Dorothea tutorial

Xiaosai Yao

3 June 2023

Package

epiregulon 1.0.25

Contents

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.

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

#known tfs
genes_to_plot <- c("Foxa1", "Neurod1","Pdx1","Arx")
```
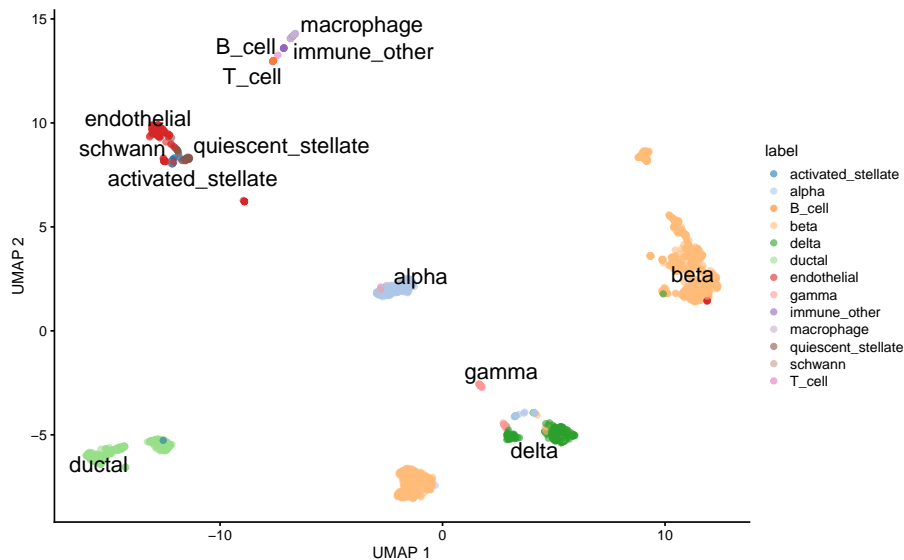
2   Load scRNA-seq data

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")
```

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.

```r
library(epiregulon)


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

#Calculate activity
score.combine <- calculateActivity(sce,
                                   regulon = regulon.ms,
                                   mode = "weight",
                                   method = "weightedMean")
```

4   Perform differential activity

```r
markers <- findDifferentialActivity(activity_matrix = score.combine,
                                    groups = sce$label,
```

```
                                            pval.type = "some",
                                            direction = "up",
                                            test.type = "t")
```

Take the top TFs

```
markers.sig <- getSigGenes(markers, topgenes = 5 )
## Using a logFC cutoff of 0.1 for class activated_stellate
## Using a logFC cutoff of 0.1 for class alpha
## Using a logFC cutoff of 0.1 for class B_cell
## Using a logFC cutoff of 0.1 for class beta
## Using a logFC cutoff of 0.1 for class delta
## Using a logFC cutoff of 0.1 for class ductal
## Using a logFC cutoff of 0.1 for class endothelial
## Using a logFC cutoff of 0.1 for class gamma
## Using a logFC cutoff of 0.1 for class immune_other
## Using a logFC cutoff of 0.1 for class macrophage
## Using a logFC cutoff of 0.1 for class quiescent_stellate
## Using a logFC cutoff of 0.1 for class schwann
## Using a logFC cutoff of 0.1 for class T_cell
```

