

May 02, 2022

Region velocity estimation and visulization with seurat

DOI

<https://dx.doi.org/10.17504/protocols.io.eq2lynejrvx9/v1>



Chen Chen. Zhang¹

¹BGI researach

BGI research



Chen Zhang

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.eq2lynejrvx9/v1>

Protocol Citation: Chen Chen. Zhang 2022. Region velocity estimation and visulization with seurat. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.eq2lynejrvx9/v1>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: In development

We are still developing and optimizing this protocol

Created: May 02, 2022

Last Modified: May 04, 2022

Protocol Integer ID: 61795

Keywords: visulization with seurat example pipeline, region velocity estimation, seurat example pipeline, using em algorithm, visulization, seurat, dynamical model

Abstract

Example pipeline of region velocity estimation and visulization of steady-state model and dynamical model using EM algorithm in R with Seurat to pretreat scRNA-seq data



Read expression matrices in R

1m

1

1m

```
path_e <- commandArgs(T)[1] #path of exons and introns expression
matrices
path_i <- commandArgs(T)[2]
prefix <- commandArgs(T)[3]
path_RNA <- commandArgs(T)[4]
rd <- commandArgs(T)[5]
Result_dir = paste(rd,"/",prefix,"/",sep = "")
cell_e_gexp = as.matrix(read.table(path_e, row.names=1 , header =
TRUE))
cell_i_gexp = as.matrix(read.table(path_i, row.names=1 , header =
TRUE))
cell_RNA_gexp = as.matrix(read.table(path_RNA, row.names=1 ,
header = TRUE))
cell_gexp = cell_RNA_gexp
```

Alternative step of step 1 to get data from example data of Region velocity package

1m

2

1m

```
library(Regionvelocity)
data(cell_e_gexp_sper_pb)
data(cell_i_gexp_sper_pb)
data(cell_RNA_gexp_sper_pb)
cell_e_gexp = cell_e_gexp_sper_pb
cell_i_gexp = cell_i_gexp_sper_pb
cell_RNA_gexp = cell_RNA_gexp_sper_pb
cell_gexp = cell_RNA_gexp
```

Single cell data pretreatment using Seurat

10m

3

2m

```
library(Seurat)
library(limma)
library(dplyr)
library(magrittr)
library(ggplot2)
scRNA_seurat <- CreateSeuratObject(counts = cell_gexp,p=prefix)
table(Idsents(scRNA_seurat))
head(scRNA_seurat@meta.data)
scRNA_seurat[["percent.mt"]] <- PercentageFeatureSet(object =
scRNA_seurat, pattern = "^Mt-")
pdf(paste(Result_dir,prefix,"_seurat_stat.pdf",sep = ""))
VlnPlot(object = scRNA_seurat, features = c("nFeature_RNA",
"nCount_RNA", "percent.mt"), ncol = 3)
FeatureScatter(object = scRNA_seurat, feature1 = "nCount_RNA",
feature2 = "nFeature_RNA",,pt.size=1.5)
hist(scRNA_seurat$nCount_RNA,breaks = 100)
hist(scRNA_seurat$nFeature_RNA,breaks = 100)
hist(colSums(scRNA_seurat[["RNA"]][@data],breaks = 100,main =
"Total expression before normalization",xlab = "Sum of
expression")
scRNA_seurat <- NormalizeData(object = scRNA_seurat,
normalization.method = "LogNormalize", scale.factor = 10000)
hist(colSums(scRNA_seurat[["RNA"]][@data],breaks = 100,main =
"Total expression after normalization",xlab = "Sum of expression")
scRNA_seurat <- FindVariableFeatures(object = scRNA_seurat,
selection.method = "vst", nfeatures = 2000)
head(scRNA_seurat[["RNA"]][@var.features)
top10 <- head(VariableFeatures(scRNA_seurat), 10)
plot1 <- VariableFeaturePlot(scRNA_seurat)
plot1
LabelPoints(plot = plot1, points = top10, repel = TRUE)
dev.off()
```



```
all.genes <- rownames(scRNA_seurat)
scRNA_seurat <- ScaleData(scRNA_seurat, features = all.genes)
scRNA_seurat <- RunPCA(scRNA_seurat, npcs=50, features =
VariableFeatures(object = scRNA_seurat))
print(scRNA_seurat[["pca"]], dims = 1:5, nfeatures = 5)
pct <-
scRNA_seurat[["pca"]] $\text{@stdev}$ /sum(scRNA_seurat[["pca"]] $\text{@stdev}$ )*100
cumu <- cumsum(pct)
co1 <- which(cumu > 90 & pct < 5)[1]
co2 <- sort(which((pct[1:length(pct) - 1] - pct[2:length(pct)]) >
0.1), decreasing = T)[1] + 1
nPCs <- min(co1,co2)
sprintf("nPCs:%d",nPCs)
```

