

SCENIC

Joan Abinet

2024-12-12 15:55:07 +0100

Contents

Load packages	1
Load Seurat objects	1
prepare cellinfo et expression matrix	2
Download binding motifs for human	2
Initialise Scenic	3
Gene filtering	3
Correlation	3
Plotting the results	4
Viewing markers based on specificity score	4
Plot on Heatmap the 10 markers the most specific of each cell type	5

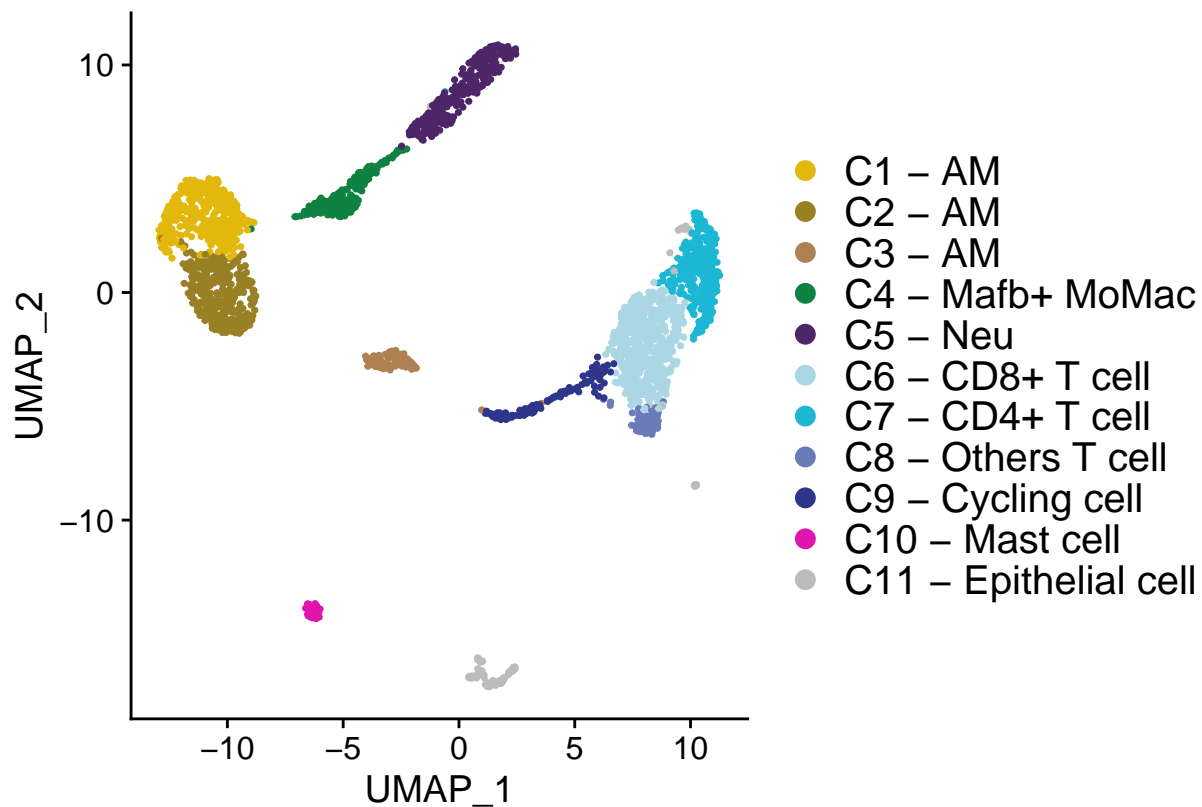
Load packages

```
suppressMessages(library(dplyr))
suppressMessages(library(Seurat))
suppressMessages(library(patchwork))
suppressMessages(library(ggplot2))
suppressMessages(library(SCENIC))
suppressMessages(library(ComplexHeatmap))
suppressMessages(library(AUCell))
```

