

# R Notebook

```
install.packages('hdf5r', repos = "http://cran.us.r-project.org") #need to read h5 files

##  
## The downloaded binary packages are in  
## /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpSRvlQR/downloaded_packages

setRepositories(ind=1:3) # needed to automatically install Bioconductor dependencies
install.packages("Signac", repos = "http://cran.us.r-project.org") #seurat addon for analyzing chromati

##  
## The downloaded binary packages are in  
## /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpSRvlQR/downloaded_packages

install.packages('Seurat', repos = "http://cran.us.r-project.org")

## also installing the dependencies 'SeuratObject', 'spatstat.explore'

##  
## There are binary versions available but the source versions are later:  
##           binary source needs_compilation  
## SeuratObject 4.1.2 4.1.3          TRUE  
## Seurat       4.2.0 4.2.1          TRUE  
##  
##  
## The downloaded binary packages are in  
## /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpSRvlQR/downloaded_packages

## installing the source packages 'SeuratObject', 'Seurat'

library(Signac)
library(Seurat)

## Attaching SeuratObject

counts <- Read10X_h5(filename = "ATAC/GSM5723631_Young_HSC_filtered_peak_bc_matrix.h5")

meta <- read.csv(  
  file = 'ATAC/GSM5723631_Young_HSC_singlecell.csv.gz',  
  header = TRUE,  
  row.names = 1)
```

```

chrom_assay <- CreateChromatinAssay(
  counts = counts,
  sep = c(":", "-"),
  genome = 'mm10',
  fragments = './ATAC/GSM5723631_Young_HSC_fragments.tsv.gz',
  min.cells = 10,
  min.features = 200
)

## Computing hash

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
## 
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
## 
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
## 
##     expand.grid, I, unname

data <- CreateSeuratObject(
  counts = chrom_assay,
  assay = "peaks",
  meta.data = meta
)

## Warning in CreateSeuratObject.Assay(counts = chrom_assay, assay = "peaks", :
## Some cells in meta.data not present in provided counts matrix.

## Warning: Keys should be one or more alphanumeric characters followed by an
## underscore, setting key from peaks to peaks_

```

```
data[[]]
```

```
##          orig.ident nCount_peaks nFeature_peaks total duplicate
## AAACGAAACTGATTAG-1 SeuratProject      20006        8700    17314     2988
## AAACGAAAGTTCAACC-1 SeuratProject       8290        3904     7086     1064
## AAACGAACAAATTGAG-1 SeuratProject      15697        6854    12583     1790
## AAACGAACAAGCAGGT-1 SeuratProject      20326        8886    17930     2668
## AAACGAACAAGCCCTG-1 SeuratProject      14083        6324    11334     1859
## AAACGAACACACACAT-1 SeuratProject      15845        6794    12901     2368
## AAACGAACACTCCTCA-1 SeuratProject      21027        8831    17283     2372
## AAACGAAGTTCCGG-1 SeuratProject       13747        6119    10569     1083
## AAACTCGAGCGAGCTA-1 SeuratProject      18193        7725    15056     2679
## AAACTCGAGCGTCAAG-1 SeuratProject      12249        5506     9661     1170
## AAAACTCGAGGCCCTCGT-1 SeuratProject     14013        6187    10744     1153
## AAACTCGAGGGCTTCC-1 SeuratProject      28155        11350   23047     3540
## AAACTCGAGGTCACTT-1 SeuratProject      15568        6642    11575     1336
## AAACTCGCAACTCGAT-1 SeuratProject      14555        6371    13122     2476
## AAAACTCGCAATTCTGT-1 SeuratProject     8762         4081    6852      857
## AAAACTCGCACTGATAC-1 SeuratProject     14499        6610    13231     1655
## AAAACTCGCAGGTGTCC-1 SeuratProject     11719        5344    9213      1057
## AAAACTGCAGAGAATTC-1 SeuratProject     14356        6338    11825     1736
## AAAACTGCAGGCCATTCA-1 SeuratProject     27737        10956   24775     3450
## AAAACTGCAGGCCAATG-1 SeuratProject     12990        5824    10293     1337
## AAAACTGCAGTACAGTA-1 SeuratProject     7938         3690    6514      992
## AAAACTGCAGTCCTTCC-1 SeuratProject     17025        7439    15321     2707
## AAAACTGCGTTTCAGCA-1 SeuratProject     13729        6055    10981     1359
## AAAACTGCTCAGGTCTA-1 SeuratProject     17336        7401    15154     3181
## AAAACTGCTCCTCCATG-1 SeuratProject     19644        8364    16945     3233
## AAAACTGCTCTACTGCC-1 SeuratProject     8828         4105    7147      912
## AAAGATGAGTCCAGAG-1 SeuratProject     21038        8762    16838     2034
## AAAGATGTTAACACTC-1 SeuratProject     7818         3733    6279      710
## AAAGATGGTACGCCA-1 SeuratProject      19077        8109    16770     2750
## AAAGATGGTGGTCGAA-1 SeuratProject     30303        12050   27882     6117
## AAAGATGTCGGAGTTT-1 SeuratProject     12760        5879    11215     1397
## AAAGATGTCTATTGTC-1 SeuratProject     12649        5815    10317     1128
## AAAGATGTCTTACCTA-1 SeuratProject     14819        6604    12451     1855
## AAAGGATAGCGCAATG-1 SeuratProject     10601        4826    9083     1635
## AAAGGATAGCTAACAA-1 SeuratProject     6333         3022    5010      771
## AAAGGATAGGCAAGCT-1 SeuratProject     7396         3495    6167      618
## AAAGGATCAAACCGAG-1 SeuratProject     13784        6046    10809     1585
## AAAGGATCACTGTCGG-1 SeuratProject     9925         4548    8563     1340
## AAAGGATCAGGGTAAC-1 SeuratProject     27889        11245   26122     5624
## AAAGGATGTCATCAC-1 SeuratProject     19664        8177    15848     2276
## AAAGGATTTCAGCACGC-1 SeuratProject     10238        4670    8055      730
## AAAGGATTCAAGGCCT-1 SeuratProject     15347        7023    13121     1348
## AAAGGATTCTGGCCAG-1 SeuratProject     17189        7268    14486     2855
## AAAGGATTCTGGCTAA-1 SeuratProject     18409        8098    15772     2265
## AAAGGGCAGCTAGCAG-1 SeuratProject     16282        6947    14132     2901
## AAAGGGCAGTGTCCA-1 SeuratProject     11699        5276    9134     1314
## AAAGGGCAGTCGCGC-1 SeuratProject     10326        4733    7738      771
## AAAGGGCCAACGCACC-1 SeuratProject     13646        6114    11614     1624
## AAAGGGCCAGAGAGAA-1 SeuratProject     20326        8826    17853     2975
## AAAGGGCCTAACATAG-1 SeuratProject     18400        7991    15369     2075
```

