0.6"

# Visualization
all.genes <-rownames(organoids.combined.sct)
organoids.combined.sct <-ScaleData(organoids.combined.sct, features
=all.genes)
```

```
DimPlot(organoids.combined.sct, reduction ="umap", group.by
="seurat_clusters", label =TRUE,repel =TRUE)

DimPlot(organoids.combined.sct, reduction ="umap", split.by
="orig.ident")
```

Part 5. Obtain information from the datasets

5

```
#Extract identity and sample information from seurat object to
determine the number of cells per cluster per sample
n_cells <-FetchData(organoids.combined.sct,vars =c("ident",
"orig.ident")) %>% dplyr::count(ident, orig.ident) %>%
tidyr::spread(ident, n)

write.csv(n_cells, "n_cells.csv")

## Identify conserved cell type markers
# Switch back to the original data
DefaultAssay(organoids.combined.sct) <-"RNA"
annotations <-read.csv("annotation.txt")

# conserved markers to any cluster
get_conserved <-function(cluster)
{FindConservedMarkers(organoids.combined.sct, ident.1 =cluster,
grouping.var ="orig.ident", only.pos =TRUE)
%>%rownames_to_column(var ="gene") %>%left_join(y
=unique(annotations[, c("gene_name", "description")]),by
=c("gene"="gene_name")) %>%cbind(cluster_id =cluster, .)}

# Iterate function across desired clusters.
conserved_markers <-map_dfr(0:20, get_conserved)

# Extract top 100 markers per cluster
top100 <-conserved_markers %>%mutate(avg_fc =(GC1_avg_log2FC
+GC2_avg_log2FC +GC3_avg_log2FC + MUT1_avg_log2FC +MUT2_avg_log2FC
+MUT3_avg_log2FC) /6) %>%group_by(cluster_id) %>%top_n(n =100,wt =
avg_fc)

write.csv(top100, "Clusters_top100.csv")

## Identifying gene markers for each cluster
```

```
Genes <-c ("SOX2", "DCX", "TH", "NEUROD4") # Change target genes
depending on the cell type

# UMAP plot
FeaturePlot(organoids.combined.sct,reduction ="umap",features =
Genes, sort.cell =TRUE, min.cutoff ='q10', max.cutoff =5,label = T,
pt.size =0.5)

# Violin plot
plots <-VlnPlot(organoids.combined.sct, features =Genes, split.by
="orig.ident", pt.size =0, combine =FALSE)
wrap_plots(plots = plots, ncol =1)

# Dot plot
DotPlot(organoids.combined.sct, features =Genes) +RotatedAxis()
```

Part 6. Merge and analyse subclusters

6

```
## Combine the clusters according to the identity
new.cluster.ids <-c(1="Radial Glia",
2="Neurons",
3="NPC",
4="Oligodendrocytes",
5="Astrocytes", "...")

names(new.cluster.ids) <-levels(organoids.combined.sct)

organoids.combined.newnames <-RenameIdents(organoids.combined.sct,
new.cluster.ids)

DimPlot(organoids.combined.newnames, reduction ="umap", label
=TRUE, pt.size =0.5) +NoLegend()

#Using DE analysis in specific clusters (after merging) MAST

annotations <-read.csv("annotation.txt")

Markers <-FindMarkers(organoids.combined.newnames, ident.1 ="GC1",
ident.2 ="MUT1", group.by ="orig.ident", subset.ident ="Radial
Glia", min.pct =0.1, test.use ="MAST")%>%rownames_to_column(var
="gene") %>%left_join(y =unique(annotations[, c("gene_name",
"description")]),by =c("gene"="gene_name"))

#Save
write.csv(Markers, "DEgenes_cRadial Glia_1couple.csv")
```

Part 7. Create a subset of cells from a selected cluster to re.

7

```
Neurons_subset <-subset(organoids.combined.newnames, idents
="Neurons")
Neurons_subset

DefaultAssay(Neurons_subset) <-"integrated"

# Run the standard workflow for visualization and clustering
Neurons_subset <-RunPCA(Neurons_subset, verbose =FALSE)

# Plot PCA
PCAPlot(Neurons_subset,
split.by ="orig.ident")
```

```
#Run variable features
Neurons_subset <-
FindVariableFeatures(Neurons_subset,selection.method ="vst",
nfeatures =2000, verbose =FALSE)

# Scale the counts
Neurons_subset <-ScaleData(Neurons_subset)

#Find neighbors for cluster analysis
Neurons_subset <-FindNeighbors(Neurons_subset, reduction ="pca",
dims =1:40)

Neurons_subset <-FindClusters(Neurons_subset, resolution =0.4)

# Assign identity of clusters
Idents(object =Neurons_subset) <-"integrated_snn_res.0.4"

# Visualization
all.genes <-rownames(Neurons_subset)
Neurons_subset <-ScaleData(Neurons_subset, features =all.genes)

DimPlot(Neurons_subset, reduction ="umap", group.by ="orig.ident")
DimPlot(Neurons_subset, reduction ="umap")

# to remove a non-specific cluster
Finalcluster <-subset(Neurons_subset, idents =5, invert = T)
DimPlot(Finalcluster, reduction ="umap")