5

3m

```
pdf(paste(Result_dir,prefix,"_seurat_pca.pdf",sep = ""))
VizDimLoadings(scRNA_seurat, dims = 1:nPCs, reduction = "pca")
DimPlot(scRNA_seurat, reduction = "pca",split.by = 'ident')
DimHeatmap(scRNA_seurat, dims = 1, cells = 500, balanced =
TRUE,fast = F)+scale_fill_viridis_b()
DimHeatmap(object = scRNA_seurat, dims = 1:(nPCs+1), cells = 500,
balanced = TRUE,ncol=3,fast = F)
scRNA_seurat <- JackStraw(object = scRNA_seurat, num.replicate =
100)
scRNA_seurat <- ScoreJackStraw(object = scRNA_seurat, dims = 1:20)
JackStrawPlot(object = scRNA_seurat, dims = 1:15)
ElbowPlot(scRNA_seurat)
dev.off()
scRNA_seurat <- FindNeighbors(object = scRNA_seurat, dims =
1:nPCs,k.param = 20)
scRNA_seurat <- FindClusters(object = scRNA_seurat, resolution =
c(seq(0,1.6,0.2)))
library(clustree)
pdf(paste(Result_dir,prefix,"_seurat_cluster.pdf",sep = ""))
clustree(scRNA_seurat@meta.data, prefix = "RNA_snn_res.")
Idents(object = scRNA_seurat) <- "RNA_snn_res.0.2"
scRNA_seurat <- RunTSNE(object = scRNA_seurat, dims =
1:nPCs,check_duplicates=F)
TSNEPlot(object = scRNA_seurat, pt.size = 1.5, label = TRUE)
scRNA_seurat <- RunUMAP(scRNA_seurat, dims = 1:nPCs)
DimPlot(scRNA_seurat, reduction = "umap",pt.size=1.5,label = TRUE)
dev.off()
```



```
marker.gene <- c("Dmrt1", "Piwil1", "Tex21", "Tnp1", "Cldn11", "Fabp3")
if(!all(is.na(match(marker.gene, rownames(scRNA_seurat))))) {
  cat("Marker genes are existed. Calculating marker
gene(", na.omit(match(marker.gene, rownames(scRNA_seurat))), ")
expression in clusters\n")
pdf(paste(Result_dir, prefix, "_seurat_marker_anno.pdf", sep = ""))
DoHeatmap(object = scRNA_seurat, features = marker.gene)
VlnPlot(object = scRNA_seurat, features = marker.gene)
FeaturePlot(object = scRNA_seurat, features = marker.gene, label =
T, cols = c("lightgrey", "blue", "red"))
DotPlot(object = scRNA_seurat, features = marker.gene)
dev.off()
}
save(scRNA_seurat, file=paste(Result_dir, prefix, "_seurat.Rdata", sep
= ""))
```

Region velocity estimation and visulization using steady-state model

4h 30m

7

3h

```
library(Regionvelocity)
umap_plot <- DimPlot(scRNA_seurat, reduction = "umap", label =
TRUE, pt.size = 1.5)
pdf(paste(Result_dir,prefix,"_seurat_umap.pdf",sep = ""))
umap_plot
dev.off()
emb <- scRNA_seurat@reductions$umap@cell.embeddings
cell.colors <- ggplot_build(umap_plot)$data[[1]]$colour
names(cell.colors) <- rownames(emb)
cell.dist <- as.dist(1-
armaCor(t(scRNA_seurat@reductions$umap@cell.embeddings)))
save(cell.colors,file=paste(Result_dir,prefix,"_seurat_cell_colors
.Rdata",sep = ""))
region_vel <- gene.region.velocity.estimates(cell_e_gexp,
cell_i_gexp, theta.s = T, RNA_mat = cell_RNA_gexp,
fit.quantile =
0.05,
kCells = 10,
n.cores = 8,
cell.dist =
cell.dist )
save(region_vel,file=paste(Result_dir,prefix,"_seurat_velo.Rdata",
sep = ""))
length(region_vel$gamma)
dim(region_vel$projected)
```

8

1h 30m

```
pdf(paste(Result_dir,prefix,"_seurat_pca_velocity.pdf",sep = ""))
pca.velocity.plot(region_vel,
nPcs=2,
plot.cols=1,
cell.colors=cell.colors,
pc.multipliers=c(1, -1),
show.grid.flow = TRUE,
grid.n=20
)
dev.off()
pdf(paste(Result_dir,prefix,"_seurat_emb_velocity.pdf",sep = ""))
show.velocity.on.embedding.cor(emb,region_vel,cell.colors=cell.col
ors,show.grid.flow = TRUE,grid.n=20,arrow.scale=6)
dev.off()
```




Region velocity estimation and visulization using EM algorithm

16h

9

16h

```
region_vel_EM <- gene.EM.velocity.estimates(region_vel,n.cores =
8)
save(region_vel_EM,file=paste(Result_dir,prefix,"_seurat_velo_EM.R
data",sep = ""))
pdf(paste(Result_dir,prefix,"_seurat_pca_velocity_EM.pdf",sep =
""))
pca.velocity.plot(region_vel_EM,
                  nPcs=2,
                  plot.cols=1,
                  cell.colors=cell.colors,
                  pc.multipliers=c(1,-1), ## adjust as needed to
orient pcs
                  show.grid.flow = TRUE,
                  grid.n=20,
                  arrow.scale=1
)
dev.off()
pdf(paste(Result_dir,prefix,"_seurat_emb_velocity_EM.pdf",sep =
""))
show.velocity.on.embedding.cor(emb,region_vel_EM,cell.colors=cell.
colors,show.grid.flow = TRUE,grid.n=20,arrow.scale=6)
dev.off()
```