## TTGTCTAGTATCAGCT-1	17674
## TTGTCTAGTCATAGCT-1	9566
## TTGTCTAGTCTAAAGA-1	12704
## TTGTCTAGTGAGTCGA-1	15107
## TTGTCTAGTTAACCGA-1	17284
## TTGTCTATCCGTTAGA-1	20468
## TTGTCTATGCCACTT-1	13780
## TTGTTCAAGATAAGGTT-1	22053
## TTGTTCAAGGGAGTTC-1	9589
## TTGTTCAAGTCGTATC-1	14749
## TTGTTCAAGTTACACC-1	12794
## TTGTTCACAAAGCGTC-1	12960
## TTGTTCACAGAACAGC-1	19460
## TTGTTCAGTATCTGCA-1	5182
## TTGTTCAGTATGGTTC-1	16725
## TTGTTCAGTGATAACA-1	18325
## TTGTTCAGTGCGCTCA-1	12103
## TTGTTCAGTGGACTGA-1	3448
## TTGTTCAGTGGCCTCA-1	11703
## TTGTTCAGTTAGTAGA-1	9970
## TTGTTCAGTTCTTC-1	12078
## TTGTTCATCCAAGAGG-1	21881
## TTGTTCATCCATACGA-1	15638
## TTGTTCATCCATTGAG-1	19760
## TTGTTCATCCTCCAA-1	17090
## TTGTTCATCGTCAACA-1	22021
## TTGTTGTAGCCTGTAT-1	18097
## TTGTTGTAGGTCGTT-1	20973
## TTGTTGTAGGTCGTT-1	20975
## TTGTTGTAGTCGCTG-1	10157
## TTGTTGTCAAGCAGGT-1	15277
## TTGTTGTACACATGT-1	23707
## TTGTTGTCACTAGCCG-1	7823
## TTGTTGTAGTCGGC-1	16377
## TTGTTGTAAAGCTA-1	27310
## TTGTTGTAAACCGAG-1	10747
## TTGTTGTGCAACAG-1	18641
## TTGTTGTGTTGTCTT-1	26263
## TTGTTGTTACGATTG-1	14150
## TTTACGTAGAGGTCCA-1	18169
## TTTACGTAGATGGCA-1	20923
## TTTACGTAGTATACC-1	11622
## TTTACGTACTAGAA-1	16935
## TTTACGTATACGCT-1	13449
## TTTACGTGTACAGCT-1	15175
## TTTACGTGTGATAACA-1	21109
## TTTACGTGTTGTCGGC-1	16644
## TTTACGTTCGCAGATT-1	20578
## TTTACGTTCTCTGAGA-1	9342
## TTTGAGGAGGATCCTT-1	5747
## TTTGAGGCCAAGCGTC-1	15070
## TTTGAGGCAGTTAAAG-1	16464
## TTTGAGGCATCATCGA-1	10983
## TTTGAGGCATTATGGC-1	8534

```

## TTTGAGGCATTGATGC-1          29870
## TTTGAGGGTTCGGGAA-1         21580
## TTTGAGGTCAAGCACTA-1        19812
## TTTGAGGTACATCATGG-1        10827
## TTTGAGGTCCACGCTT-1         22660
## TTTGAGGTCCGGACGAA-1        20415
## TTTGAGGTCTTCGTG-1          35590
## TTTGAGGTCTTCGTTA-1         19621
## TTTGCGCCACTGCTTC-1         16384
## TTTGCGCCAGTATCTG-1         11701
## TTTGCGCCTTAGGAA-1          14235
## [ reached 'max' / getOption("max.print") -- omitted 29 rows ]

```

EnsDb.Hsapiens.v86 for human

```

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

## Bioconductor version '3.15' is out-of-date; the current release version '3.16'
##   is available with R version '4.2'; see https://bioconductor.org/install

BiocManager::install("EnsDb.Mmusculus.v79")

## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##   BioCsoft: https://bioconductor.org/packages/3.15/bioc
##   BioCann: https://bioconductor.org/packages/3.15/data/annotation

## Bioconductor version 3.15 (BiocManager 1.30.19), R 4.2.1 (2022-06-23)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'EnsDb.Mmusculus.v79'

## Old packages: 'evaluate', 'ggfun', 'plotly', 'RcppEigen', 'scCATCH', 'sp'

BiocManager::install("GenomeInfoDb") #translation between chromosome names

## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##   BioCsoft: https://bioconductor.org/packages/3.15/bioc
##   BioCann: https://bioconductor.org/packages/3.15/data/annotation
##
## Bioconductor version 3.15 (BiocManager 1.30.19), R 4.2.1 (2022-06-23)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'GenomeInfoDb'

## Old packages: 'evaluate', 'ggfun', 'plotly', 'RcppEigen', 'scCATCH', 'sp'

```

```

BioManager::install("biovizBase")

## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##   BioCsoft: https://bioconductor.org/packages/3.15/bioc
##   BioCAnn: https://bioconductor.org/packages/3.15/data/annotation
##
## Bioconductor version 3.15 (BiocManager 1.30.19), R 4.2.1 (2022-06-23)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'biovizBase'

## Old packages: 'evaluate', 'ggfun', 'plotly', 'RcppEigen', 'scCATCH', 'sp'

library(GenomeInfoDb)
library(EnsDb.Mmusculus.v79)

## Loading required package: ensemblldb

## Loading required package: GenomicRanges

## Loading required package: GenomicFeatures

## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor
##
##   Vignettes contain introductory material; view with
##   'browseVignettes()'. To cite Bioconductor, see
##   'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: AnnotationFilter

##
## Attaching package: 'ensemblldb'

## The following object is masked from 'package:stats':
## 
##   filter

annotations <- GetGRangesFromEnsDb(ensdb = EnsDb.Mmusculus.v79)

```

```

## Warning in .Seqinfo.mergexy(x, y): The 2 combined objects have no sequence levels in common. (Use
##   suppressWarnings() to suppress this warning.)

## Warning in .Seqinfo.mergexy(x, y): The 2 combined objects have no sequence levels in common. (Use
##   suppressWarnings() to suppress this warning.)

## Warning in .Seqinfo.mergexy(x, y): The 2 combined objects have no sequence levels in common. (Use
##   suppressWarnings() to suppress this warning.)

seqlevelsStyle(annotations) <- 'UCSC'

Annotation(data) <- annotations

data <- NucleosomeSignal(object = data) #fragment ratio 147-294: <147

data <- TSSEnrichment(object = data, fast = FALSE)

## Extracting TSS positions

## Finding + strand cut sites

## Finding - strand cut sites

## Computing mean insertion frequency in flanking regions

## Normalizing TSS score

data$blacklist_ratio <- data$blacklist_region_fragments / data$peak_region_fragments

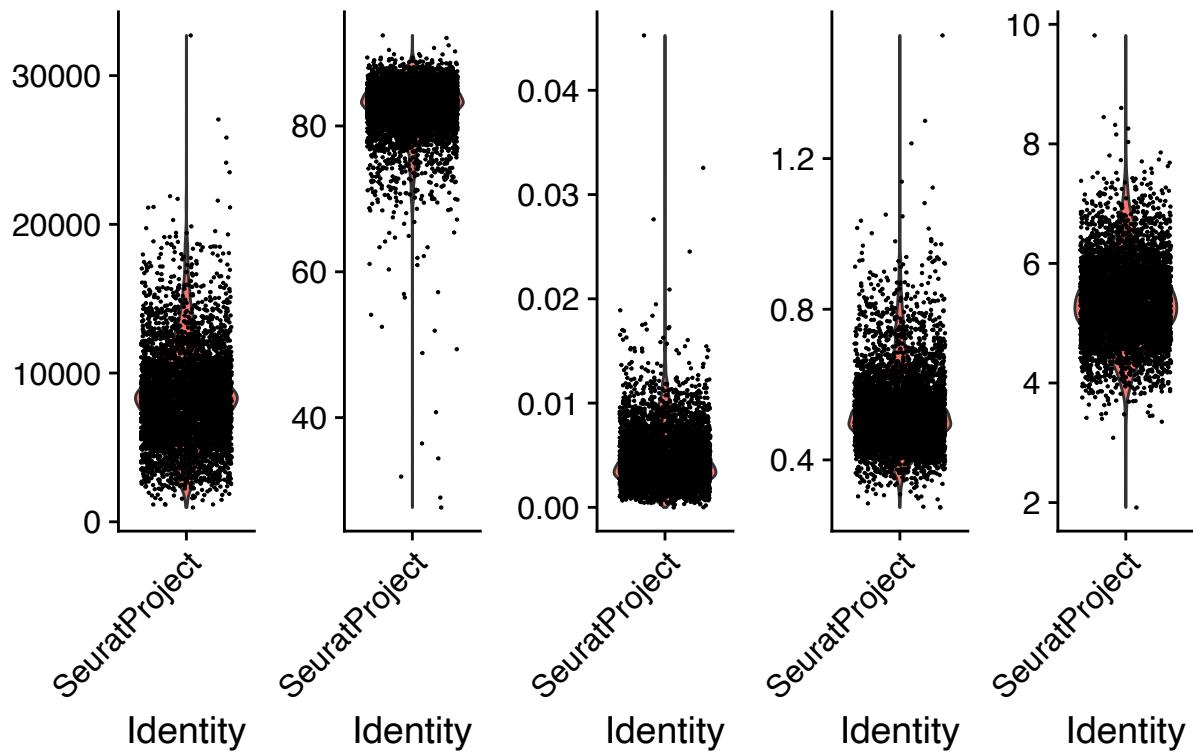
#data[]]

data$pct_reads_in_peaks <- data$peak_region_fragments / data$passed_filters * 100

VlnPlot(
  object = data,
  features = c('peak_region_fragments', 'pct_reads_in_peaks',
              'blacklist_ratio', 'nucleosome_signal', 'TSS.enrichment'),
  pt.size = 0.1,
  ncol = 5
)

