

# Dorothea regulon

*Xiaosai Yao*

October 4, 2022

**Package**

epiregulon 1.0.14

## Contents

1	Load regulon . . . . .	2
2	Load scRNA-seq data . . . . .	2
3	Calculate activity . . . . .	3
4	Visualize activity . . . . .	4
5	Pathway enrichment . . . . .	8
6	Session Info . . . . .	10

## Dorothea regulon

Epiregulon also supports transcription factor activity inference when users only have scRNA-seq. After all, multiome or scATAC-seq data is still relatively rare. To enable TF activity inference on scRNA-seq, users can supply a pre-constructed gene regulatory network. [Dorothea](#) provides both human and mouse pre-constructed gene regulatory networks based on curated experimental and computational data. In this vignette, we bypass the regulon construction step and go straight to calculate TF activity from a Dorothea GRN.

## 1 Load regulon

---

Dorothea assigns confidence level to its regulons with A being the most confident (i.e. supported by multiple lines of evidence) and E being the least confident. For this demo, we further trim the regulons to only 4 TFs.

```
library(dorothea)
data(dorothea_mm, package = "dorothea")
regulon <- dorothea_mm

#trim regulon
genes_to_plot <- c("Foxa1", "Neurod1", "Pdx1", "Arx")
regulon <- regulon[regulon$tf %in% genes_to_plot, ]
```

## 2 Load scRNA-seq data

---

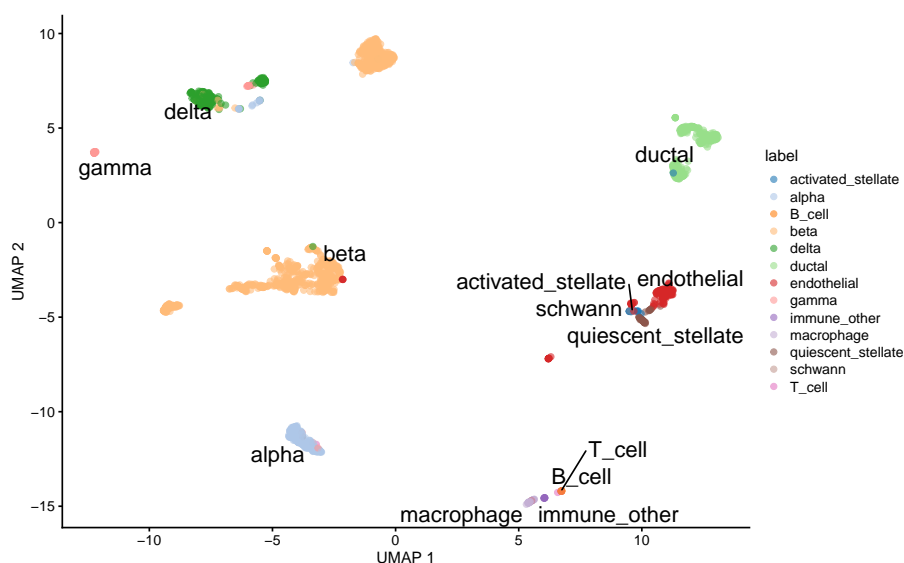
We download the raw counts of a mouse pancreas data set from [scRNAseq](#). We add normalized logcounts, perform dimension reduction and visualize the embeddings using [scater](#).

```
library(scRNAseq)
library(scater)

sce <- BaronPancreasData('mouse')
sce <- logNormCounts(sce)
sce <- runPCA(sce)
sce <- runUMAP(sce)

plotUMAP(sce, colour_by = "label", text_by = "label")
```

## Dorothea regulon



### 3 Calculate activity

Even though Dorothea provides weights under the `mor` column, we achieved superior performance if we recompute the weights based on the correlation between `tf` and target gene expression based on our own data. We performed 2 steps, the first step is to add weights to the Dorothea regulons and the second step is to estimate the TF activity by taking the weighted average of the target gene expression.

```
library(epiregulon)

#Add weights to regulon
regulon.ms <- addWeights(regulon = regulon,
                        sce = sce,
                        cluster_factor = "label",
                        BPPARAM = BiocParallel::MulticoreParam())

## calculating average expression across clusters...
## computing correlation of the regulon...
##
```