Load Seurat objects

```
BAL_10x.integrated <- readRDS("../10x-2-Annotation_cells/BAL_10x.annotated_noDB.rds")

col <- c("#E2B80C", "#998025", "#AE8052", "#0F8140",
        "#4F2569", "#ABD6E4", "#1CB7D5", "#6B7BBA",
        "#2D368B", "#DF15AE", "#BDBCBC")
DimPlot(BAL_10x.integrated, group.by = "Annotation",
        cols = col) + theme(legend.text = element_text(size = 15)) +
  labs(title = NULL)
```



prepare cellinfo et expression matrix

```
exprMat <- BAL_10x.integrated@assays$RNA@data
cellInfo <- data.frame(seuratCluster = Idents(BAL_10x.integrated))
cellInfo$Annotation <- BAL_10x.integrated$Annotation
cellInfo$nGene <- colSums(exprMat > 0)
```

```
dir.create("int")
saveRDS(exprMat, file = "int/exprMat.Rds")
saveRDS(cellInfo, file = "int/cellInfo.Rds")
```

Download binding motifs for human

```
dbFiles <- c("https://resources.aertslab.org/cistarget/databases/old/homo_sapiens/hg19/refseq_r45/mc9nr",
             "https://resources.aertslab.org/cistarget/databases/old/homo_sapiens/hg19/refseq_r45/mc9nr/gene_base")

dir.create("cisTarget_databases")
setwd("cisTarget_databases")
for (featherURL in dbFiles) {
  download.file(featherURL, destfile = basename(featherURL)) # saved in current dir
}
```

Initialise Scenic

```
data(list = "motifAnnotations_hgnc_v9", package = "RcisTarget")
motifAnnotations_hgnc <- motifAnnotations_hgnc_v9

org <- "hgnc"
dbDir <- "cisTarget_databases"
dbDir <- path.expand(dbDir)
myDatasetTitle <- "SCENIC Analysis"
data(defaultDbNames)
dbs <- defaultDbNames[[org]]
scenicOptions <- initializeScenic(org = org, dbDir = dbDir,
  dbs = dbs, nCores = 25)

## Motif databases selected:
##   hg19-500bp-upstream-7species.mc9nr.feather
##   hg19-tss-centered-10kb-7species.mc9nr.feather

## Using the column 'features' as feature index for the ranking database.
## Using the column 'features' as feature index for the ranking database.
```

Gene filtering

1. Filter by the total number of reads per gene. Keeps only the genes with at least 6 UMI counts across all samples.
2. Filter by the number of cells in which the gene is detected.

```
exprMat <- as.matrix(exprMat) # need a regulat matrix
genesKept <- geneFiltering(exprMat, scenicOptions = scenicOptions,
  minCountsPerGene = 3 * 0.01 * ncol(exprMat),
  minSamples = ncol(exprMat) * 0.01)

dim(exprMat) # 14386
exprMat_filtered <- exprMat[genesKept, ]
dim(exprMat_filtered) # 9389
```

Correlation

```
runCorrelation(exprMat_filtered, scenicOptions)

runGenie3(exprMat_filtered, scenicOptions)

scenicOptions@settings$verbose <- TRUE
scenicOptions@settings$nCores <- 10
scenicOptions@settings$seed <- 123

runSCENIC_1_coexNetwork2modules(scenicOptions)
runSCENIC_2_createRegulons(scenicOptions)
runSCENIC_3_scoreCells(scenicOptions, exprMat_filtered)
```