```

## peak\_region\_fragments\_in\_blacklist\_ratio\_nucleosome\_TSS\_enrichment



could do this....

```

data <- subset(
  x = data,
  subset = peak_region_fragments > 3000 &
    peak_region_fragments < 20000 &
    pct_reads_in_peaks > 15 &
    blacklist_ratio < 0.05 &
    nucleosome_signal < 4 &
    TSS.enrichment > 2
)

low_prf <- quantile(data[["peak_region_fragments"]]$peak_region_fragments, probs = 0.02)
high_prf <- quantile(data[["peak_region_fragments"]]$peak_region_fragments, probs = 0.98)
low_prp <- quantile(data[["pct_reads_in_peaks"]]$pct_reads_in_peaks, probs = 0.02)

high_blr <- quantile(data[["blacklist_ratio"]]$blacklist_ratio, probs = 0.98)

high_ns <- quantile(data[["nucleosome_signal"]]$nucleosome_signal, probs = 0.98)

low_ts <- quantile(data[["TSS.enrichment"]]$TSS.enrichment, probs = 0.02)

print(low_prf)

##      2%
## 3531.7

```

```

print(hig_prf)

##      98%
## 15713

print(low_prp)

##      2%
## 72.68298

print(high_blr)

##      98%
## 0.01136435

print(hig_ns)

##      98%
## 0.8160401

print(low_ts)

##      2%
## 4.098172

data <- subset(
  x = data,
  subset = peak_region_fragments > low_prf &
  peak_region_fragments < hig_prf &
  pct_reads_in_peaks > low_prp &
  blacklist_ratio < high_blr &
  nucleosome_signal < hig_ns &
  TSS.enrichment > low_ts
)

data

## An object of class Seurat
## 165871 features across 4363 samples within 1 assay
## Active assay: peaks (165871 features, 0 variable features)

Normalization, dimension reduction

data <- RunTFIDF(data)

## Performing TF-IDF normalization

```

```

data <- FindTopFeatures(data, min.cutoff = 'q0')
data

## An object of class Seurat
## 165871 features across 4363 samples within 1 assay
## Active assay: peaks (165871 features, 165871 variable features)

data <- RunSVD(data)

## Running SVD

## Scaling cell embeddings

DepthCor(data)

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

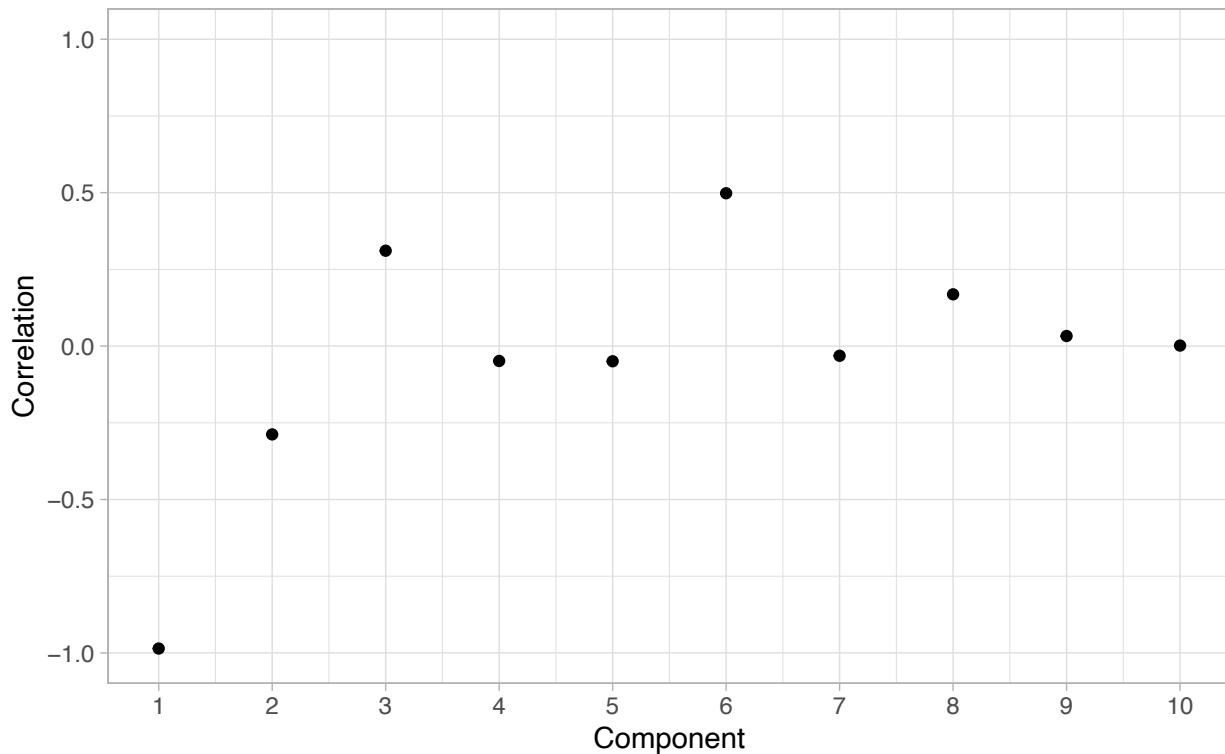
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

```

## Correlation between depth and reduced dimension components

Assay: peaks Reduction: lsi



```
data <- RunUMAP(object = data, reduction = 'lsi', dims = 2:30)

## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session

## 00:05:56 UMAP embedding parameters a = 0.9922 b = 1.112

## 00:05:56 Read 4363 rows and found 29 numeric columns

## 00:05:56 Using Annoy for neighbor search, n_neighbors = 30

## 00:05:56 Building Annoy index with metric = cosine, n_trees = 50

## 0%   10   20   30   40   50   60   70   80   90   100%
## [----|----|----|----|----|----|----|----|----|----|
## ****|*****|*****|*****|*****|*****|*****|*****|*****|*****|
## 00:05:57 Writing NN index file to temp file /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpSR
## 00:05:57 Searching Annoy index using 1 thread, search_k = 3000
## 00:05:58 Annoy recall = 100%
## 00:06:03 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 00:06:08 Initializing from normalized Laplacian + noise (using irlba)
## 00:06:08 Commencing optimization for 500 epochs, with 171814 positive edges
## 00:06:16 Optimization finished
```