5   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)
```
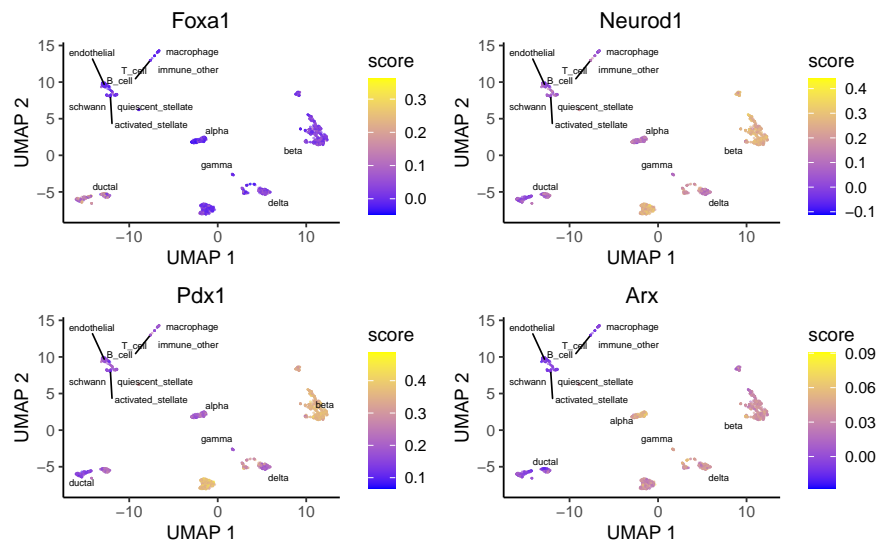
```
# plot violin plot
plotActivityViolin(score.combine,
                   tf = genes_to_plot,
                   clusters = sce$label)
```
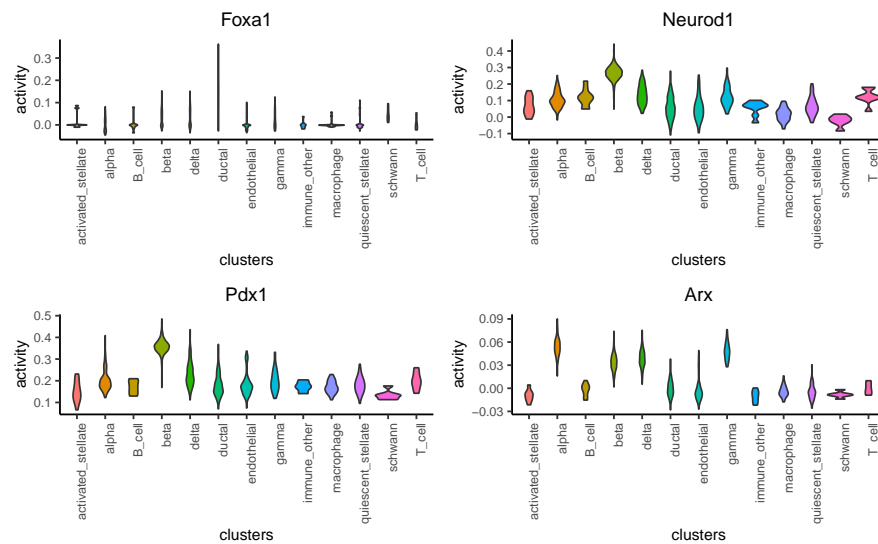


```
# plot bubble plot
plotBubble(score.combine,
           tf = genes_to_plot,
           clusters = sce$label)
```

Dorothea tutorial
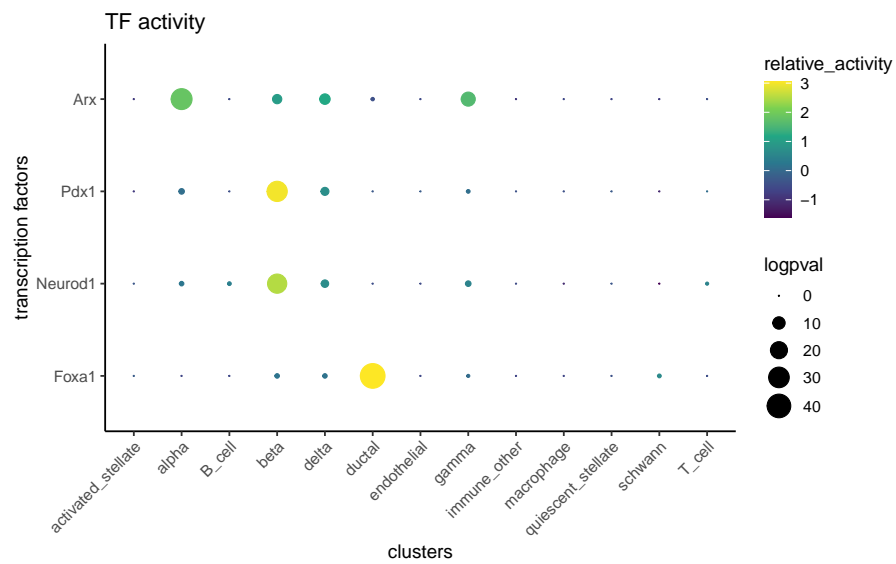
TF activity

Plot bubble plot of differential TFs