		0%
		25%
		50%
		75%
		100%

```
#Calculate activity
```

## Dorothea regulon

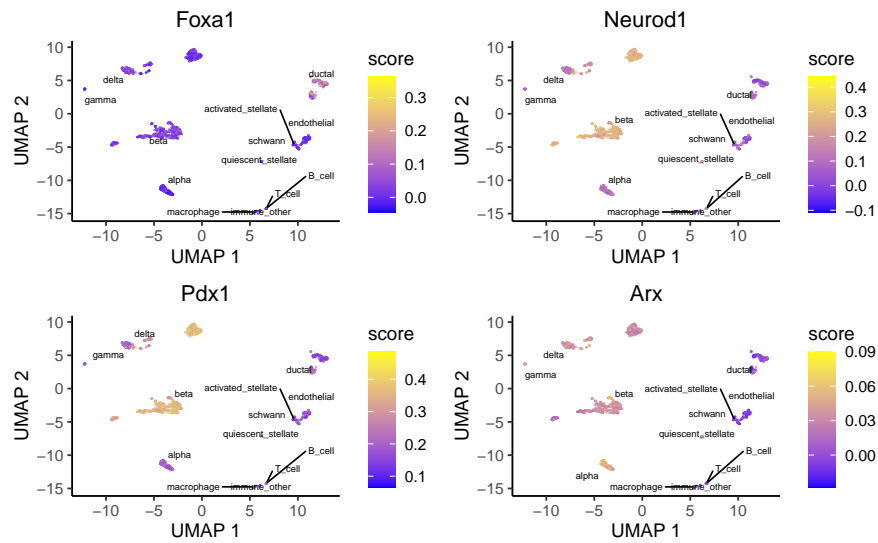
```
score.combine <- calculateActivity(sce,  
                                regulon = regulon.ms,  
                                mode = "weight",  
                                method = "weightedMean")  
  
##  
|  
|  
|=====| 0%  
|  
|=====| 25%  
|=====| 50%  
|=====| 75%  
|=====| 100%
```

## 4 Visualize activity

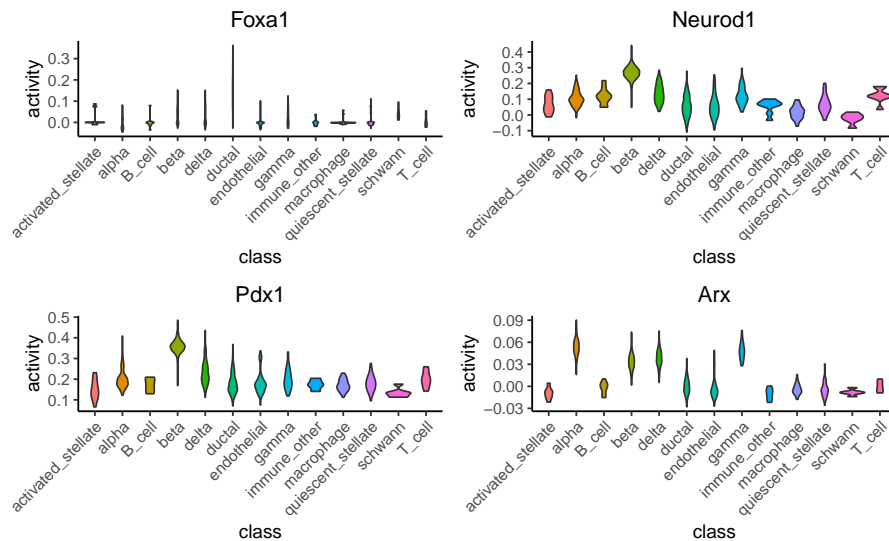
Finally we visualize the TF activity by either UMAP, violin plots or bubble plots. We confirm the activity of known lineage factors Pdx1 and Neurod1 in beta cells, Arx in alpha cells and Foxa1 in ductal cells.

```
# plot umap  
plotActivityDim(sce = sce,  
               activity_matrix = score.combine,  
               tf = genes_to_plot,  
               legend.label = "score",  
               point_size = 0.1,  
               dimtype = "UMAP",  
               label = "label",  
               combine = TRUE,  
               text_size = 2)
```

