Dorothea tutorial

Xiaosai Yao

18 December 2022

Package

epiregulon 1.0.22

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")
```

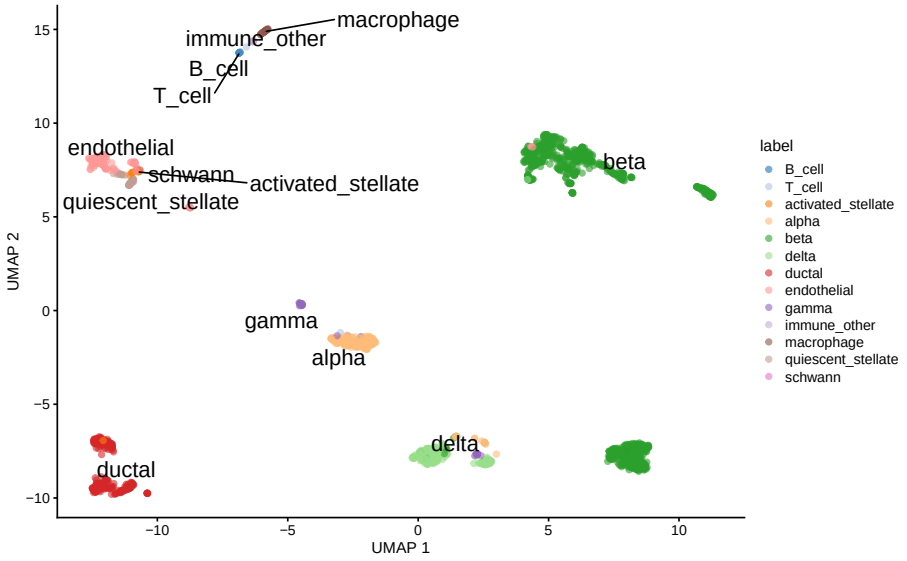
2 Load scRNA-seq data

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.

```
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

```
markers <- findDifferentialActivity(activity_matrix = score.combine,
                                    groups = sce$label,
                                    pval.type = "some",
```

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

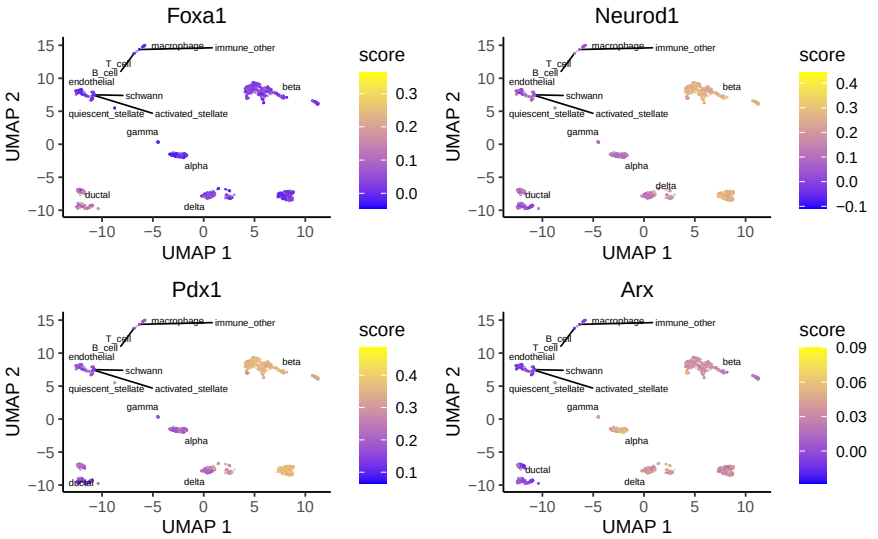
Take the top TFs

```
markers.sig <- getSigGenes(markers, topgenes = 5 )
## Using a logFC cutoff of 0.1 for class B_cell
## Using a logFC cutoff of 0.1 for class T_cell
## 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 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
```