```
plotBubble(score.combine,
           tf = markers.sig$tf,
           clusters = sce$label)
```

TF activity

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),
                tf = genes_to_plot,
                legend.label = "score",
                point_size = 0.1,
                dimtype = "UMAP",
                label = "label",
                combine = TRUE,
                text_size = 2,
                colors = c("gray","blue"),
```
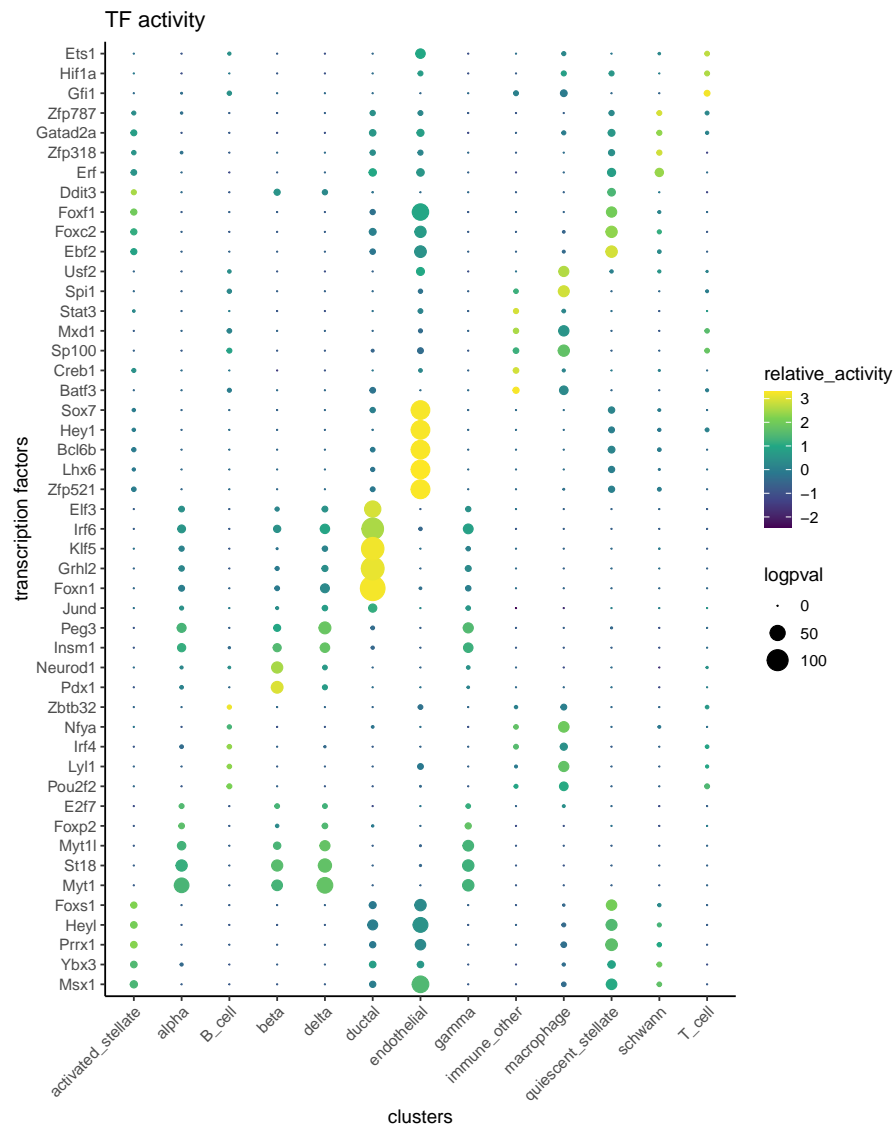
```
                    limit = c(0,2))
```
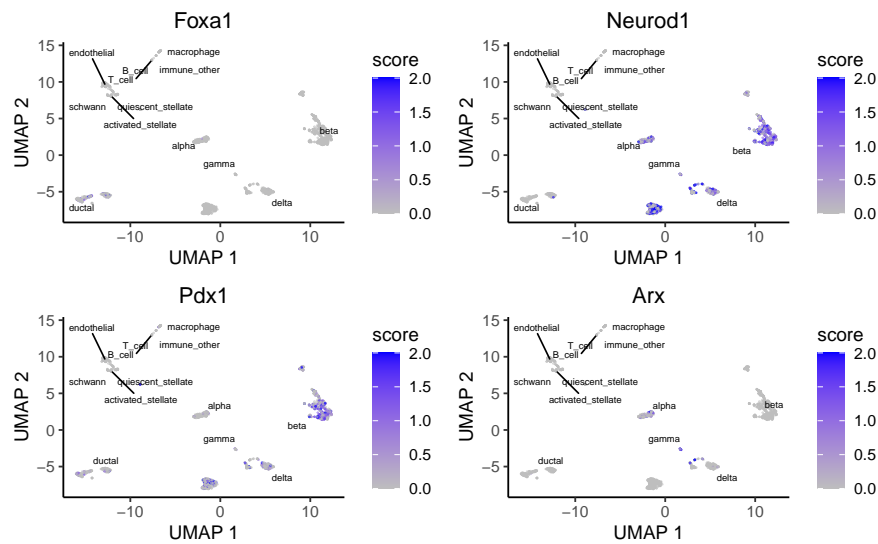


```
# plot violin plot
plotActivityViolin(logcounts(sce),
                   tf = genes_to_plot,
                   clusters = sce$label)
```
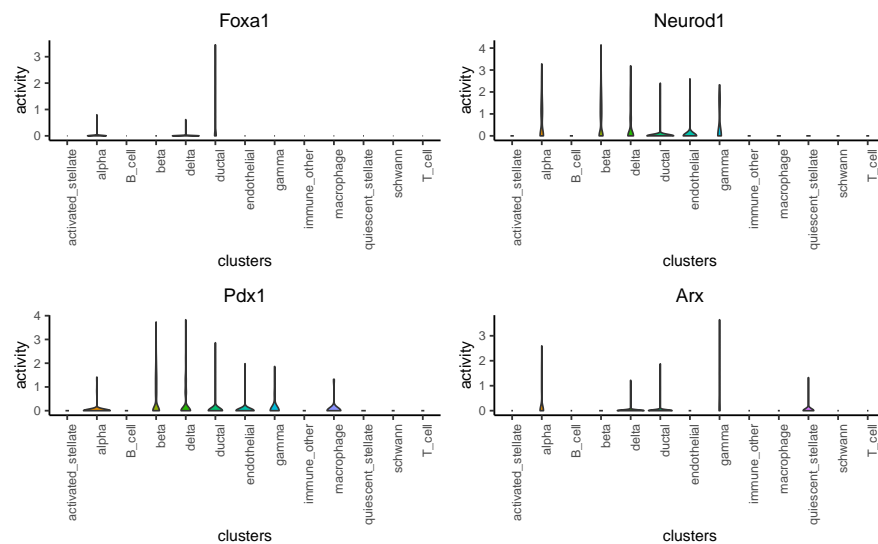


```
# plot Bubble plot
plotBubble(logcounts(sce),
           tf = markers.sig$tf,
           clusters = sce$label)
```

