01_explore_archr_demodata

May 1, 2020

0.1 Testing the ArchR framework on provided demo datasets

Lucas T. Graybuck 2020-05-01

0.1.1 Overview

In this notebook, we'll test the ArchR pipeline for scATAC analysis. ArchR is a new framework built by the Greenleaf lab (Jeffrey Granja and Ryan Corces in particular), with great documentation supplied at https://www.archrproject.com

We'll run their tutorial on data provided with the ArchR package.

0.1.2 Session Preparation

Load packages

[1]:
```
start_time <- Sys.time()

quiet_library <- function(...) {
  suppressPackageStartupMessages(library(...))
}
quiet_library(ArchR)

options(stringsAsFactors = FALSE)
```

Set parameters ArchR has a few global parameters, so we'll set those here:

[2]:
```
set.seed(3030)

addArchRThreads(16)
addArchRGenome("hg19")
inputFiles <- getTutorialData("Hematopoiesis")
```

```
Setting default number of Parallel threads to 16.
Setting default genome to Hg19.
```

0.1.3 Data preparation and QC

Next, we'll generate the Arrow Files that ArchR uses for analysis. Arrow should be a very fast format for data storage and access.

Build ArrowFiles

[3]:

```
ArrowFiles <- createArrowFiles(inputFiles = inputFiles,
                               sampleNames = names(inputFiles),
                               filterTSS = 4,
                               filterFrags = 1000,
                               addTileMat = TRUE,
                               addGeneScoreMat = TRUE)
```

```
Using GeneAnnotation set by addArchRGenome(Hg19)!
Using GeneAnnotation set by addArchRGenome(Hg19)!
ArchR logging to : ArchRLogs/ArchR-
createArrows-22c83a0f919f-Date-2020-05-01_Time-18-48-52.log
If there is an issue, please report to github with logFile!
Cleaning Temporary Files
2020-05-01 18:48:52 : Batch Execution w/ safelapply!, 0 mins elapsed.
ArchR logging successful to : ArchRLogs/ArchR-
createArrows-22c83a0f919f-Date-2020-05-01_Time-18-48-52.log
```

Score Doublets

[4]:

```
doubScores <- addDoubletScores(input = ArrowFiles,
                               k = 10,
                               knnMethod = "UMAP",
                               LSIMethod = 1)
```

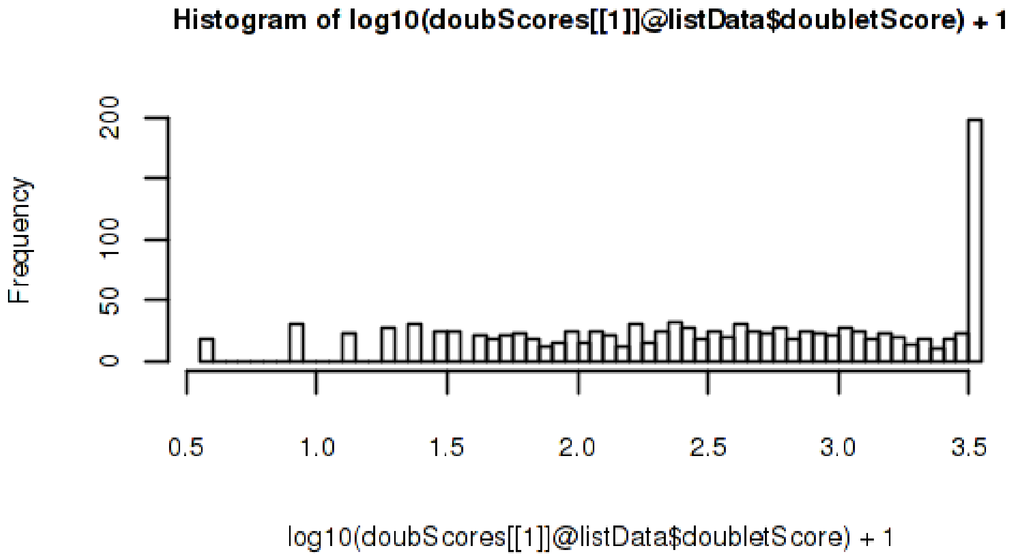
```
ArchR logging to : ArchRLogs/ArchR-
addDoubletScores-22c84d8ea607-Date-2020-05-01_Time-18-48-52.log
If there is an issue, please report to github with logFile!
2020-05-01 18:48:52 : Batch Execution w/ safelapply!, 0 mins elapsed.
2020-05-01 18:48:52 : scATAC_BMMC_R1 (1 of 3) :  Computing Doublet Statistics, 0
mins elapsed.
scATAC_BMMC_R1 (1 of 3) : UMAP Projection R^2 = 0.97354
scATAC_BMMC_R1 (1 of 3) : UMAP Projection R^2 = 0.97354
2020-05-01 18:51:10 : scATAC_CD34_BMMC_R1 (2 of 3) :  Computing Doublet
Statistics, 2.289 mins elapsed.
scATAC_CD34_BMMC_R1 (2 of 3) : UMAP Projection R^2 = 0.98995
scATAC_CD34_BMMC_R1 (2 of 3) : UMAP Projection R^2 = 0.98995
2020-05-01 18:52:42 : scATAC_PBMC_R1 (3 of 3) :  Computing Doublet Statistics,
3.821 mins elapsed.
scATAC_PBMC_R1 (3 of 3) : UMAP Projection R^2 = 0.9758
scATAC_PBMC_R1 (3 of 3) : UMAP Projection R^2 = 0.9758
```