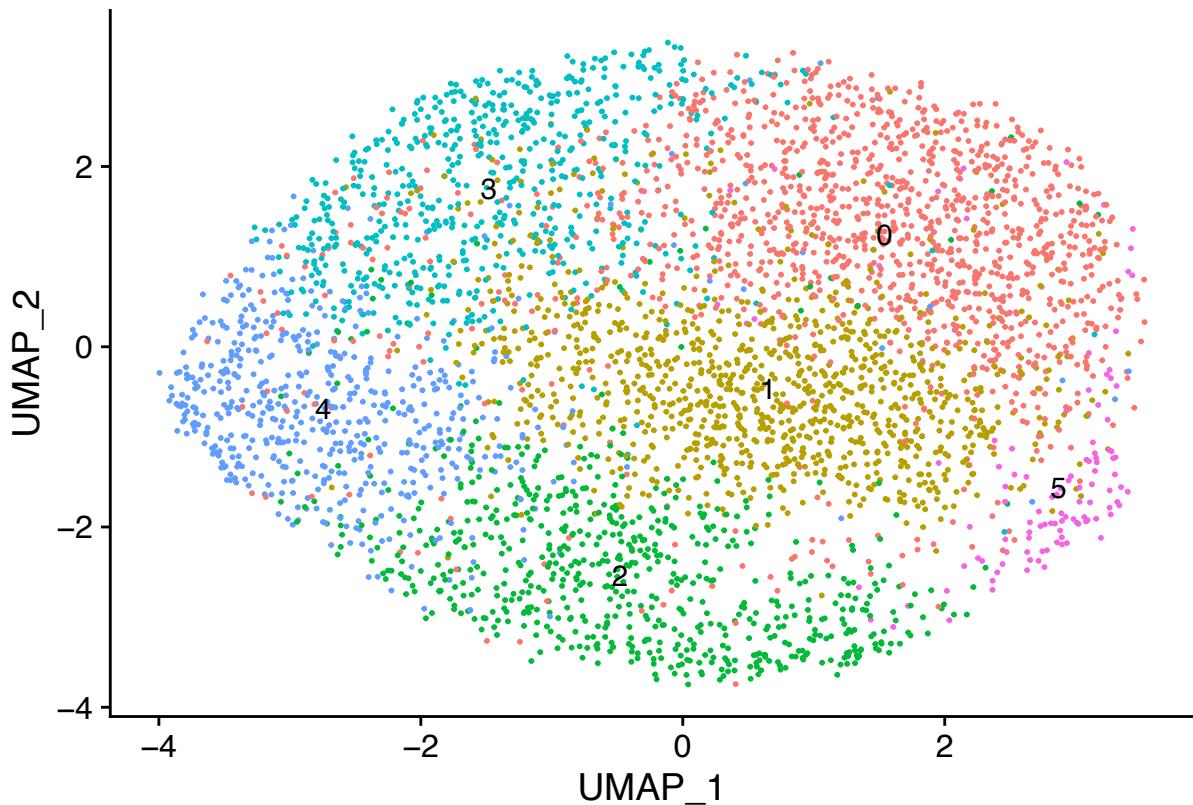
```

data <- FindNeighbors(object = data, reduction = 'lsi', dims = 2:30)

## Computing nearest neighbor graph
## Computing SNN

data <- FindClusters(object = data, verbose = FALSE, algorithm = 3)
DimPlot(object = data, label = TRUE) + NoLegend()

```



Multiple samples....

```

import_atac <- function(count_path, meta_path, fragment_path){
  counts <- Read10X_h5(filename = count_path)

  meta <- read.csv(
    file = meta_path,
    header = TRUE,
    row.names = 1)

  chrom_assay <- CreateChromatinAssay(
    counts = counts,
    sep = c(":", "-"),
    genome = 'mm10',
    fragments = fragment_path,
    min.cells = 10,
    min.features = 200
  )

```

```

)

data <- CreateSeuratObject(
  counts = chrom_assay,
  assay = "peaks",
  meta.data = meta
)

Annotation(data) <- annotations

data <- NucleosomeSignal(object = data) #fragment ratio 147-294: <147 --- mononucleosome:nucleosome

data <- TSSEnrichment(object = data, fast = FALSE)

data$blacklist_ratio <- data$blacklist_region_fragments / data$peak_region_fragments

data$pct_reads_in_peaks <- data$peak_region_fragments / data$passed_filters * 100

low_prf <- quantile(data[["peak_region_fragments"]]$peak_region_fragments, probs = 0.02)
high_prf <- quantile(data[["peak_region_fragments"]]$peak_region_fragments, probs = 0.98)
low_prr <- quantile(data[["pct_reads_in_peaks"]]$pct_reads_in_peaks, probs = 0.02)

high_blr <- quantile(data[["blacklist_ratio"]]$blacklist_ratio, probs = 0.98)

high_ns <- quantile(data[["nucleosome_signal"]]$nucleosome_signal, probs = 0.98)

low_ts <- quantile(data[["TSS.enrichment"]]$TSS.enrichment, probs = 0.02)

data <- subset(
  x = data,
  subset = peak_region_fragments > low_prf &
    peak_region_fragments < high_prf &
    pct_reads_in_peaks > low_prr &
    blacklist_ratio < high_blr &
    nucleosome_signal < high_ns &
    TSS.enrichment > low_ts
)

#data <- RunTFIDF(data)
#data <- FindTopFeatures(data, min.cutoff = 'q0')
#data <- RunSVD(data)

return(data)
}

young <- import_atac("ATAC/GSM5723631_Young_HSC_filtered_peak_bc_matrix.h5",
  'ATAC/GSM5723631_Young_HSC_singlecell.csv.gz',
  './ATAC/GSM5723631_Young_HSC_fragments.tsv.gz')

```

```

## Computing hash

## Warning in CreateSeuratObject.Assay(counts = chrom_assay, assay = "peaks", :
## Some cells in meta.data not present in provided counts matrix.

## Warning: Keys should be one or more alphanumeric characters followed by an
## underscore, setting key from peaks to peaks_

## Extracting TSS positions

## Finding + strand cut sites

## Finding - strand cut sites

## Computing mean insertion frequency in flanking regions

## Normalizing TSS score

old <- import_atac("ATAC/GSM5723632_Aged_HSC_filtered_peak_bc_matrix.h5",
  'ATAC/GSM5723632_Aged_HSC_singlecell.csv.gz',
  './ATAC/GSM5723632_Aged_HSC_fragments.tsv.gz')

## Computing hash

## Warning in CreateSeuratObject.Assay(counts = chrom_assay, assay = "peaks", :
## Some cells in meta.data not present in provided counts matrix.

## Warning in CreateSeuratObject.Assay(counts = chrom_assay, assay = "peaks", :
## Keys should be one or more alphanumeric characters followed by an underscore,
## setting key from peaks to peaks_

## Extracting TSS positions

## Finding + strand cut sites

## Finding - strand cut sites

## Computing mean insertion frequency in flanking regions

## Normalizing TSS score

young$dataset <- "young"
old$dataset <- "old"

data <- merge(young, old)

## Warning in CheckDuplicateCellNames(object.list = objects): Some cell names are
## duplicated across objects provided. Renaming to enforce unique cell names.

```

```

##  

## Binding matrix rows  

##  

## An object of class Seurat  

## 177359 features across 7792 samples within 1 assay  

## Active assay: peaks (177359 features, 0 variable features)  

data <- FindTopFeatures(data, min.cutoff = 'q0')  

data <- RunTFIDF(data)  

## Performing TF-IDF normalization  

data <- RunSVD(data)  

## Running SVD  

## Scaling cell embeddings  

data  

## An object of class Seurat  

## 177359 features across 7792 samples within 1 assay  

## Active assay: peaks (177359 features, 177359 variable features)  

## 1 dimensional reduction calculated: lsi  

data <- RunUMAP(object = data, reduction = 'lsi', dims = 2:30)  

## 09:07:46 UMAP embedding parameters a = 0.9922 b = 1.112  

## 09:07:46 Read 7792 rows and found 29 numeric columns  

## 09:07:46 Using Annoy for neighbor search, n_neighbors = 30  

## 09:07:46 Building Annoy index with metric = cosine, n_trees = 50  

## 0%   10    20    30    40    50    60    70    80    90    100%  

## [----|----|----|----|----|----|----|----|----|----|