Dorothea tutorial

TF activity
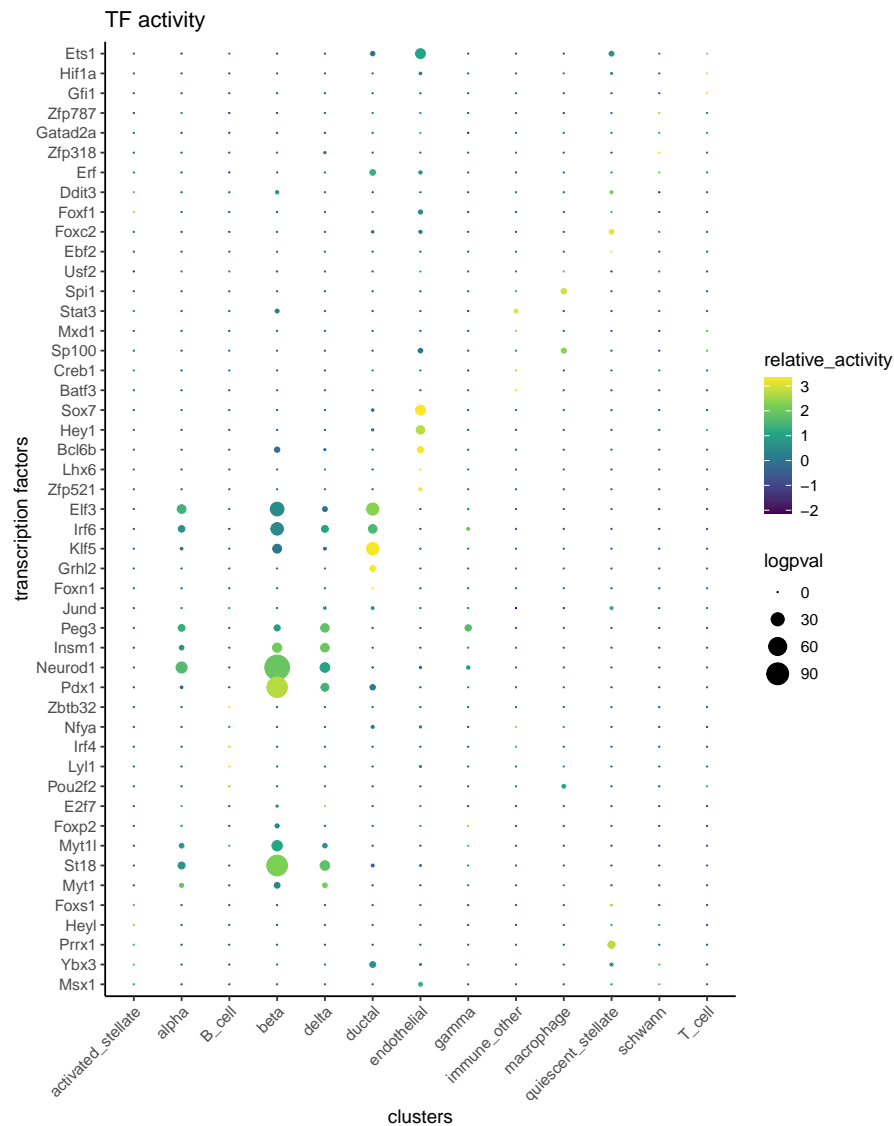
We can visualize the target genes for transcription factors of interest

```
plotHeatmapRegulon(sce=sce,
                   tfs=genes_to_plot,
                   regulon=regulon.ms,
                   regulon_cutoff=0.5,
                   downsample=1000,
                   cell_attributes="label",
                   col_gap="label",
                   exprs_values="logcounts",
                   name="regulon heatmap",
                   column_title_rot = 45)
```
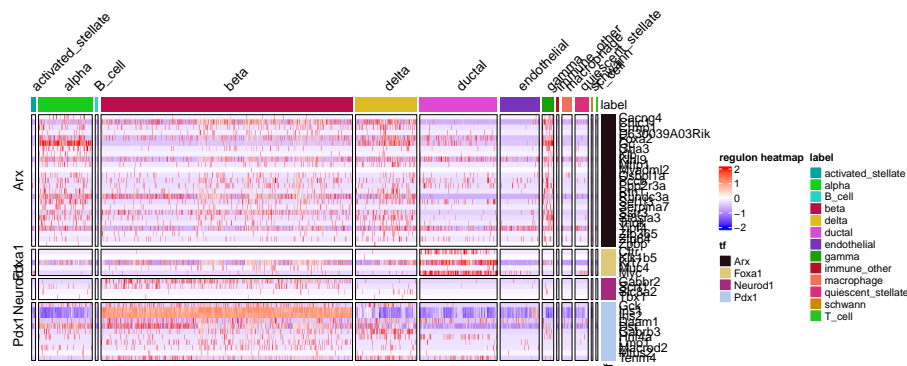
```
plotHeatmapActivity(activity_matrix = score.combine,
                    sce=sce,
                    tfs=genes_to_plot,
                    downsample=1000,
                    cell_attributes="label",
                    col_gap="label",
                    name="regulon heatmap",
                    column_title_rot = 45)
```
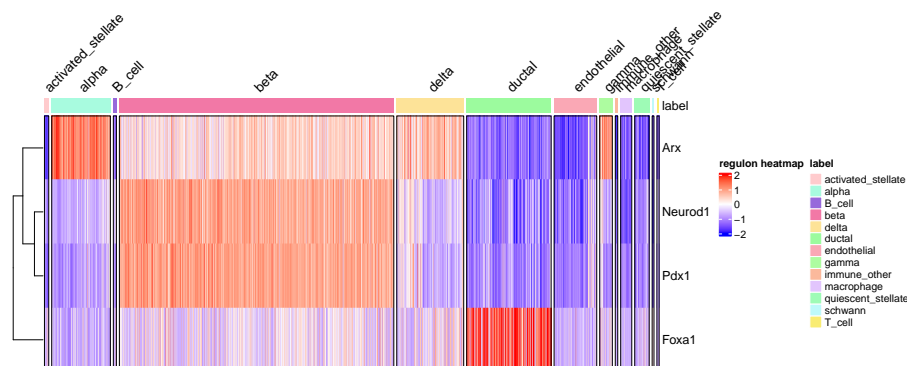