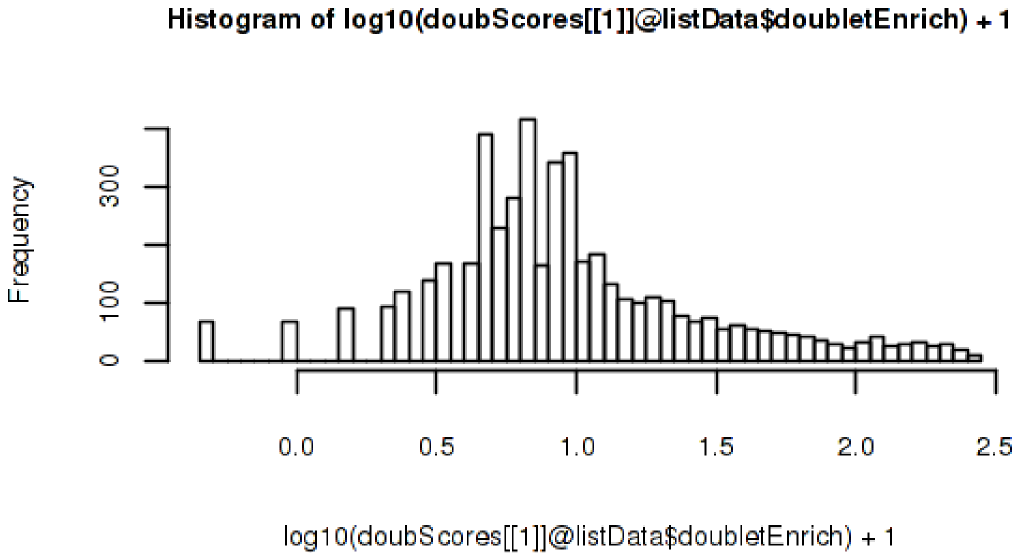
```
ArchR logging successful to : ArchRLogs/ArchR-
addDoubletScores-22c84d8ea607-Date-2020-05-01_Time-18-48-52.log
```

Let’s take a quick look at the distributions of the doubletScores result values:

[5]:

```
options(repr.plot.width = 5, repr.plot.height = 3)
hist(log10(doubScores[[1]]@listData$doubletScore) + 1,
     breaks = 50,
     cex.lab = 0.7,
     cex.axis = 0.7,
     cex.main = 0.7)
hist(log10(doubScores[[1]]@listData$doubletEnrich) + 1,
     breaks = 50,
     cex.lab = 0.7,
     cex.axis = 0.7,
     cex.main = 0.7)
```





Make an ArchRProject For downstream steps, we'll need to now build an ArchRProject object.