Plotting the results

Viewing markers based on specificity score

```
regulonAUC <- loadInt(scenicOptions, "aucell_regulonAUC") # require file int/3.4_regulonAUC.Rds

rss <- calcRSS(AUC = getAUC(regulonAUC), cellAnnotation = cellInfo[colnames(regulonAUC),
  "seuratCluster"])

## Loading required package: BiocParallel

rssPlot <- plotRSS(rss)

## Showing regulons and cell types with any RSS > 0.01 (dim: 175x11)

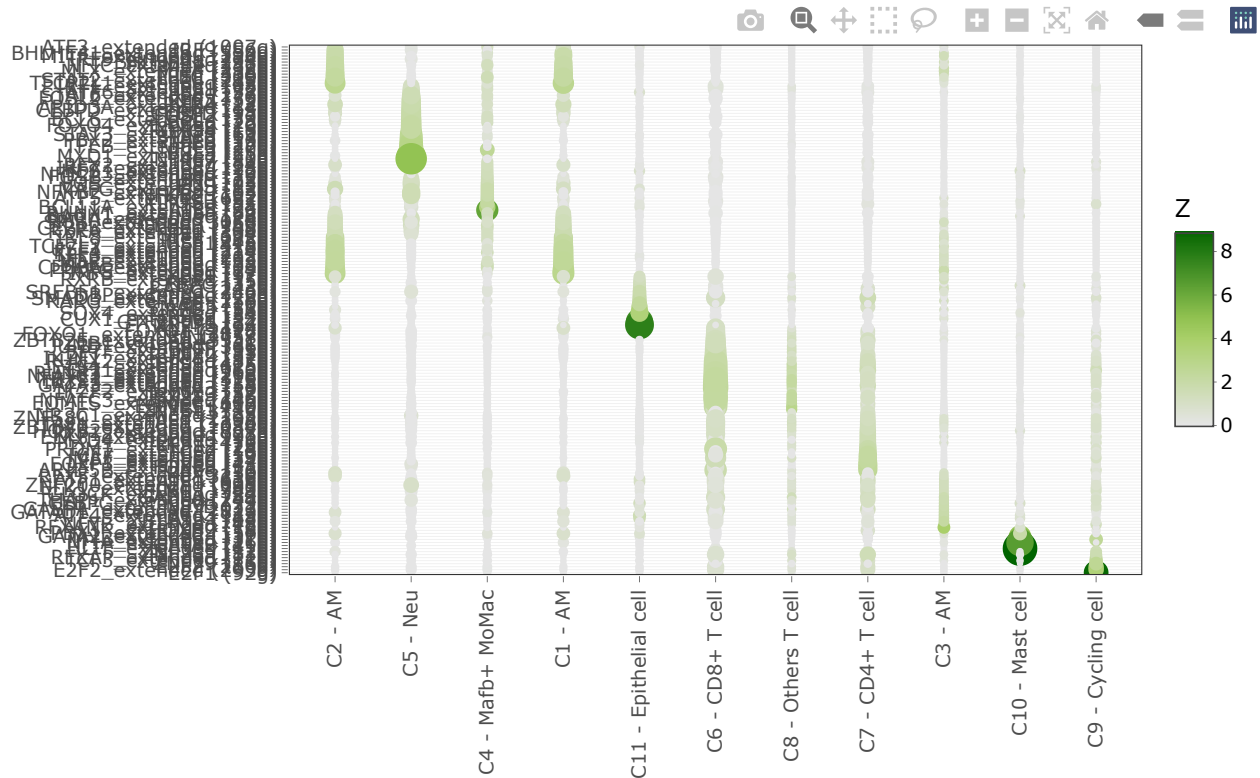
## The automatically generated colors map from the 1st and 99th of the
## values in the matrix. There are outliers in the matrix whose patterns
## might be hidden by this color mapping. You can manually set the color
## to `col` argument.
##
## Use `suppressMessages()` to turn off this message.

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##   between, first, last

plotly::ggplotly(rssPlot$plot)
```