```

```

## ****|  

## 09:07:48 Writing NN index file to temp file /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpSR  

## 09:07:48 Searching Annoy index using 1 thread, search_k = 3000  

## 09:07:50 Annoy recall = 100%  

## 09:08:07 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30  

## 09:08:16 Initializing from normalized Laplacian + noise (using irlba)  

## 09:08:17 Commencing optimization for 500 epochs, with 295494 positive edges  

## 09:08:33 Optimization finished

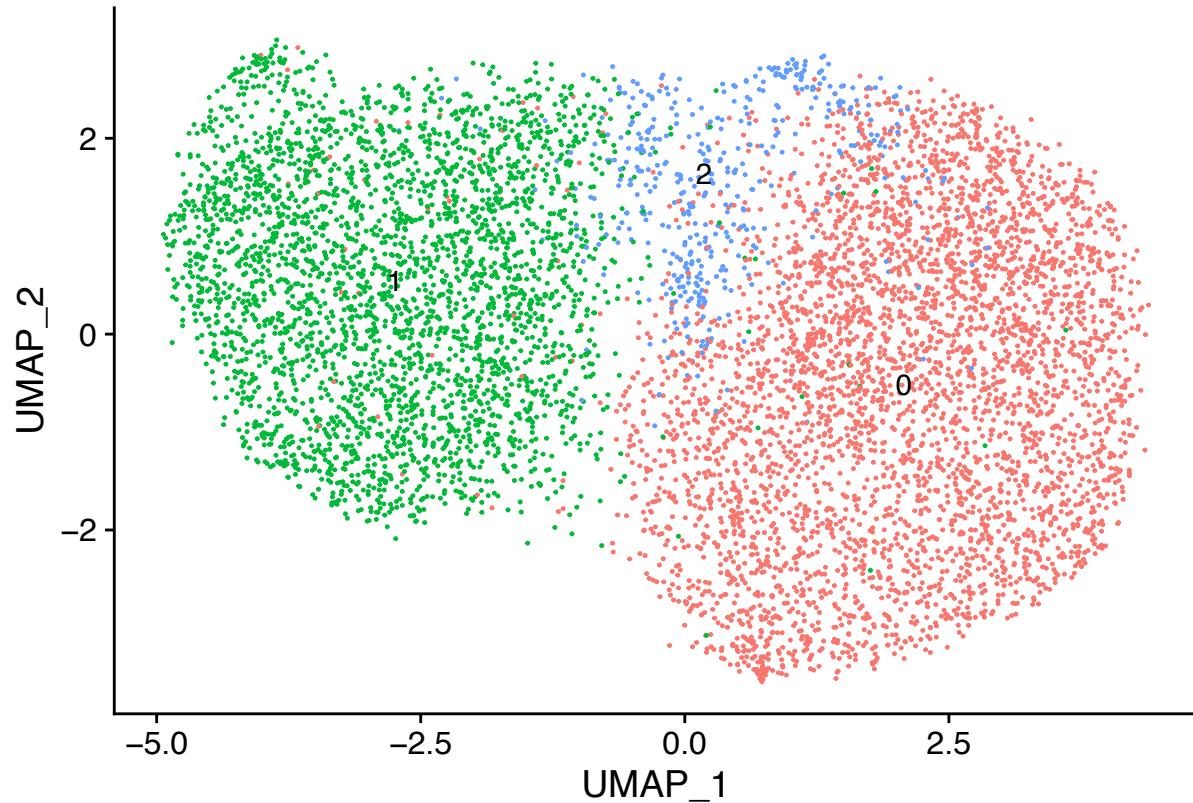
data <- FindNeighbors(object = data, reduction = 'lsi', dims = 2:30)

## Computing nearest neighbor graph
## Computing SNN

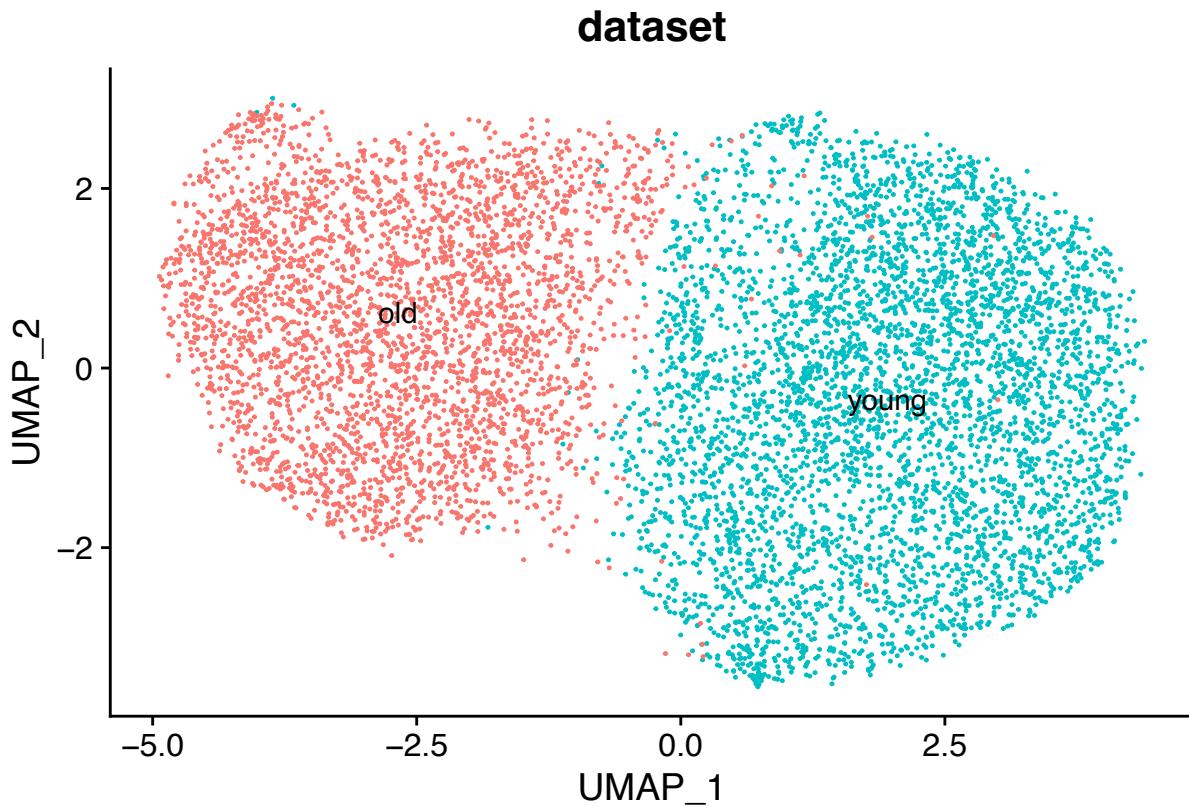
data <- FindClusters(object = data, verbose = FALSE, algorithm = 3, resolution = .4)

DimPlot(object = data, label = TRUE) + NoLegend()

```



```
DimPlot(object = data, label = TRUE, group_by = "dataset") + NoLegend()
```



Data analysis

```

gene.activities <- GeneActivity(data)

## Extracting gene coordinates

## Extracting reads overlapping genomic regions
## Extracting reads overlapping genomic regions

data[['RNA']] <- CreateAssayObject(counts = gene.activities)

data <- NormalizeData(
  object = data,
  assay = 'RNA',
  normalization.method = 'LogNormalize',
  scale.factor = median(data$nCount_RNA)
)

data[['RNA']]

## Assay data with 21808 features for 7792 cells
## First 10 features:
## Hnf4g, Zfhx4, Pex2, UBC, 1700008P02Rik, Pkia, Zc2hc1a, Il17,
## 1700010I02Rik, Stmn2

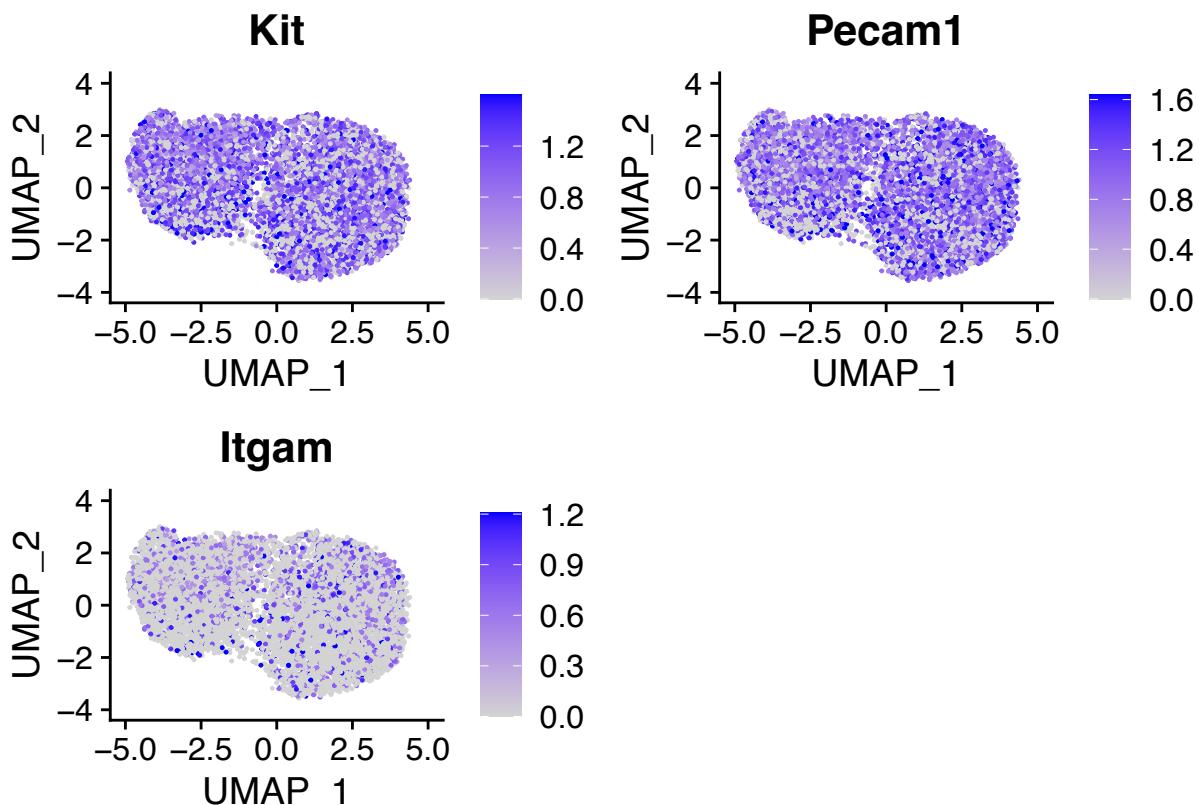
```

```

DefaultAssay(data) <- 'RNA'

FeaturePlot(
  object = data,
  features = c('Kit', 'Pecam1', 'Itgam'),
  max.cutoff = 'q95'
)