[15]:
```r
proj <- ArchRProject(ArrowFiles = ArrowFiles,
                     outputDirectory = "data/",
                     copyArrows = TRUE)
```

```
Using GeneAnnotation set by addArchRGenome(Hg19)!
Using GeneAnnotation set by addArchRGenome(Hg19)!
Validating Arrows…
Getting SampleNames…

Copying ArrowFiles to Ouptut Directory! If you want to save disk space set
copyArrows = FALSE
1 2 3
Getting Cell Metadata…

Merging Cell Metadata…
Initializing ArchRProject…

                                                   / |
                                                 /    \
            .                                  /      |.
            \\\                              /        |.
              \\\                          /           `|.
                \\\                      /              |.
            \                    /                |\
            \\#####\           /                  ||
          ==###########>      /                   ||
           \\##==…\    /                     ||
      ______ =       =|__ /__                     ||      \\\
  ,--' ,----`-,__ ___/'  --,-`-===================##========>
 \               '        ##_______ _____ ,--,__,=##,__   ///
  ,    __==    ___,-,__,--'#'  ==='      `-'    | ##,-/
  -,____,---'       \\####\\________________,--\\_##,/
     ___      .______        ______  __    __  .______
    /   \     |   _  \      /      ||  |  |  | |   _  \
   /  ^  \    |  |_)  |    |  ,----'|  |__|  | |  |_)  |
  /  /_\  \   |      /     |  |     |   __   | |      /
 /  _____  \  |  |\  \\___ |  `----.|  |  |  | |  |\  \\___.
/__/     \__\ | _| `._____| \______||__|  |__| | _| `._____|
```

Fancy ASCII.

Filter Doublets Now, we can apply our detected doublet scores to the data to filter doublets and get some doublet stats:

[16]:
```
proj <- filterDoublets(ArchRProj = proj)
```

```
Filtering 410 cells from ArchRProject!
        scATAC_BMMC_R1 : 243 of 4932 (4.9%)
        scATAC_CD34_BMMC_R1 : 107 of 3275 (3.3%)
        scATAC_PBMC_R1 : 60 of 2454 (2.4%)
```

[17]:
```
getAvailableMatrices(proj)
```

1. 'GeneScoreMatrix' 2. 'TileMatrix'

0.1.4 Dim Reduction and Clustering

ArchR uses "Iterative LSI" To select features for clustering. They recommend using the TileMatrix, which is a 500 bp tiling of the genome.

[18]:
```
proj <- addIterativeLSI(ArchRProj = proj,
                        useMatrix = "TileMatrix",
                        name = "IterativeLSI")
```

```
Checking Inputs…
ArchR logging to : ArchRLogs/ArchR-
addIterativeLSI-22c86cde2092-Date-2020-05-01_Time-18-59-15.log
If there is an issue, please report to github with logFile!
2020-05-01 18:59:16 : Computing Total Accessibility Across All Features, 0.004
mins elapsed.
```

```
2020-05-01 18:59:18 : Computing Top Features, 0.041 mins elapsed.
###########
2020-05-01 18:59:19 : Running LSI (1 of 2) on Top Features, 0.049 mins elapsed.
###########
2020-05-01 18:59:19 : Sampling Cells (N = 10002) for Estimated LSI, 0.05 mins
elapsed.
2020-05-01 18:59:19 : Creating Sampled Partial Matrix, 0.051 mins elapsed.
2020-05-01 18:59:24 : Computing Estimated LSI (projectAll = FALSE), 0.142 mins
elapsed.
2020-05-01 19:00:16 : Identifying Clusters, 1.009 mins elapsed.
2020-05-01 19:00:42 : Identified 5 Clusters, 1.444 mins elapsed.
2020-05-01 19:00:42 : Saving LSI Iteration, 1.444 mins elapsed.
Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
```

```
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."2020-05-01 19:00:59
: Creating Cluster Matrix on the total Group Features, 1.72 mins elapsed.
2020-05-01 19:01:07 : Computing Variable Features, 1.858 mins elapsed.
###########
2020-05-01 19:01:07 : Running LSI (2 of 2) on Variable Features, 1.86 mins
elapsed.
###########
2020-05-01 19:01:07 : Creating Partial Matrix, 1.86 mins elapsed.
2020-05-01 19:01:13 : Computing LSI, 1.952 mins elapsed.
2020-05-01 19:02:03 : Finished Running IterativeLSI, 2.793 mins elapsed.
```