Plot on Heatmap the 10 markers the most specific of each cell type

```
rss <- rss[, c(4, 1, 9, 3, 2, 6, 8, 7, 11, 10,
5)]
# rss <-
# rss[,c(1,6,8,3,7,2,13,12,5,4,9,11,10)]

top_tf <- c()
for (i in 1:length(colnames(rss))) {
  TFs <- sort(rss[, i], decreasing = T)
  top_10 <- head(TFs, 10)
  top_tf <- c(top_tf, top_10)
}

length(top_tf)

# top_tf

# Creation of the Heatmap

regulonAUC.mat <- regulonAUC@assays@data@listData$AUC
Subset_regulonActivity <- regulonAUC.mat[names(top_tf),
]

regulonActivity_byCellType_Scaled <- t(scale(t(Subset_regulonActivity),
center = T, scale = T))
```

```

colors <- c("#E2B80C", "#998025", "#AE8052", "#0F8140",
            "#4F2569", "#ABD6E4", "#1CB7D5", "#6B7BBA",
            "#2D368B", "#DF15AE", "#BDBCBC")

names(colors) <- levels(BAL_10x.integrated$Annotation)

df <- as.data.frame(BAL_10x.integrated$Annotation)
colnames(df) <- "CellType"

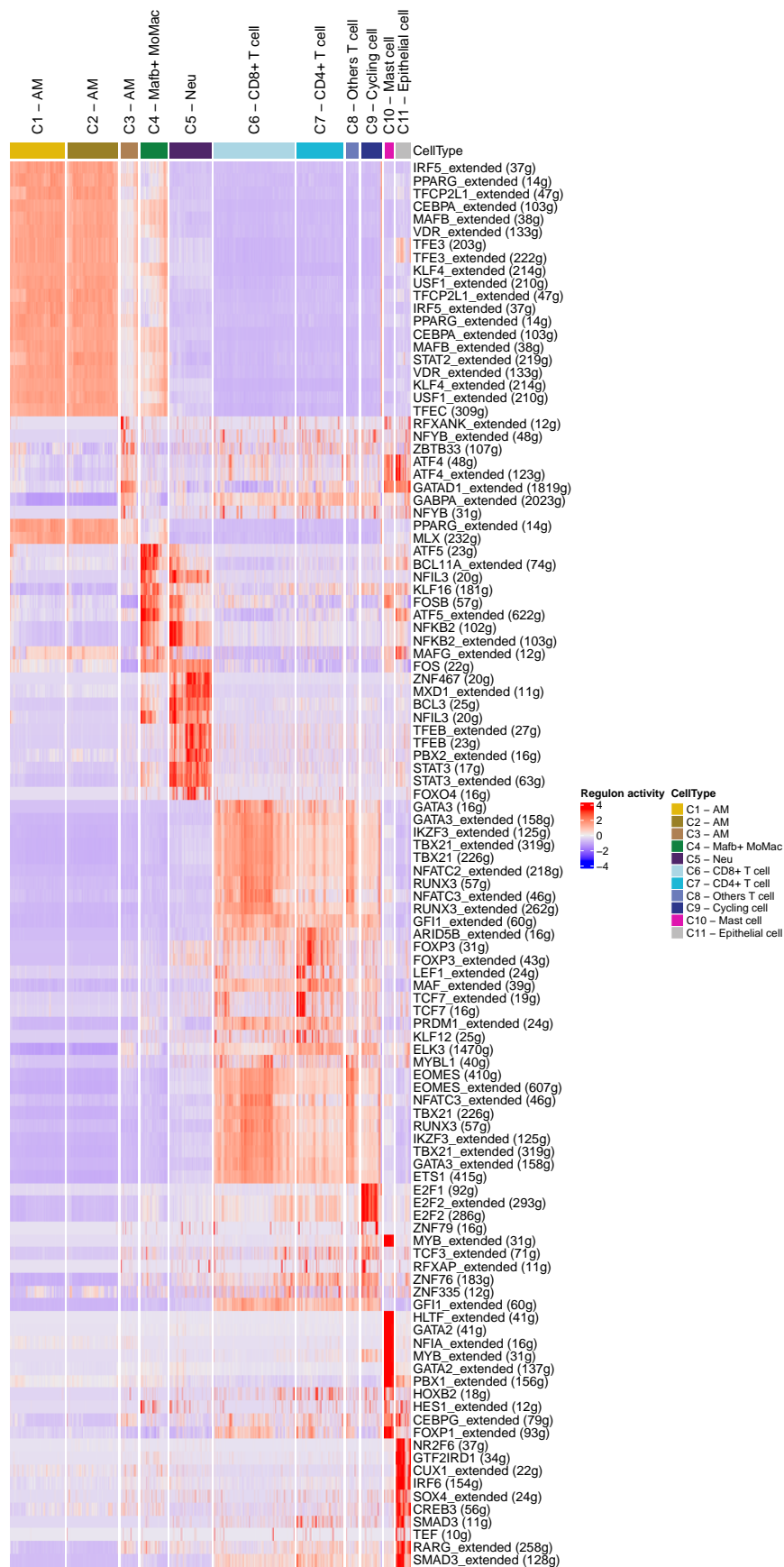
color_df <- list(CellType = c(`C1 - AM` = "#E2B80C",
`C2 - AM` = "#998025", `C3 - AM` = "#AE8052",
`C4 - Mafb+ MoMac` = "#0F8140", `C5 - Neu` = "#4F2569",
`C6 - CD8+ T cell` = "#ABD6E4", `C7 - CD4+ T cell` = "#1CB7D5",
`C8 - Others T cell` = "#6B7BBA", `C9 - Cycling cell` = "#2D368B",
`C10 - Mast cell` = "#DF15AE", `C11 - Epithelial cell` = "#BDBCBC"))

# est-ce que je devrais pas inclure que les
# normaux ou les extended. Merger les 2 est
# bizarre.
Heatmap <- Heatmap(regulonActivity_byCellType_Scaled,
  name = "Regulon activity", show_column_names = FALSE,
  column_split = factor(BAL_10x.integrated$Annotation),
  cluster_column_slices = F, cluster_rows = F,
  top_annotation = HeatmapAnnotation(df = df,
    col = color_df), show_column_dend = F,
  column_title_rot = 90)

# tidyHeatmap::save_pdf(Heatmap, 'Scenic_10x.pdf',
# width = 40, height = 50, units = 'cm')

Heatmap