```



```

DefaultAssay(data) <- 'peaks'

da_peaks <- FindMarkers(
  object = data,
  ident.1 = rownames(data)[data$dataset == "old",],
  ident.2 = rownames(data)[data$dataset == "young",],
  min.pct = 0.05,
  test.use = 'LR',
  latent.vars = 'peak_region.fragments'
)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

da_peaks

```

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
## chr2-107231278-107232435	1.094643e-134	11.4364859	0.112	0.000	1.941448e-129

## chr15-42552345-42559570	2.749120e-01	0.2778099	0.061	0.047	1.000000e+00
## chr7-66954652-66956694	2.756335e-01	0.3167454	0.051	0.039	1.000000e+00
## chr12-12171398-12172937	2.758496e-01	0.2906140	0.053	0.041	1.000000e+00
## chr10-62049514-62050787	2.759347e-01	0.2573447	0.058	0.046	1.000000e+00
## chr10-42944043-42945019	2.759548e-01	0.2810724	0.050	0.039	1.000000e+00
## chr8-35764874-35766750	2.774231e-01	0.2598499	0.059	0.047	1.000000e+00
## chr1-7146158-7147685	2.775826e-01	0.3178811	0.051	0.038	1.000000e+00
## chr7-99864800-99866726	2.776475e-01	0.2686872	0.056	0.044	1.000000e+00
## chr14-119260108-119261223	2.783021e-01	0.2560069	0.051	0.042	1.000000e+00
## chr19-27123535-27124917	2.793634e-01	0.2686907	0.057	0.045	1.000000e+00
## chr5-139748090-139751704	2.798919e-01	0.2611308	0.065	0.050	1.000000e+00
## chr2-101600011-101601796	2.804176e-01	0.3063705	0.051	0.039	1.000000e+00
## chr6-8734988-8736794	2.814348e-01	0.3094166	0.051	0.039	1.000000e+00
## chr13-107325593-107326477	2.820075e-01	0.2863083	0.059	0.045	1.000000e+00
## chr13-51885648-51888129	2.841647e-01	0.2797441	0.050	0.039	1.000000e+00
## chr3-27369924-27372749	2.843121e-01	0.2979048	0.056	0.043	1.000000e+00
## chr16-90124758-90127407	2.852964e-01	0.2585911	0.051	0.040	1.000000e+00
## chr6-100501037-100503972	2.878537e-01	0.2600198	0.050	0.039	1.000000e+00
## chrX-20816547-20817892	2.889863e-01	0.2512107	0.060	0.049	1.000000e+00
## chr7-43880973-43884212	2.919117e-01	0.2569148	0.062	0.049	1.000000e+00
## chr17-80165486-80166393	2.924126e-01	0.2599583	0.054	0.043	1.000000e+00
## chr4-47521717-47522812	2.926588e-01	0.2776718	0.055	0.042	1.000000e+00
## chr9-37175625-37177697	2.927055e-01	0.2732671	0.061	0.047	1.000000e+00
## chr4-94900553-94902337	2.927963e-01	0.2627356	0.054	0.041	1.000000e+00
## chr9-63644894-63649084	2.934874e-01	0.2584338	0.050	0.039	1.000000e+00
## chr8-111543112-111548829	2.938704e-01	0.2507536	0.074	0.057	1.000000e+00
## chr12-87100889-87105369	2.942545e-01	0.2612185	0.062	0.049	1.000000e+00
## chr18-15344767-15347776	2.943885e-01	0.2774468	0.079	0.061	1.000000e+00
## chr13-21262754-21265333	2.958956e-01	0.2506192	0.051	0.041	1.000000e+00
## chr14-59340809-59342025	2.959527e-01	0.2631900	0.053	0.042	1.000000e+00
## chr4-117002239-117003415	2.968079e-01	0.2515823	0.056	0.046	1.000000e+00
## chr8-12735009-12737228	2.990732e-01	0.2814398	0.055	0.042	1.000000e+00
## chr15-27788284-27790742	2.993869e-01	0.2799894	0.054	0.042	1.000000e+00
## chr18-61012292-61015624	2.994598e-01	0.2540256	0.065	0.051	1.000000e+00
## chr12-33182253-33184004	3.009582e-01	0.2557808	0.063	0.049	1.000000e+00
## chr5-138857009-138857882	3.047333e-01	0.2521291	0.051	0.041	1.000000e+00
## chr3-38257067-38258774	3.049921e-01	0.2808895	0.057	0.044	1.000000e+00
## chr18-43206732-43208412	3.055809e-01	0.2725891	0.057	0.044	1.000000e+00
## chr19-53773190-53775431	3.059185e-01	0.2744288	0.053	0.042	1.000000e+00
## chr6-55789387-55790757	3.071402e-01	0.2570012	0.053	0.042	1.000000e+00
## chrX-102074332-102075703	3.084524e-01	0.2584039	0.050	0.039	1.000000e+00
## chr10-21099322-21101582	3.133837e-01	0.2544647	0.058	0.046	1.000000e+00
## chr12-112000637-112001792	3.145824e-01	0.2651784	0.050	0.039	1.000000e+00
## chr10-31329611-31331993	3.183533e-01	0.2695838	0.051	0.041	1.000000e+00
## chr17-37196375-37198383	3.184551e-01	0.2575313	0.085	0.066	1.000000e+00
## chr9-58466944-58468147	3.187634e-01	0.2732076	0.051	0.039	1.000000e+00
## chr5-121947142-121949877	3.189587e-01	0.2723385	0.051	0.040	1.000000e+00
## chr12-79049947-79051480	3.198327e-01	0.2606247	0.055	0.043	1.000000e+00
## chr8-8333886-8335694	3.205719e-01	0.2821504	0.053	0.041	1.000000e+00
## chr4-154105014-154106246	3.206332e-01	0.2509180	0.062	0.049	1.000000e+00
## chr15-99259099-99260791	3.206730e-01	0.2670246	0.059	0.046	1.000000e+00
## chr6-91280795-91281893	3.282972e-01	0.2503077	0.052	0.042	1.000000e+00
## chr8-105464161-105465728	3.297473e-01	0.2685380	0.055	0.043	1.000000e+00
## chr12-39665365-39666083	3.325996e-01	0.2560305	0.053	0.043	1.000000e+00