## Dorothea regulon

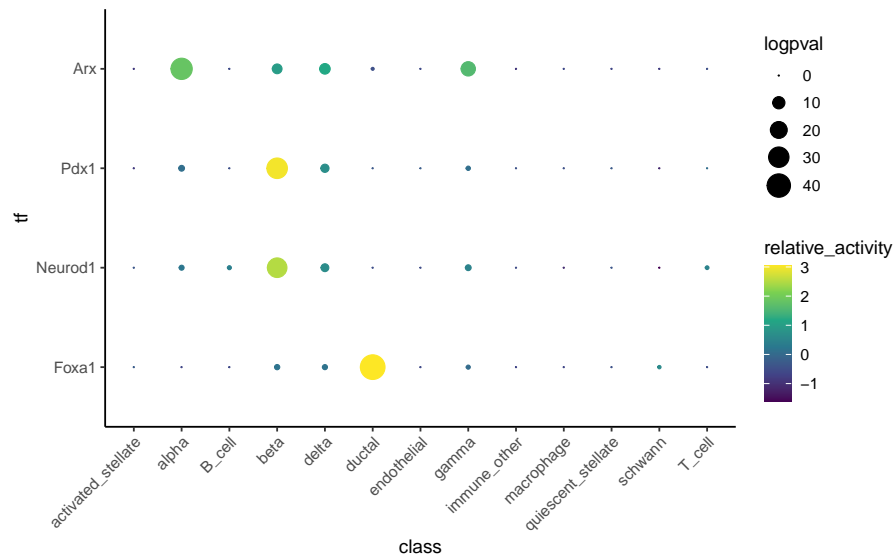


```
# plot violin plot
plotActivityViolin(score.combine,
  tf = genes_to_plot,
  class = sce$label)
```



```
# plot Bubble plot
plotBubble(score.combine,
  tf = genes_to_plot,
  class = sce$label)
```