6   Pathway enrichment

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",
                                    cache = FALSE)
C6 <- EnrichmentBrowser::getGenesets(org = "mmu",
                                     db = "msigdb",
                                     cat = "C6",
```

```r
                                 gene.id.type = "SYMBOL",
                                 cache = FALSE)

#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,
                               weight = "weight",
                               weight_cutoff = 0.5,
                               genesets = gs.list)
## Foxa1
##
## Neurod1
## Pdx1
## Arx

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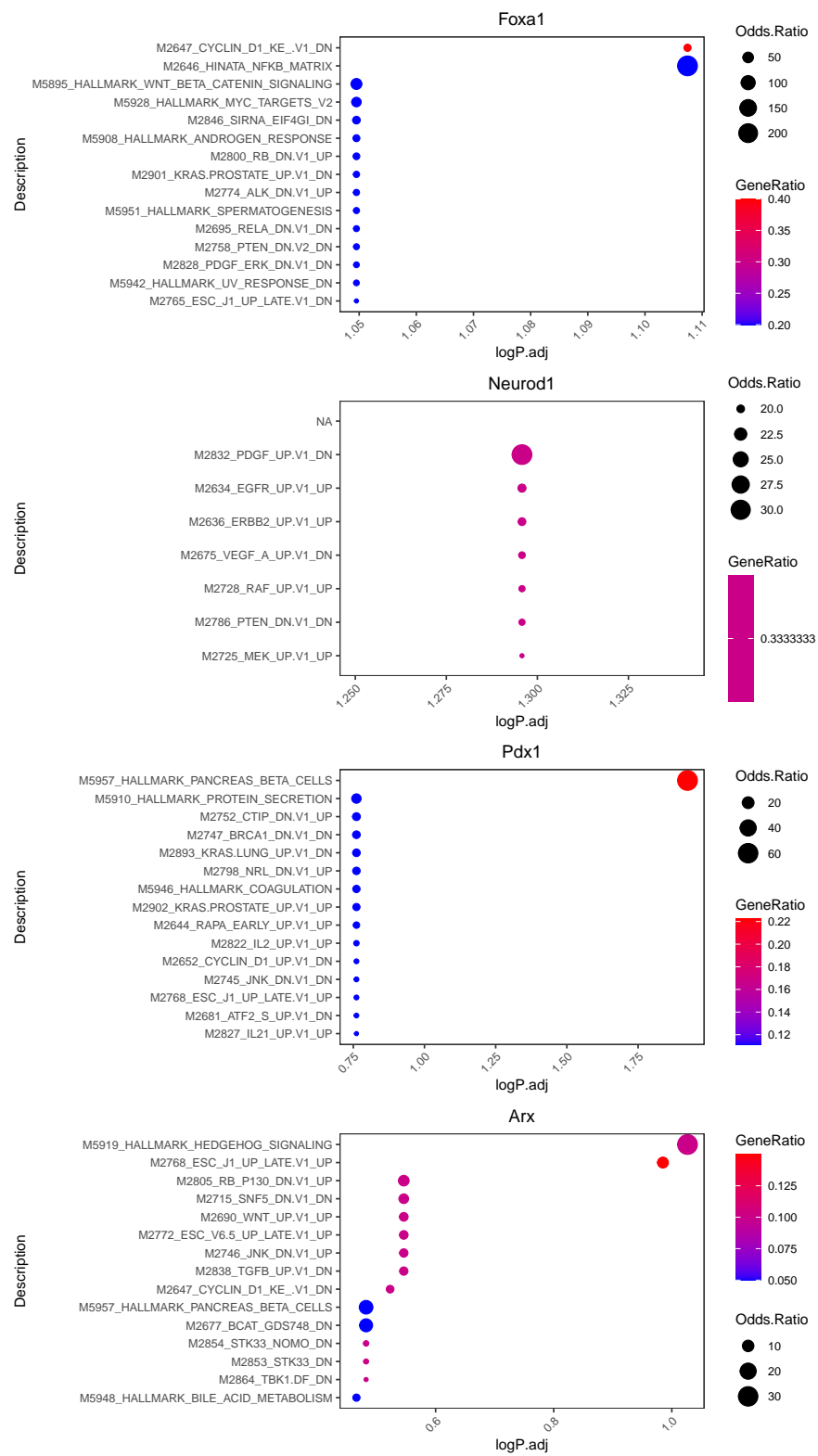
7   Session Info

```
sessionInfo()
## R Under development (unstable) (2022-11-21 r83371)
## 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
##
## time zone: Etc/UTC
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] org.Mm.eg.db_3.17.0         AnnotationDbi_1.61.2
##  [3] msigdbr_7.5.1               epiregulon_1.0.25
##  [5] scater_1.27.9               ggplot2_3.4.2
##  [7] scuttle_1.9.4               scRNAseq_2.13.0
##  [9] SingleCellExperiment_1.21.1 SummarizedExperiment_1.29.1
## [11] Biobase_2.59.0              GenomicRanges_1.51.4
## [13] GenomeInfoDb_1.35.17        IRanges_2.33.1
## [15] S4Vectors_0.37.7            BiocGenerics_0.45.3
## [17] MatrixGenerics_1.11.1       matrixStats_0.63.0
## [19] dorothea_1.11.0             BiocStyle_2.27.2
##
## loaded via a namespace (and not attached):
##   [1] ProtGenerics_1.31.0              GSVA_1.47.3
##   [3] bitops_1.0-7                     enrichplot_1.19.2
##   [5] HDO.db_0.99.1                    httr_1.4.5
##   [7] RColorBrewer_1.1-3               doParallel_1.0.17
##   [9] Rgraphviz_2.43.0                 tools_4.3.0
##  [11] backports_1.4.1                  utf8_1.2.2
##  [13] R6_2.5.1                         HDF5Array_1.27.0
##  [15] lazyeval_0.2.2                   uwot_0.1.14
##  [17] rhdf5filters_1.11.2              GetoptLong_1.0.5
##  [19] withr_2.5.0                      prettyunits_1.1.1
##  [21] gridExtra_2.3                    cli_3.6.1
```

```