## chr3-11456184-11457542	3.346766e-01	0.2626706	0.051	0.040	1.000000e+00
## chr12-108326550-108329581	3.347471e-01	0.2782291	0.062	0.048	1.000000e+00
## chr17-78454196-78455744	3.348459e-01	0.2731971	0.055	0.043	1.000000e+00
## chrX-144402302-144404107	3.363894e-01	0.2574629	0.051	0.040	1.000000e+00
## chr8-22565233-22566957	3.378128e-01	0.2596510	0.052	0.041	1.000000e+00
## chr11-96535764-96537610	3.390139e-01	0.2509868	0.055	0.043	1.000000e+00
## chr3-102365994-102367056	3.393981e-01	0.2585304	0.057	0.045	1.000000e+00
## chr17-26511362-26514738	3.426090e-01	0.2583869	0.051	0.040	1.000000e+00
## chr18-70100869-70102635	3.434534e-01	0.2925942	0.058	0.045	1.000000e+00
## chr4-97673429-97675079	3.444001e-01	0.2525241	0.053	0.043	1.000000e+00
## chr7-125522127-125523994	3.488605e-01	0.2660844	0.051	0.039	1.000000e+00
## chr8-122750781-122753323	3.516711e-01	0.2503590	0.062	0.048	1.000000e+00
## chr8-126698464-126700253	3.528003e-01	0.2506856	0.060	0.048	1.000000e+00
## chr14-96961817-96963629	3.577643e-01	0.2518460	0.053	0.043	1.000000e+00
## chr5-118649190-118653881	3.722990e-01	0.2516280	0.070	0.056	1.000000e+00
## chr3-122634204-122636017	3.726035e-01	0.2575202	0.056	0.046	1.000000e+00
## chr8-12278784-12280440	3.759894e-01	0.2593295	0.057	0.044	1.000000e+00
## chr7-140057313-140059317	3.798408e-01	0.2734907	0.052	0.041	1.000000e+00
## chr16-31908424-31911509	3.830976e-01	0.2751210	0.058	0.046	1.000000e+00
## chr4-3503229-3506134	3.834660e-01	0.2558751	0.052	0.041	1.000000e+00
## chr2-17363627-17366172	3.873807e-01	0.2741811	0.050	0.038	1.000000e+00
## chr11-34395466-34396947	4.262795e-01	0.2583720	0.051	0.039	1.000000e+00
## chr9-99080682-99082185	4.684080e-01	0.2862183	0.056	0.043	1.000000e+00
##		closest_gene	distance		
		Kcna4	58203		
		Ehd1	0		
		Wars2	26045		
		Cntnap5c	232871		
		Wwtr1	0		
		Sst	43979		
		Rbfox3	0		
		Gm29327	23863		
		Tmem132c	0		
		Aldh1a1	0		
		Arhgap26	0		
	## chr6-81737474-81739083	1700009C05Rik	161380		
	## chr14-18460174-18464129	Ube2e2	109445		
	## chr1-15677637-15680698	Kcnb2	0		
	## chr2-160224927-160226517	Gm826	84875		
	## chr5-26903642-26905903	Dpp6	0		
	## chr9-30313064-30315735	Snx19	111593		
	## chr13-69592861-69597041	Srd5a1	0		
	## chrX-83584530-83604474	Dmd	0		
	## chr11-30997947-31002306	Asb3	0		
	## chr18-84203116-84205110	Zfp407	0		
	## chr4-124684960-124688440	Utp11l	0		
	## chr1-51287923-51293408	Sdpr	0		
	## chr10-86876096-86880365	Stab2	0		
	## chr10-69611392-69612675	Ank3	0		
	## chr3-57554145-57558727	Wwtr1	0		
	## chr11-66166898-66168976	Dnah9	0		
	## chr15-101397948-101398984	Krt7	12058		
	## chr3-67601361-67602314	Mfsd1	0		
	## chr13-48442459-48445612	Zfp169	42034		

## chr15-41823894-41826428	Oxr1	0
## chr7-72728445-72729718	Gm28744	60088
## chr16-6320518-6322339	Rbfox1	486882
## chr14-73086275-73096470	Rcbtb2	26566
## chr13-79442519-79443841	Gm29318	144276
## chr19-6290930-6294154	Ehd1	0
## chr10-74147864-74148629	Pcdh15	0
## chr7-63315796-63317114	4930554H23Rik	27575
## chr7-49320361-49322630	Nav2	0
## chr10-52132018-52134641	Ros1	0
## chr17-63682575-63686265	A930002H24Rik	177034
## chr14-105588234-105589331	4930449E01Rik	82604
## chr8-125839578-125841247	Pcnxl2	0
## chr1-15285819-15288023	Kcnb2	0
## chr13-100317251-100318129	Naip6	0
## chr10-73096310-73102125	Pcdh15	0
## chr2-144186497-144189529	Gm5535	0
## chr18-45557793-45561592	Kcnn2	0
## chr6-3182174-3184431	Samd9l	187825
## chr5-125583041-125584699	Tmem132b	0
## chr3-57515769-57518214	Wwtr1	0
## chr5-120185385-120187452	Rbm19	0
## chr16-29981839-29984284	Gm1968	19531
## chr8-107542962-107548550	Wwp2	0
## chr19-6333857-6336770	Men1	0
## chr17-64715403-64717192	Man2a1	0
## chr3-57293821-57296465	Tm4sf1	0
## chr19-28769087-28771222	D930032P07Rik	49059
## chr7-47345476-47347189	Mrgpra1	0
## chr5-25846317-25850648	Actr3b	0
## chr6-107494034-107495804	Lrrn1	33921
## chr12-114561879-114565165	RP23-354D10.5	697518
## chr4-120611767-120616637	Gm12860	0
## chr1-114038945-114040858	9330185C12Rik	74810
## chr4-123791849-123795578	Rhbdl2	0
## chr6-126158932-126168436	Ntf3	0
## chr8-10138908-10140217	Tnfsf13b	103466
## chr8-61257630-61260758	Sh3rf1	0
## chr7-119525970-119528157	Acsm5	0
## chr10-92325101-92326056	Gm20757	0
## chr13-10356935-10361955	Chrm3	0
## chr5-113775874-113777556	Iscu	0
## chr13-48316403-48318189	A330048009Rik	42518
## chr14-53132284-53135800	Trav9n-1	29882
## chr12-103213777-103215351	Prima1	0
## chr1-77280560-77282518	Epha4	84666
## chr3-129509435-129510758	Elov16	21627
## chr1-144852825-144854089	Rgs18	77389
## chr14-121100817-121103284	Farp1	0
## chr15-44175698-44176493	Trhr	19641
## chr3-145209407-145211172	Odf2l	55491
## chr9-12495705-12497706	Gm25365	282458
## chr15-75719818-75721867	Rhpn1	5407
## chr16-77704149-77705336	Mir99ahg	12745

## chr15-72985742-72989047	Trappc9	0
## chr2-91974467-91977267	Dgkz	0
## chr13-43234776-43238338	Gfod1	0
## chr13-114895630-114897515	Itga2	0
## chr3-27858530-27859990	Tmem212	6075
## chr1-60911043-60912370	Ctla4	0
## chr3-41415966-41418044	C430002E04Rik	70759
## chr1-183319350-183321705	Aida	0
## chr15-25523570-25526172	Myo10	96352
## chr12-110973515-110976526	Ankrd9	0
## chr15-42552345-42559570	Angpt1	0
## chr7-66954652-66956694	Adamts17	0
## chr12-12171398-12172937	Gm26171	64530
## chr10-62049514-62050787	Gm5424	20227
## chr10-42944043-42945019	Scml4	0
## chr8-35764874-35766750	Gm22216	17311
## chr1-7146158-7147685	Pcmtd1	0
## chr7-99864800-99866726	Xrra1	0
## chr14-119260108-119261223	Hs6st3	0
## chr19-27123535-27124917	Vldlr	91566
## chr5-139748090-139751704	Ints1	0
## chr2-101600011-101601796	B230118H07Rik	0
## chr6-8734988-8736794	Ica1	0
## chr13-107325593-107326477	Apoo-ps	87387
## chr13-51885648-51888129	Gadd45g	37179
## chr3-27369924-27372749	Ghsr	0
## chr16-90124758-90127407	Sod1	93334
## chr6-100501037-100503972	1700049E22Rik	23427
## chrX-20816547-20817892	Araf	0
## chr7-43880973-43884212	Klk4	0
## chr17-80165486-80166393	Galm	0
## chr4-47521717-47522812	Sec61b	38474
## chr9-37175625-37177697	Pknox2	28302
## chr4-94900553-94902337	Eqtn	4929
## chr9-63644894-63649084	Smad3	0
## chr8-111543112-111548829	Znrf1	0
## chr12-87100889-87105369	Ngb	0
## chr18-15344767-15347776	Aqp4	41617
## chr13-21262754-21265333	Gpx5	21095
## chr14-59340809-59342025	Phf11b	0
## chr4-117002239-117003415	Hectd3	0
## chr8-12735009-12737228	Gm15348	15881
## chr15-27788284-27790742	Trio	0
## chr18-61012292-61015624	Slc6a7	0
## chr12-33182253-33184004	Atxn7l1	0
## chr5-138857009-138857882	Gm5294	35389
## chr3-38257067-38258774	Ankrd50	190486
## chr18-43206732-43208412	Stk32a	0
## chr19-53773190-53775431	Rbm20	0
## chr6-55789387-55790757	Ccdc129	46137
## chrX-102074332-102075703	Nhs12	0
## chr10-21099322-21101582	Ahi1	18893
## chr12-112000637-112001792	Tdrd9	0
## chr10-31329611-31331993	Tpd5211	386