For clustering, ArchR implements Seurat's SNN/Louvain algorithms

[19]:

```r
proj <- addClusters(input = proj,
                    reducedDims = "IterativeLSI")
```

```
ArchR logging to : ArchRLogs/ArchR-
addClusters-22c84135fe36-Date-2020-05-01_Time-19-02-03.log
If there is an issue, please report to github with logFile!
2020-05-01 19:02:04 : Running Seurats FindClusters (Stuart et al. Cell 2019),
0.001 mins elapsed.
Computing nearest neighbor graph
Computing SNN
```

```
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 10251
Number of edges: 496384

Running Louvain algorithm…
Maximum modularity in 10 random starts: 0.8574
Number of communities: 12
Elapsed time: 1 seconds
```

```
2020-05-01 19:02:27 : Testing Outlier Clusters, 0.386 mins elapsed.
2020-05-01 19:02:27 : Assigning Cluster Names to 12 Clusters, 0.387 mins
elapsed.
2020-05-01 19:02:27 : Finished addClusters, 0.388 mins elapsed.
```

We can check these out using UMAP embeddings

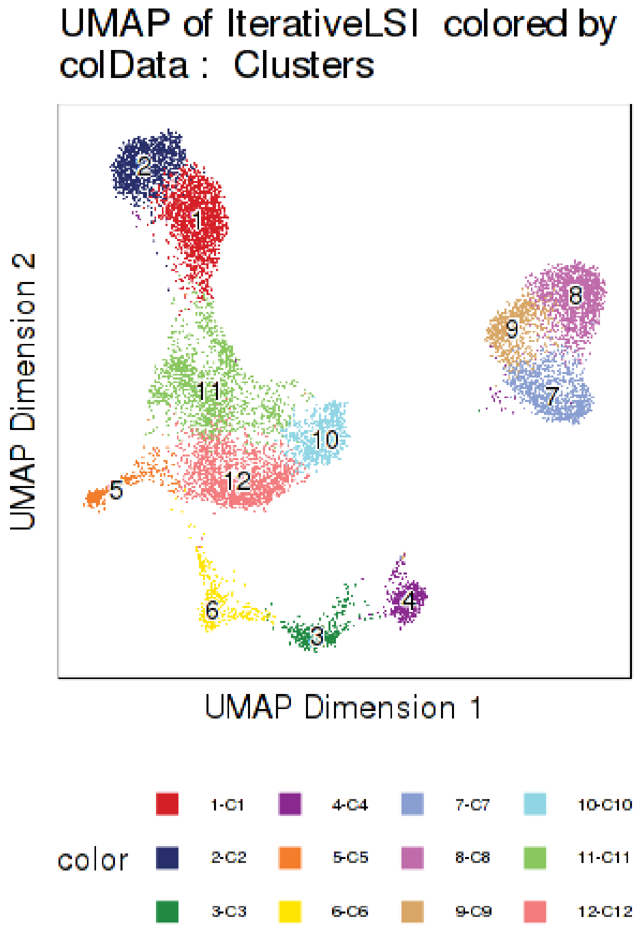
[20]:
```r
proj <- addUMAP(ArchRProj = proj,
                reducedDims = "IterativeLSI")
```

```
19:02:27 UMAP embedding parameters a = 0.7669 b = 1.223
19:02:27 Read 10251 rows and found 30 numeric columns
19:02:27 Using Annoy for neighbor search, n_neighbors = 40
19:02:27 Building Annoy index with metric = cosine, n_trees = 50
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
19:02:29 Writing NN index file to temp file /tmp/Rtmp7VrImc/file22c836e709be
19:02:29 Searching Annoy index using 16 threads, search_k = 4000
19:02:30 Annoy recall = 100%
19:02:31 Commencing smooth kNN distance calibration using 16 threads
19:02:32 Initializing from normalized Laplacian + noise
19:02:32 Commencing optimization for 200 epochs, with 621110 positive edges
19:02:52 Optimization finished
```

[21]:
```r
options(repr.plot.width = 5, repr.plot.height = 5)
p2 <- plotEmbedding(ArchRProj = proj,
                    colorBy = "cellColData",
                    name = "Clusters",
                    embedding = "UMAP")
p2
```

```
ArchR logging to : ArchRLogs/ArchR-
plotEmbedding-22c8de9934c-Date-2020-05-01_Time-19-02-53.log
If there is an issue, please report to github with logFile!
Getting UMAP Embedding
```

```
ColorBy = cellColData
Plotting Embedding
1
ArchR logging successful to : ArchRLogs/ArchR-
plotEmbedding-22c8de9934c-Date-2020-05-01_Time-19-02-53.log
Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."Warning message:
"Use of `dfMean$color` is discouraged. Use `color` instead."
```



[ ]:

[ ]:

0.1.5 Session Info

[13]:

```
sessionInfo()
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Debian GNU/Linux 9 (stretch)
```

```
Matrix products: default
BLAS:   /usr/lib/openblas-base/libblas.so.3
LAPACK: /usr/lib/libopenblasp-r0.2.19.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
 [1] grid      parallel  stats4    stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] uwot_0.1.8                        gridExtra_2.3
 [3] ggrastr_0.1.7                     nabor_0.5.0
 [5] Seurat_3.1.5                      BSgenome.Hsapiens.UCSC.hg19_1.4.0
 [7] BSgenome_1.54.0                   rtracklayer_1.46.0
 [9] Biostrings_2.54.0                 XVector_0.26.0
[11] ArchR_0.9.2                       magrittr_1.5
[13] rhdf5_2.30.1                      Matrix_1.2-17
[15] data.table_1.12.8                 SummarizedExperiment_1.16.1
[17] DelayedArray_0.12.3               BiocParallel_1.20.1
[19] matrixStats_0.56.0                Biobase_2.46.0
[21] GenomicRanges_1.38.0              GenomeInfoDb_1.22.1
[23] IRanges_2.20.2                    S4Vectors_0.24.4
[25] BiocGenerics_0.32.0               ggplot2_3.3.0

loaded via a namespace (and not attached):
 [1] Rtsne_0.15             colorspace_1.4-1         ellipsis_0.3.0
 [4] ggridges_0.5.2         IRdisplay_0.7.0          base64enc_0.1-3
 [7] farver_2.0.3           leiden_0.3.3             listenv_0.8.0
[10] npsurv_0.4-0           ggrepel_0.8.2            RSpectra_0.16-0
[13] codetools_0.2-16       splines_3.6.1            lsei_1.2-0
[16] IRkernel_1.0.2         jsonlite_1.6.1           Cairo_1.5-12
[19] Rsamtools_2.2.3        ica_1.0-2                cluster_2.1.0
[22] png_0.1-7              sctransform_0.2.1        compiler_3.6.1
[25] httr_1.4.1             assertthat_0.2.1         lazyeval_0.2.2
[28] htmltools_0.4.0        tools_3.6.1              rsvd_1.0.3
[31] igraph_1.2.5           gtable_0.3.0             glue_1.4.0
[34] GenomeInfoDbData_1.2.2 RANN_2.6.1               reshape2_1.4.4
[37] dplyr_0.8.5            rappdirs_0.3.1           Rcpp_1.0.4.6
[40] vctrs_0.2.4            gdata_2.18.0             ape_5.3
[43] nlme_3.1-140           lmtest_0.9-37            stringr_1.4.0
[46] globals_0.12.5         lifecycle_0.2.0          irlba_2.3.3
```

```
[49] gtools_3.8.2             XML_3.99-0.3             future_1.17.0
[52] zlibbioc_1.32.0          MASS_7.3-51.4            zoo_1.8-7
[55] scales_1.1.0             RColorBrewer_1.1-2       reticulate_1.15
[58] pbapply_1.4-2            stringi_1.4.6            caTools_1.18.0
[61] repr_1.0.1               rlang_0.4.5              pkgconfig_2.0.3
[64] bitops_1.0-6             evaluate_0.14            lattice_0.20-38
[67] ROCR_1.0-7               purrr_0.3.4              Rhdf5lib_1.8.0
[70] labeling_0.3             GenomicAlignments_1.22.1 patchwork_1.0.0
[73] htmlwidgets_1.5.1        cowplot_1.0.0            tidyselect_1.0.0
[76] RcppAnnoy_0.0.16         plyr_1.8.6               R6_2.4.1
[79] gplots_3.0.3             pbdZMQ_0.3-3             pillar_1.4.3
[82] withr_2.2.0              fitdistrplus_1.0-14      survival_3.1-12
[85] RCurl_1.98-1.2           tibble_3.0.1             future.apply_1.5.0
[88] tsne_0.1-3               crayon_1.3.4             uuid_0.1-4
[91] KernSmooth_2.23-15       plotly_4.9.2             digest_0.6.25
[94] tidyr_1.0.2              munsell_0.5.0            viridisLite_0.3.0
```

[14]:

```
end_time <- Sys.time()
diff_time <- end_time - start_time
time_message <- paste0("Elapsed Time: ",
                       round(diff_time, 3),
                       " ", units(diff_time))
print(time_message)
```

```
[1] "Elapsed Time: 10.072 mins"
```

[ ]:

```

```

[ ]:

```

```