##  [23] Cairo_1.6-0                        scatterpie_0.1.9
##  [25] labeling_0.4.2                     KEGGgraph_1.59.3
##  [27] yulab.utils_0.0.6                  Rsamtools_2.15.3
##  [29] gson_0.1.0                         DOSE_3.25.0
##  [31] bcellViper_1.35.0                  BSgenome_1.67.4
##  [33] limma_3.55.10                      RSQLite_2.3.1
##  [35] FNN_1.1.3.2                        gridGraphics_0.5-1
##  [37] generics_0.1.3                     shape_1.4.6
##  [39] BiocIO_1.9.2                       dplyr_1.1.2
##  [41] GO.db_3.17.0                       Matrix_1.5-3
##  [43] ggbeeswarm_0.7.1                   fansi_1.0.3
##  [45] lifecycle_1.0.3                    yaml_2.3.6
##  [47] edgeR_3.41.9                       rhdf5_2.43.5
##  [49] qvalue_2.31.1                      BiocFileCache_2.7.2
##  [51] grid_4.3.0                         blob_1.2.4
##  [53] promises_1.2.0.1                   dqrng_0.3.0
##  [55] ExperimentHub_2.7.1               crayon_1.5.2
##  [57] lattice_0.20-45                    beachmat_2.15.2
##  [59] cowplot_1.1.1                      GenomicFeatures_1.51.4
##  [61] annotate_1.77.0                    KEGGREST_1.39.0
##  [63] magick_2.7.4                       pillar_1.9.0
##  [65] knitr_1.42                         ComplexHeatmap_2.15.3
##  [67] metapod_1.7.0                      fgsea_1.25.2
##  [69] rjson_0.2.21                       codetools_0.2-18
##  [71] fastmatch_1.1-3                    glue_1.6.2
##  [73] ggfun_0.0.9                        downloader_0.4
##  [75] data.table_1.14.8                  treeio_1.23.1
##  [77] vctrs_0.6.2                        png_0.1-8
##  [79] gtable_0.3.3                       cachem_1.0.6
##  [81] xfun_0.39                          mime_0.12
##  [83] tidygraph_1.2.3                    iterators_1.0.14
##  [85] statmod_1.5.0                      bluster_1.9.1
##  [87] interactiveDisplayBase_1.37.0     ellipsis_0.3.2
##  [89] nlme_3.1-162                       ArchR_1.0.3
##  [91] ggtree_3.7.2                       bit64_4.0.5
##  [93] progress_1.2.2                     filelock_1.0.2
##  [95] irlba_2.3.5.1                      vipor_0.4.5
##  [97] colorspace_2.1-0                   DBI_1.1.3
##  [99] tidyselect_1.2.0                   bit_4.0.5
## [101] compiler_4.3.0                     curl_4.3.3
## [103] graph_1.77.3                       BiocNeighbors_1.17.1
## [105] BSgenome.Mmusculus.UCSC.mm10_1.4.3 xml2_1.3.3
## [107] DelayedArray_0.25.0               shadowtext_0.1.2
## [109] bookdown_0.33                      rtracklayer_1.59.1
## [111] checkmate_2.1.0                    scales_1.2.1
## [113] rappdirs_0.3.3                     stringr_1.5.0
## [115] digest_0.6.30                      rmarkdown_2.21
## [117] BSgenome.Hsapiens.UCSC.hg19_1.4.3 XVector_0.39.0
## [119] htmltools_0.5.5                    pkgconfig_2.0.3
## [121] sparseMatrixStats_1.11.1          dbplyr_2.3.2
## [123] fastmap_1.1.0                      ensembldb_2.23.2
```