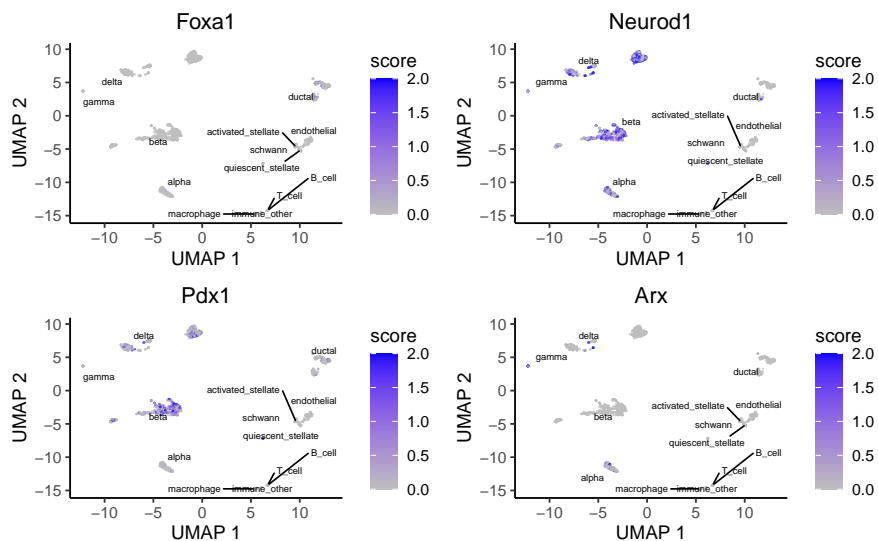
## Dorothea regulon



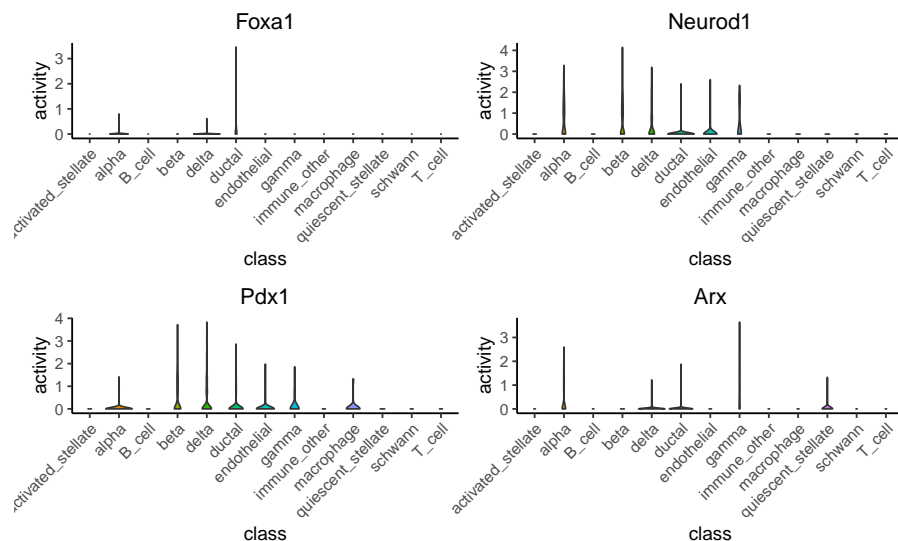
We can adapt the `epiregulon` package to plot gene expression. When compared against TF activity, gene expression of `Foxa1` and `Arx` has noisy signals and high dropout rates. `Epiregulon` enhances the signal to noise ratio of TF activity and better resolves lineage differences.

```
# plot umap
plotActivityDim(sce = sce,
  activity_matrix = logcounts(sce)[genes_to_plot,],
  tf = genes_to_plot,
  legend.label = "score",
  point_size = 0.1,
  dimtype = "UMAP",
  label = "label",
  combine = TRUE,
  text_size = 2,
  colors = c("gray", "blue"),
  limit = c(0, 2))
```

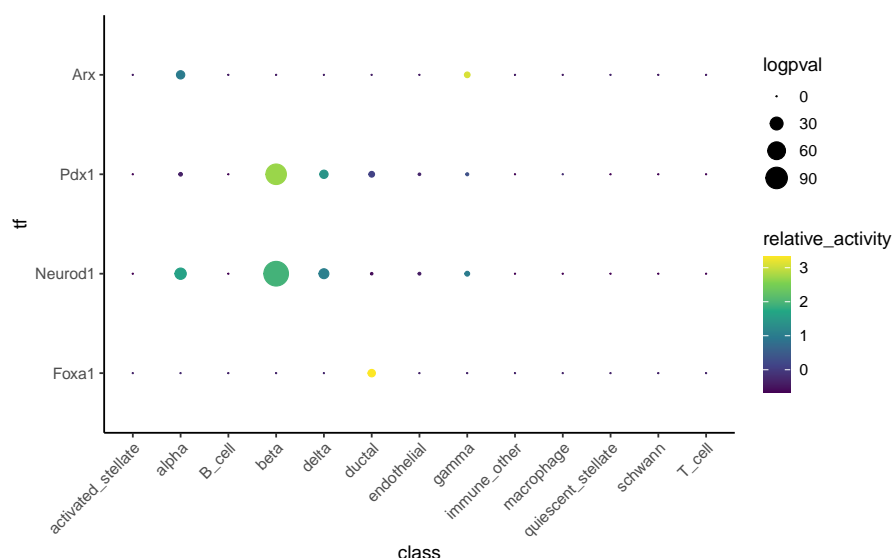
## Dorothea regulon



```
# plot violin plot
plotActivityViolin(logcounts(sce)[genes_to_plot,],
  tf = genes_to_plot,
  class = sce$label)
```



```
# plot Bubble plot
plotBubble(logcounts(sce)[genes_to_plot,],
  tf = genes_to_plot,
  class = sce$label)
```