```



sessionInfo()

```
## R version 4.3.3 (2024-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p0.3.20.so; LAPACK version 3.10.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=fr_BE.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=fr_BE.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=fr_BE.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_BE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Brussels
## tzcode source: system (glibc)
##
## attached base packages:
## [1] grid      stats      graphics  grDevices utils      datasets  methods
## [8] base
##
## other attached packages:
##  [1] data.table_1.14.8      BiocParallel_1.34.2    RcisTarget_1.20.0
##  [4] AUCell_1.22.0          ComplexHeatmap_2.16.0 SCENIC_1.3.1
##  [7] ggplot2_3.4.2          patchwork_1.1.2        SeuratObject_4.1.3
## [10] Seurat_4.3.0           dplyr_1.1.2
##
## loaded via a namespace (and not attached):
##  [1] RcppAnnoy_0.0.21      splines_4.3.3
##  [3] later_1.3.1           bitops_1.0-7
##  [5] tibble_3.2.1          R.oo_1.25.0
##  [7] polyclip_1.10-4       graph_1.78.0
##  [9] XML_3.99-0.14         lifecycle_1.0.3
## [11] doParallel_1.0.17     processx_3.8.1
## [13] globals_0.16.2        lattice_0.22-5
## [15] MASS_7.3-60.0.1       crosstalk_1.2.0
## [17] magrittr_2.0.3         plotly_4.10.2
## [19] rmarkdown_2.23        yaml_2.3.7
## [21] httpuv_1.6.11         sctransform_0.3.5
## [23] spam_2.9-1            sp_2.0-0
## [25] spatstat.sparse_3.0-2 reticulate_1.30
## [27] chromote_0.1.2         cowplot_1.1.1
## [29] pbapply_1.7-2          DBI_1.1.3
## [31] RColorBrewer_1.1-3     abind_1.4-5
## [33] zlibbioc_1.46.0        Rtsne_0.16
## [35] GenomicRanges_1.52.0   purrr_1.0.1
## [37] R.utils_2.12.2         BiocGenerics_0.46.0
## [39] RCurl_1.98-1.12       webshot2_0.1.1
## [41] circlize_0.4.15        GenomeInfoDbData_1.2.10
## [43] IRanges_2.34.0         S4Vectors_0.38.1
```