```
##  [125] rlang_1.1.0                        GlobalOptions_0.1.2
##  [127] shiny_1.7.4                        DelayedMatrixStats_1.21.0
##  [129] farver_2.1.1                       jsonlite_1.8.4
##  [131] BiocParallel_1.33.12               GOSemSim_2.25.0
##  [133] BiocSingular_1.15.0                RCurl_1.98-1.12
##  [135] magrittr_2.0.3                     ggplotify_0.1.0
##  [137] GenomeInfoDbData_1.2.10            patchwork_1.1.2
##  [139] Rhdf5lib_1.21.1                    munsell_0.5.0
##  [141] Rcpp_1.0.10                        ape_5.7-1
##  [143] babelgene_22.9                     viridis_0.6.2
##  [145] EnrichmentBrowser_2.29.2           stringi_1.7.8
##  [147] ggraph_2.1.0                       MASS_7.3-58.1
##  [149] zlibbioc_1.45.0                    AnnotationHub_3.7.4
##  [151] plyr_1.8.8                         parallel_4.3.0
##  [153] ggrepel_0.9.3                      graphlayouts_0.8.4
##  [155] splines_4.3.0                      Biostrings_2.67.2
##  [157] hms_1.1.3                          circlize_0.4.15
##  [159] locfit_1.5-9.7                     BSgenome.Hsapiens.UCSC.hg38_1.4.5
##  [161] igraph_1.4.2                       reshape2_1.4.4
##  [163] biomaRt_2.55.4                     ScaledMatrix_1.7.1
##  [165] BiocVersion_3.17.1                 XML_3.99-0.14
##  [167] evaluate_0.20                      scran_1.27.3
##  [169] BiocManager_1.30.20                tweenr_2.0.2
##  [171] foreach_1.5.2                      httpuv_1.6.9
##  [173] polyclip_1.10-4                    tidyr_1.3.0
##  [175] purrr_1.0.1                        clue_0.3-64
##  [177] ggforce_0.4.1                      rsvd_1.0.5
##  [179] xtable_1.8-4                       restfulr_0.0.15
##  [181] AnnotationFilter_1.23.0            tidytree_0.4.2
##  [183] later_1.3.0                        viridisLite_0.4.1
##  [185] tibble_3.2.1                       aplot_0.1.10
##  [187] clusterProfiler_4.7.1              memoise_2.0.1
##  [189] beeswarm_0.4.0                     GenomicAlignments_1.35.1
##  [191] cluster_2.1.4                      GSEABase_1.61.2
```