## 5 Pathway enrichment

Sometimes it is useful to understand what pathways are enriched in the regulons. We take the highly correlated target genes of a regulon and perform geneset enrichment using the `enricher` function from [clusterProfiler](#).

```
#retrieve genesets
H <- EnrichmentBrowser::getGenesets(org = "mmu", db = "msigdb", cat = "H", gene.id.type = "SYMBOL" )
## Using cached version from 2022-10-05 20:58:12
C6 <- EnrichmentBrowser::getGenesets(org = "mmu", db = "msigdb", cat = "C6", gene.id.type = "SYMBOL" )
## Using cached version from 2022-10-05 20:58:21

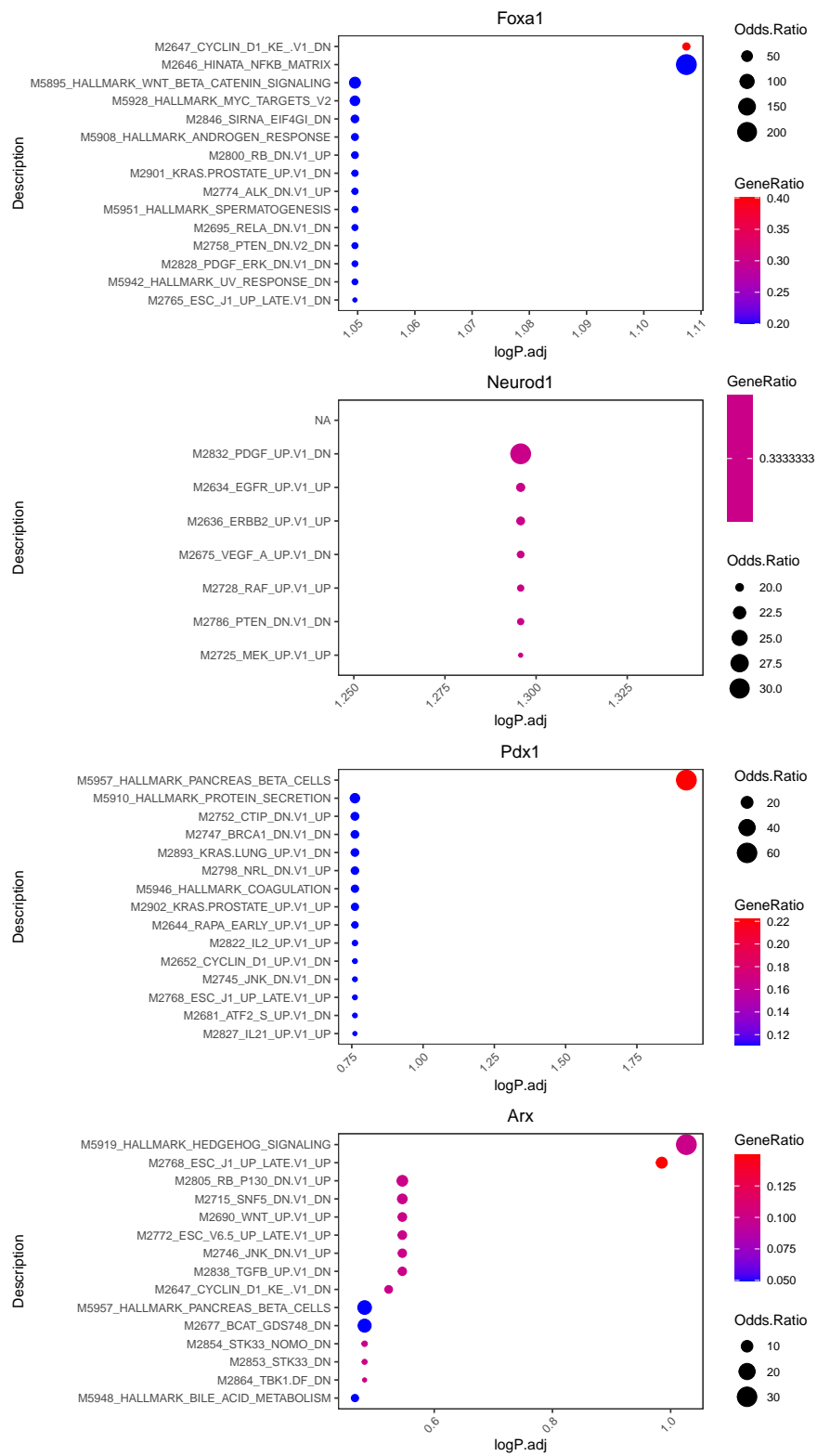
#combine genesets and convert genesets to be compatible with enricher
gs <- c(H,C6)
gs.list <- do.call(rbind,lapply(names(gs), function(x)
  {data.frame(gs = x, genes = gs[[x])}))

enrichresults <- regulonEnrich(genes_to_plot,
                              regulon = regulon.ms,
                              corr = "weight",
                              corr_cutoff = 0.5,
                              genesets = gs.list)

#plot results
enrichPlot(results = enrichresults, ncol = 1)
```