## [45]	ggrepel_0.9.3	irlba_2.3.5.1
## [47]	listenv_0.9.0	spatstat.utils_3.0-3
## [49]	goftest_1.2-3	spatstat.random_3.1-5
## [51]	annotate_1.78.0	fitdistrplus_1.1-11
## [53]	parallelly_1.36.0	DelayedMatrixStats_1.22.1
## [55]	leiden_0.4.3	codetools_0.2-19
## [57]	DelayedArray_0.26.3	shape_1.4.6
## [59]	tidyselect_1.2.0	farver_2.1.1
## [61]	matrixStats_1.0.0	stats4_4.3.3
## [63]	spatstat.explore_3.2-1	jsonlite_1.8.7
## [65]	GetoptLong_1.0.5	ellipsis_0.3.2
## [67]	progressr_0.13.0	iterators_1.0.14
## [69]	gggridges_0.5.4	survival_3.5-8
## [71]	foreach_1.5.2	tools_4.3.3
## [73]	ica_1.0-3	Rcpp_1.0.11
## [75]	glue_1.6.2	gridExtra_2.3
## [77]	xfun_0.39	MatrixGenerics_1.12.2
## [79]	websocket_1.4.1	GenomeInfoDb_1.36.0
## [81]	withr_2.5.0	formatR_1.14
## [83]	fastmap_1.1.1	fansi_1.0.4
## [85]	digest_0.6.33	R6_2.5.1
## [87]	mime_0.12	colorspace_2.1-0
## [89]	Cairo_1.6-2	scattermore_1.2
## [91]	tensor_1.5	spatstat.data_3.0-1
## [93]	RSQlite_2.3.1	R.methodsS3_1.8.2
## [95]	utf8_1.2.3	tidyr_1.3.0
## [97]	generics_0.1.3	httr_1.4.6
## [99]	htmlwidgets_1.6.2	S4Arrays_1.2.1
## [101]	uwot_0.1.16	pkgconfig_2.0.3
## [103]	gtable_0.3.3	blob_1.2.4
## [105]	lmtest_0.9-40	XVector_0.40.0
## [107]	htmltools_0.5.5	dotCall64_1.0-2
## [109]	clue_0.3-64	GSEABase_1.62.0
## [111]	scales_1.2.1	Biobase_2.60.0
## [113]	png_0.1-8	knitr_1.43
## [115]	rstudioapi_0.14	rjson_0.2.21
## [117]	reshape2_1.4.4	nlme_3.1-164
## [119]	GlobalOptions_0.1.2	zoo_1.8-12
## [121]	cachem_1.0.8	stringr_1.5.0
## [123]	KernSmooth_2.23-22	parallel_4.3.3
## [125]	miniUI_0.1.1.1	arrow_12.0.1
## [127]	AnnotationDbi_1.62.1	pillar_1.9.0
## [129]	vctrs_0.6.3	RANN_2.6.1
## [131]	promises_1.2.0.1	xtable_1.8-4
## [133]	cluster_2.1.6	evaluate_0.21
## [135]	magick_2.7.5	cli_3.6.1
## [137]	compiler_4.3.3	rlang_1.1.1
## [139]	crayon_1.5.2	future.apply_1.11.0
## [141]	labeling_0.4.2	ps_1.7.5
## [143]	plyr_1.8.8	stringi_1.7.12
## [145]	viridisLite_0.4.2	deldir_1.0-9
## [147]	assertthat_0.2.1	munsell_0.5.0
## [149]	Biostings_2.68.1	lazyeval_0.2.2
## [151]	spatstat.geom_3.2-4	Matrix_1.6-1

```
## [153] sparseMatrixStats_1.12.0    bit64_4.0.5
## [155] future_1.33.0                 KEGGREST_1.40.0
## [157] shiny_1.7.4.1                 highr_0.10
## [159] SummarizedExperiment_1.30.2  ROCR_1.0-11
## [161] igraph_1.5.0.1                memoise_2.0.1
## [163] bit_4.0.5
```