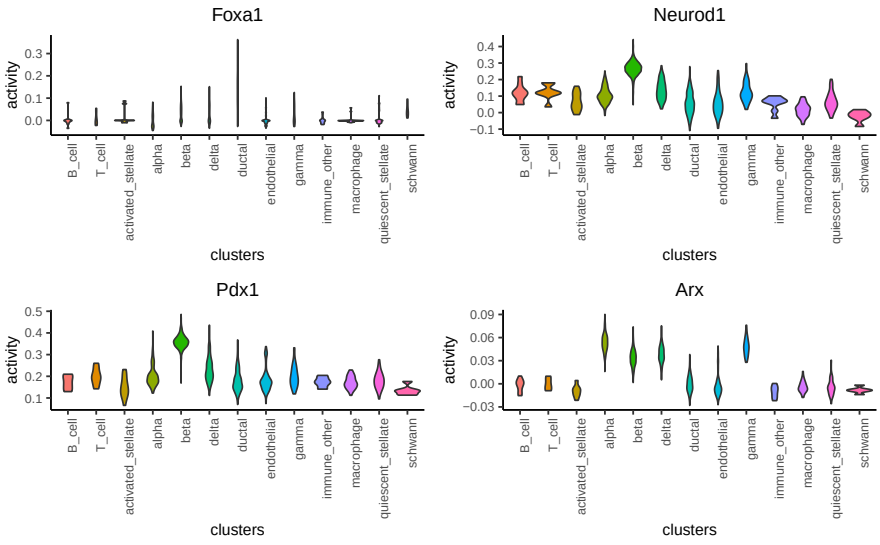
5 Visualize activity

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.

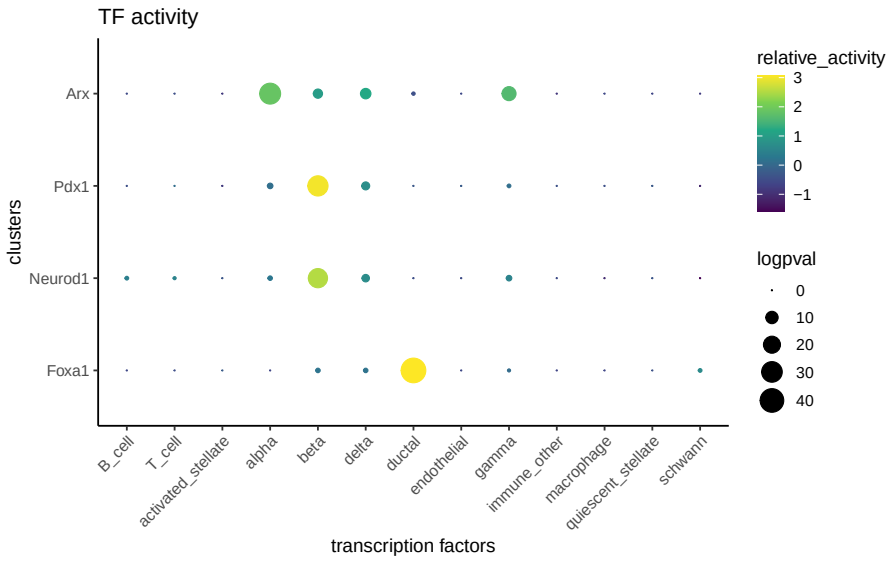
```
# plot umap
plotActivityDim(sce = sce,
                activity_matrix = score.combine,
                tf = genes_to_plot,
                legend.label = "score",
                point_size = 0.1,
                dimtype = "UMAP",
                label = "label",
                combine = TRUE,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




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

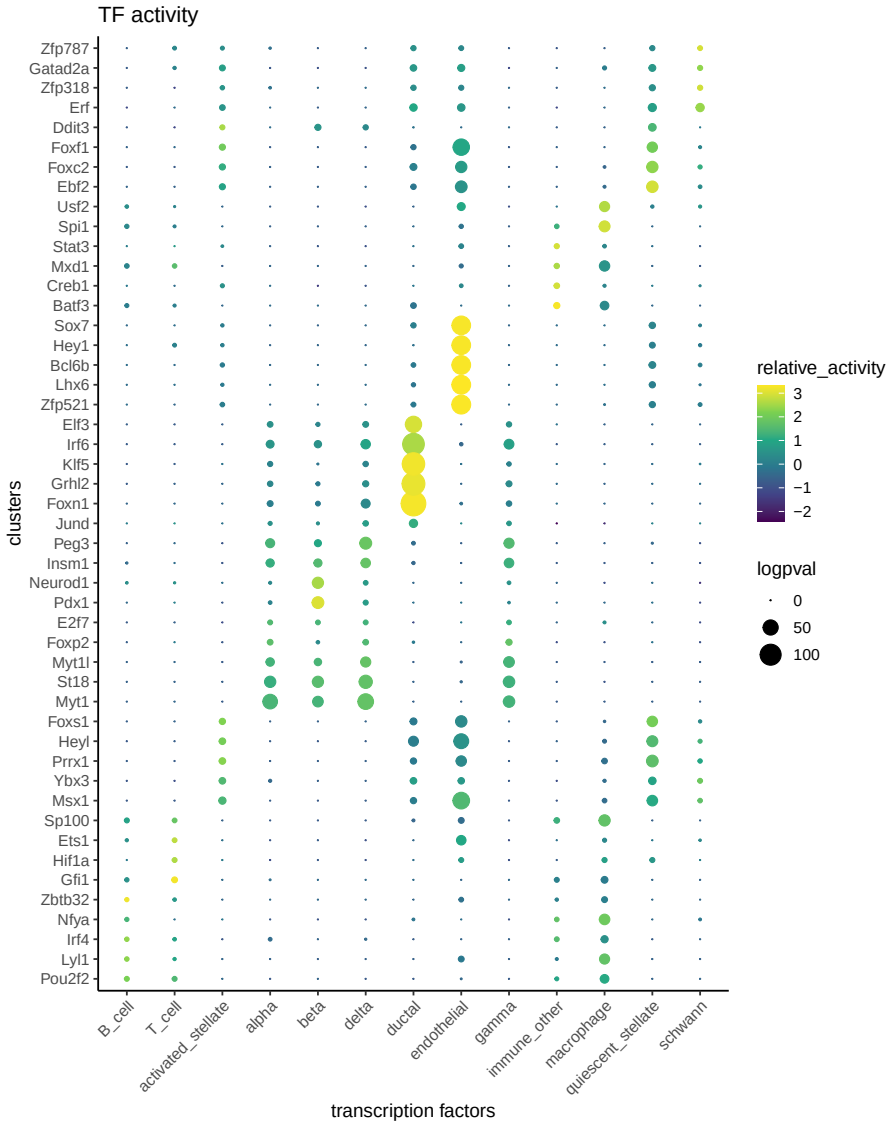