Dorothea regulon



## 6 Session Info

```

sessionInfo()
## R version 4.2.0 (2022-04-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.6 LTS
##
## Matrix products: default
## BLAS: /usr/local/lib/R/lib/libRblas.so
## LAPACK: /usr/local/lib/R/lib/libRlapack.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] stats4      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
##  [1] org.Mm.eg.db_3.16.0      AnnotationDbi_1.59.1
##  [3] scater_1.25.7            ggplot2_3.3.6
##  [5] scuttle_1.7.4            scRNAseq_2.11.0
##  [7] dorothea_1.9.0           msigdb_7.5.1
##  [9] epi regulon_1.0.14       SingleCellExperiment_1.19.1
## [11] SummarizedExperiment_1.27.3 Biobase_2.57.1
## [13] GenomicRanges_1.49.1     GenomeInfoDb_1.33.7
## [15] IRanges_2.31.2           S4Vectors_0.35.4
## [17] BiocGenerics_0.43.4      MatrixGenerics_1.9.1
## [19] matrixStats_0.62.0       BiocStyle_2.25.0
## [21] rmarkdown_2.16
##
## loaded via a namespace (and not attached):
##  [1] rappdirs_0.3.3          rtracklayer_1.57.0
##  [3] tidyr_1.2.1             bit64_4.0.5
##  [5] knitr_1.40              irlba_2.3.5.1
##  [7] DelayedArray_0.23.2     data.table_1.14.2
##  [9] AnnotationFilter_1.21.0 KEGGREST_1.37.3
## [11] RCurl_1.98-1.9          generics_0.1.3
## [13] GenomicFeatures_1.49.7  ScaledMatrix_1.5.1
## [15] cowplot_1.1.1           RSQLite_2.2.18
## [17] shadowtext_0.1.2        artificer.mae_1.3.4
## [19] bit_4.0.4               enrichplot_1.17.2
## [21] base64url_1.4           gp.cache_1.7.1
## [23] xml2_1.3.3              httpuv_1.6.6
## [25] assertthat_0.2.1        genomitory_2.1.5

```

## Dorothea regulon

```
## [27] viridis_0.6.2          xfun_0.31
## [29] hms_1.1.2             jquerylib_0.1.4
## [31] babelgene_22.9        evaluate_0.16
## [33] promises_1.2.0.1      restfulr_0.0.15
## [35] progress_1.2.2        fansi_1.0.3
## [37] dbplyr_2.2.1          Rgraphviz_2.41.1
## [39] igraph_1.3.5          DBI_1.1.3
## [41] purrr_0.3.4           ellipsis_0.3.2
## [43] dplyr_1.0.10          backports_1.4.1
## [45] bookdown_0.29         annotate_1.75.0
## [47] biomaRt_2.53.2        sparseMatrixStats_1.9.0
## [49] artificer.matrix_1.3.7 vctrs_0.4.1
## [51] ensemblDb_2.21.5      dsdb.plus_1.3.2
## [53] cachem_1.0.6          withr_2.5.0
## [55] ggforce_0.4.1         checkmate_2.1.0
## [57] GenomicAlignments_1.33.1 metacommons_1.9.0
## [59] treeio_1.21.2         prettyunits_1.1.1
## [61] MultiAssayExperiment_1.23.9 scran_1.25.1
## [63] cluster_2.1.3         DOSE_3.23.2
## [65] ExperimentHub_2.5.0   BiocBaseUtils_0.99.12
## [67] ape_5.6-2             lazyeval_0.2.2
## [69] crayon_1.5.1          edgeR_3.39.6
## [71] pkgconfig_2.0.3       labeling_0.4.2
## [73] tweenr_2.0.2          ProtGenerics_1.29.0
## [75] nlme_3.1-159          vipor_0.4.5
## [77] rlang_1.0.6           lifecycle_1.0.1
## [79] artificer.schemas_0.99.2 downloader_0.4
## [81] filelock_1.0.2        artificer.base_1.3.18
## [83] BiocFileCache_2.5.0   rsvd_1.0.5
## [85] AnnotationHub_3.5.2   polyclip_1.10-0
## [87] GSVA_1.45.3           graph_1.75.0
## [89] Matrix_1.4-1          aplot_0.1.7
## [91] Rhdf5lib_1.19.2       beeswarm_0.4.0
## [93] rjson_0.2.21          png_0.1-7
## [95] viridisLite_0.4.1     artificer.ranges_1.3.4
## [97] bitops_1.0-7          artificer.se_1.3.4
## [99] getPass_0.2-2         gson_0.0.9
## [101] rhdf5filters_1.9.0    EnrichmentBrowser_2.27.0
## [103] Biostrings_2.65.6     blob_1.2.3
## [105] DelayedMatrixStats_1.19.0 stringr_1.4.0
## [107] qvalue_2.29.0         gridGraphics_0.5-1
## [109] beachmat_2.13.4       scales_1.2.1
## [111] memoise_2.0.1         GSEABase_1.59.0
## [113] magrittr_2.0.3        plyr_1.8.7
## [115] zlibbioc_1.43.0       compiler_4.2.0
## [117] scatterpie_0.1.8      BiocIO_1.7.1
## [119] dqrng_0.3.0           RColorBrewer_1.1-3
## [121] KEGGgraph_1.57.0      Rsamtools_2.13.4
## [123] cli_3.3.0             XVector_0.37.1
## [125] patchwork_1.1.2       ArtifactDB_1.9.5
## [127] MASS_7.3-58.1         tidyselect_1.1.2
```

