droplet-spatial: supporting materila to Computational analysis of single cell RNAseq data chapter

Raffaele A Calogero

30/01/2020

Contents

Contents

Introduction

This github contains all the steps for an examplary analysis of droplet-based RNAseq and spatial-transcriptomics data, described in chapter *Computational analysis of single cell RNAseq data* in Methods in Molecular Biology, (vol. XX, 202X)

3.1 From fastq to counts table

10XGenomics example

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
# downloading the genome reference for human
setwd(paste(home, "droplet-spatial/genomes", sep="/"))
system("wget  http://cf.10xgenomics.com/supp/cell-exp/refdata-cellranger-GRCh38-3.0.0.tar.gz")
system("gzip -d refdata-cellranger-GRCh38-3.0.0.tar.gz")
system("tar xvf refdata-cellranger-GRCh38-3.0.0.tar")
system("rm refdata-cellranger-GRCh38-3.0.0.tar")
hg38reference <-
  paste(home, "droplet-spatial/genomes/refdata-cellranger-GRCh38-3.0.0", sep="/")
scratch <-
  paste(home, "droplet-spatial/scratch", sep="/")
# scratch folder should be located on a SSD disk
################################
library(rCASC)
```

```
#to be done only the first time
downloadContainers()


setwd(paste(home, "droplet-spatial/data", sep="/"))
unzip("SChs1m.zip")
dataset <- paste(home, "droplet-spatial/data/SChs1m, sep="/")
# The cellranger analysis is run without the generation of the secondary analysis
cellrangerCount(group="docker",  transcriptome.folder=hg38reference,
                fastq.folder=dataset,  expect.cells=3000,
                nosecondary=TRUE, scratch.folder=scratch)
```

Spatial transcriptomics example

```
home <- getwd()
setwd(paste(home, "droplet-spatial/genomes", sep="/"))
system("wget  http://cf.10xgenomics.com/supp/spatial-exp/refdata-cellranger-mm10-3.0.0.tar.gz")
system("gzip -d refdata-cellranger-mm10-3.0.0.tar.gz")
system("tar xvf refdata-cellranger-mm10-3.0.0.tar")
system("rm refdata-cellranger-mm10-3.0.0.tar")
mm10reference <-
  paste(home, "droplet-spatial/genomes/refdata-cellranger-mm10-3.0.0", sep="/")
scratch <-  paste(home, "droplet-spatial/scratch", sep="/")
# scratch folder should be located on a SSD disk
################################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))
dir.create("st")
setwd(paste(home, "droplet-spatial/data/st", sep="/"))
dataset <- paste(home, "droplet-spatial/data/st, sep="/")
system("wget http://s3-us-west-2.amazonaws.com/10x.files/samples/
spatial-exp/1.0.0/V1_Mouse_Kidney/V1_Mouse_Kidney_fastqs.tar")
system("tar xvf V1_Mouse_Kidney_fastqs.tar")
```

```
system("rm V1_Mouse_Kidney_fastqs.tar")
fastqs <- paste(home, "droplet-spatial/data/V1_Mouse_Kidney_fastqs", sep="/")

system("wget http://cf.10xgenomics.com/samples/spatial-exp/1.0.0/
V1_Mouse_Kidney/V1_Mouse_Kidney_image.tif")
image <- paste(home, "droplet-spatial/data/st/V1_Mouse_Kidney_image.tif", sep="/")

stpipeline(group="docker", scratch.folder=scratch, data.folder=dataset,
genome.folder=mm10reference, fastqPathFolder=fastqs,
ID="kidneyst",imgNameAndPath=image, slide="V19L29-096",area="B1")
```

3.2 Cells QC

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))
unzip("setA_5x100cells.txt.zip")
system("wget ftp://ftp.ensembl.org/pub/release-98/gtf/homo_sapiens/Homo_sapiens.GRCh38.98.gtf.gz")
system("gzip -d homo_sapiens/Homo_sapiens.GRCh38.98.gtf.gz")
mitoRiboUmi(group="docker", file=paste(getwd(), "setA_5x100cells.txt", sep="/"),
            scratch.folder=scratch, separator="\t", umiXgene=3,
            gtf.name="Homo_sapiens.GRCh38.98.gtf", bio.type="protein_coding")
```

3.3 Annotation and filtering

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))
unzip("setA_5x100cells.txt.zip")
system("wget ftp://ftp.ensembl.org/pub/release-98/gtf/homo_sapiens/Homo_sapiens.GRCh38.98.gtf.gz")
system("gzip -d homo_sapiens/Homo_sapiens.GRCh38.98.gtf.gz")
scannobyGtf(group="docker", file=paste(getwd(),"testSCumi_mm10.csv",sep="/"),
            gtf.name="Homo_sapiens.GRCh38.98.gtf", biotype="protein_coding",
            mt=TRUE, ribo.proteins=TRUE, umiXgene=3, riboStart.percentage=20,
            riboEnd.percentage=40, mitoStart.percentage=1,
            mitoEnd.percentage=20, thresholdGenes=100)
```

3.4 Selecting top ranked genes

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
```

```
setwd(paste(home, "droplet-spatial/data", sep="/"))
topx(group="docker", file=paste(getwd(), "setA_5x100cells.txt", sep="/"),
           threshold=10000, type="variance", separator="\t")

topx(group="docker", file=paste(getwd(), "filtered_variance_setA_5x100cells.txt", sep="/"),
           threshold=5000, type="expression", separator="\t")
```

3.5 Clustering

3.5.1 Clustering with tSne

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))
tsneBootstrap(group="docker",scratch.folder=scratch, file=paste(getwd(),
              "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
              nPerm=80, permAtTime=16, percent=10, range1=6, range2=6,
              separator="\t", logTen=0, seed=111, sp=0.8, perplexity=10)
```

3.5.2 Clustering with SIMLR

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
```

```
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))

simlrBootstrap(group="docker",scratch.folder=scratch, file=paste(getwd(),
               "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
               nPerm=80, permAtTime=16, percent=10, range1=6, range2=6, separator="\t",
               logTen=0, seed=111)
```

3.5.3 Clustering with Griph

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))

griphBootstrap(group="docker",scratch.folder=scratch, file=paste(getwd(),
               "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
               nPerm=80, permAtTime=8, percent=10, separator="\t",logTen=0,
               seed=111)
```

3.5.4 Clustering with Seurat

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
################################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))

seuratPCAEval(group="docker", scratch.folder=scratch,  file=paste(getwd(),
              "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
              separator="\t", logTen = 0, seed = 111, format="NULL")
#optimal threshold of PCs is 5

seuratBootstrap(group="docker",scratch.folder=scratch, file=paste(getwd(),
                "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
                nPerm=80, permAtTime=8, percent=10, separator="\t", logTen=0,
                pcaDimensions=5, seed=111)
```

3.6 Discovering cluster-specific markers

```
home <- getwd()
################################
#cloning in your working folder
system("git clone https://github.com/kendomaniac/droplet-spatial.git")
#if it was already cloned the first time
setwd(paste(home, "droplet-spatial", sep="/"))
system("git pull")
```

```
##############################
library(rCASC)
setwd(paste(home, "droplet-spatial/data", sep="/"))


# run chunk 3.5.4


cometsc(group="docker", file=paste(getwd(),
        "filtered_expression_filtered_variance_setA_5x100cells.txt", sep="/"),
        scratch.folder=scratch, threads=6, counts="True", skipvis="False",
        nCluster=6, separator="\t")
```