# To visualize the two conditions side-by-side
DimPlot(Neurons_subset, reduction ="umap", split.by ="orig.ident")

## Cluster identification. Find conserved markers to any cluster
DefaultAssay(organoids.combined.sct) <-"RNA"

get_conserved <-function(cluster)
{FindConservedMarkers(Neurons_subset,ident.1 = cluster,
grouping.var ="orig.ident",only.pos =TRUE)
%>%rownames_to_column(var ="gene") %>%left_join(y
=unique(annotations[, c("gene_name", "description")]), by
=c("gene"="gene_name")) %>%cbind(cluster_id = cluster, .)}

# Iterate function across desired clusters.
conserved_markers <-map_dfr(0:8, get_conserved)

# Extract top 100 markers per cluster
top100 <-conserved_markers %>%
mutate(avg_fc =(GC1_avg_log2FC +GC2_avg_log2FC +GC3_avg_log2FC +
```

```
MUT1_avg_log2FC +MUT2_avg_log2FC +MUT3_avg_log2FC) /6) %>%
group_by(cluster_id) %>%
top_n(n =100,
wt = avg_fc)

#OR save
write.csv(top100, "Clusters_top100_Neuronsubset.csv")
```

Part 8. Differential expressed genes using FIndMarkers wit...

8

```
annotations <-read.csv("annotation.txt")

# couple 1
Markers <-FindMarkers(Neurons_subset, ident.1 ="GC1", ident.2
="MUT1", group.by ="orig.ident", subset.ident ="11", min.pct =0.1,
test.use ="MAST")%>%
rownames_to_column(var ="gene") %>%left_join(y
=unique(annotations[, c("gene_name", "description")]), by
=c("gene"="gene_name"))

write.csv(Markers, "DEgenes_C11Neuron_1couple.csv")

# couple 2
Markers <-FindMarkers(Neurons_subset, ident.1 ="GC2", ident.2
="MUT2", group.by ="orig.ident", subset.ident ="11", min.pct =0.1,
test.use ="MAST")%>%rownames_to_column(var ="gene") %>%left_join(y
=unique(annotations[, c("gene_name", "description")]),by
=c("gene"="gene_name"))

write.csv(Markers, "DEgenes_C11Neuron_2couple.csv")

# couple 3

Markers <-FindMarkers(Neurons_subset, ident.1 ="GC3", ident.2
="MUT3", group.by ="orig.ident", subset.ident ="11", min.pct =0.1,
test.use ="MAST")%>%
rownames_to_column(var ="gene") %>%left_join(y
=unique(annotations[, c("gene_name", "description")]), by
=c("gene"="gene_name"))

write.csv(Markers, "DEgenes_C11Neuron_3couple.csv")
```

Part 9. Analysis of differential expressed genes using FIndM.

9

```
# Merge datasets, example analysis for couple 1

organoids.combined.sct <-Reduce(function(x,y) merge(x,y, all=T),
list (split_seurat$GC1, split_seurat$MUT1))
integ_features <-SelectIntegrationFeatures(object.list
=split_seurat, nfeatures =3000)
VariableFeatures(organoids.combined.sct[["SCT"]]) <-integ_features

## Run the standard workflow for visualization and clustering
DefaultAssay(organoids.combined.sct) <-"integrated"

organoids.combined.sct <-RunPCA(organoids.combined.sct, verbose
=FALSE)
PCAPlot(organoids.combined.sct,split.by ="orig.ident")

organoids.combined.sct <-RunUMAP(organoids.combined.sct, reduction
="pca", dims =1:40)

# Plot UMAP
DimPlot(organoids.combined.sct, split.by ="orig.ident")

DimPlot(organoids.combined.sct, group.by ="orig.ident")

# Elbow plot
ElbowPlot(object =organoids.combined.sct,ndims =40)

#Find neighbors for cluster analysis
organoids.combined.sct <-FindNeighbors(organoids.combined.sct,
reduction ="pca", dims =1:40)
organoids.combined.sct <-FindClusters(organoids.combined.sct,
resolution =c(0.6))

# Assign identity of clusters
Idents(object =organoids.combined.sct) <-"integrated_snn_res.0.6"

# Visualization
all.genes <-rownames(organoids.combined.sct)
organoids.combined.sct <-ScaleData(organoids.combined.sct, features
=all.genes)

DimPlot(organoids.combined.sct, reduction ="umap", group.by
="seurat_clusters", label =TRUE,repel =TRUE)
```

```
DimPlot(organoids.combined.sct, reduction ="umap", split.by
="orig.ident")

## Continue withPart 5 and/or 6 if needed

## To analyze the clusters with resolution of 0.6
annotations <-read.csv("annotation.txt")

# Change the cluster as needed
Markers <-FindMarkers(organoids.combined.sct, ident.1 ="GC1",
ident.2 ="MUT1", group.by ="orig.ident", subset.ident ="20",
min.pct =0.1, test.use ="MAST")%>%rownames_to_column(var ="gene")
%>%left_join(y =unique(annotations[, c("gene_name",
"description")]), by =c("gene"="gene_name"))

#save
write.csv(Markers, "DEgenes_Cluster20_couple1.csv")
```