## Dorothea regulon

```
## [129] stringi_1.7.6          highr_0.9
## [131] yaml_2.3.5             GOSemSim_2.23.0
## [133] BiocSingular_1.13.1    locfit_1.5-9.6
## [135] ggrepel_0.9.1         grid_4.2.0
## [137] sass_0.4.2            bcellViper_1.33.0
## [139] fastmatch_1.1-3       tools_4.2.0
## [141] parallel_4.2.0        bluster_1.7.0
## [143] metapod_1.5.0         gridExtra_2.3
## [145] farver_2.1.1          ggraph_2.0.6
## [147] digest_0.6.29         BiocManager_1.30.18
## [149] FNN_1.1.3.1           shiny_1.7.2
## [151] Rcpp_1.0.9            BiocVersion_3.16.0
## [153] later_1.3.0           gp.auth_1.7.0
## [155] httr_1.4.3            colorspace_2.0-3
## [157] XML_3.99-0.11         splines_4.2.0
## [159] uwot_0.1.14           yulab.utils_0.0.5
## [161] statmod_1.4.37        tidytree_0.4.1
## [163] graphlayouts_0.8.2    ArchR_1.0.2
## [165] ggplotify_0.1.0       xtable_1.8-4
## [167] jsonlite_1.8.2        ggtree_3.5.3
## [169] tidygraph_1.2.2       ggfun_0.0.7
## [171] ShadowArray_1.7.1     R6_2.5.1
## [173] mime_0.12             pillar_1.7.0
## [175] htmltools_0.5.3       glue_1.6.2
## [177] fastmap_1.1.0         clusterProfiler_4.5.2
## [179] BiocParallel_1.31.12  BiocNeighbors_1.15.1
## [181] interactiveDisplayBase_1.35.0 codetools_0.2-18
## [183] fgsea_1.23.2          gp.version_1.5.0
## [185] utf8_1.2.2            lattice_0.20-45
## [187] bslib_0.4.0           tibble_3.1.7
## [189] curl_4.3.2            genomitory.schemas_0.99.0
## [191] ggbeeswarm_0.6.0      artificer.sce_1.3.4
## [193] magick_2.7.3          GO.db_3.16.0
## [195] limma_3.53.10         dsassembly_1.7.6
## [197] dsdb.schemas_0.99.1  munsell_0.5.0
## [199] DO.db_2.9             rhdf5_2.41.1
## [201] GenomeInfoDbData_1.2.9 HDF5Array_1.25.2
## [203] reshape2_1.4.4        gtable_0.3.1
```