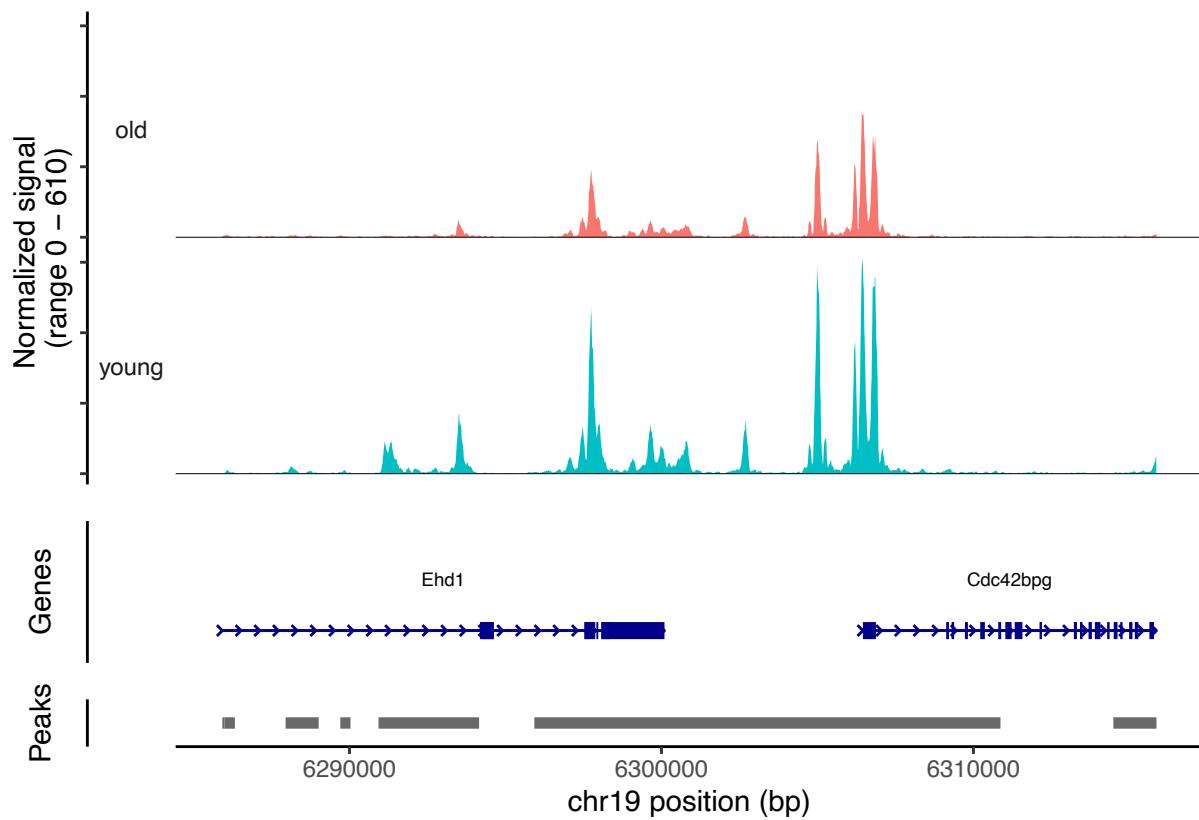
```

## chr17-37196375-37198383          Olfr94      0
## chr9-58466944-58468147          4930461G14Rik 0
## chr5-121947142-121949877       Cux2        0
## chr12-79049947-79051480       Plekhh1      0
## chr8-8333886-8335694          Efnb2       281739
## chr4-154105014-154106246       Trp73        0
## chr15-99259099-99260791      1700120C14Rik 0
## chr6-91280795-91281893       Fbln2       8254
## chr8-105464161-105465728      Lrrc36       74
## chr12-39665365-39666083       Arl4a       339363
## chr3-11456184-11457542      RP23-17107.1 365686
## chr12-108326550-108329581     Hhip11      0
## chr17-78454196-78455744      Gm10093     35820
## chrX-144402302-144404107      Trpc5        0
## chr8-22565233-22566957       Slc20a2      0
## chr11-96535764-96537610      Skap1        0
## chr3-102365994-102367056      Ngf         102871
## chr17-26511362-26514738      Dusp1       2842
## chr18-70100869-70102635      Rab27b      0
## chr4-97673429-97675079      E130114P18Rik 0
## chr7-125522127-125523994     Il4ra       28287
## chr8-122750781-122753323     Gm20388     0
## chr8-126698464-126700253     Irf2bp2     104477
## chr14-96961817-96963629      Klhl1       442714
## chr5-118649190-118653881     Med13l      0
## chr3-122634204-122636017     Fnbp11     14536
## chr8-12278784-12280440      A230072I06Rik 0
## chr7-140057313-140059317     Msx3        8223
## chr16-31908424-31911509     Mfi2        9403
## chr4-3503229-3506134         Tmem68     42906
## chr2-17363627-17366172      Neb1        0
## chr11-34395466-34396947     Fam196b      0
## chr9-99080682-99082185      Pik3cb      0

CoveragePlot(
  object = data,
  region = rownames(da_peaks)[2],
  extend.upstream = 10000,
  extend.downstream = 5000,
  group.by = "dataset"
)

## Warning: Removed 24 rows containing missing values ('geom_segment()').

```

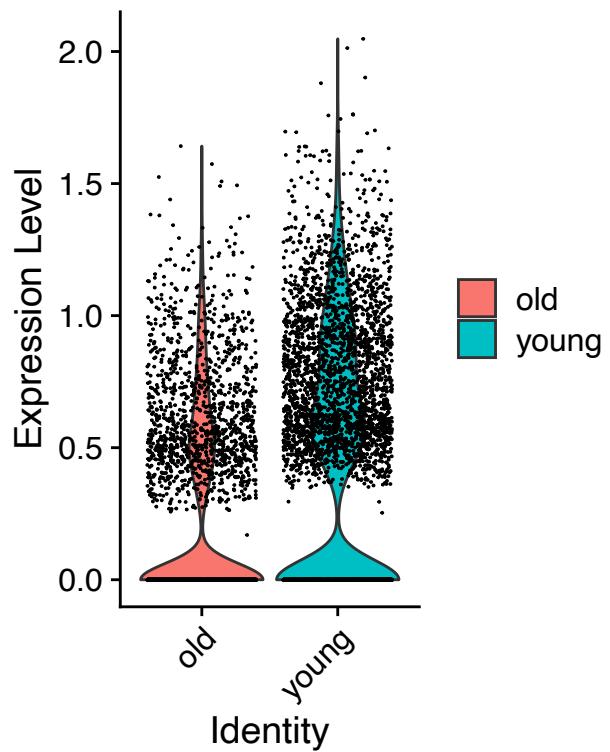


```

plot1 <- VlnPlot(
  object = data,
  features = rownames(da_peaks)[2],
  group.by = "dataset"
)
plot2 <- FeaturePlot(
  object = data,
  features = rownames(da_peaks)[2],
  max.cutoff = 'q95'
)
plot1 | plot2

```

**chr19–6295925–6310866**



**chr19–6295925–6310866**