```
# plot bubble plot
plotBubble(score.combine,
           tf = genes_to_plot,
           clusters = sce$label)
```



Plot bubble plot of differential TFs

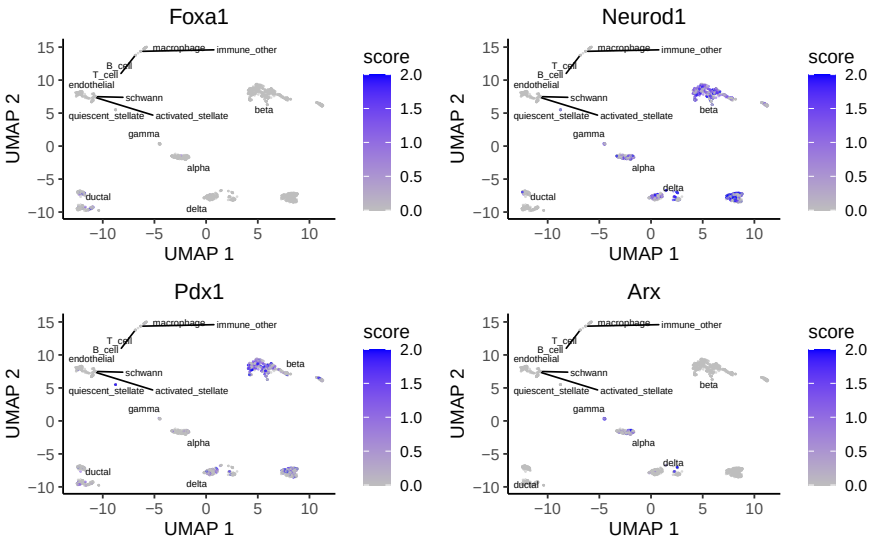
```
plotBubble(score.combine,
           tf = markers.sig$tf,
           clusters = sce$label)
```



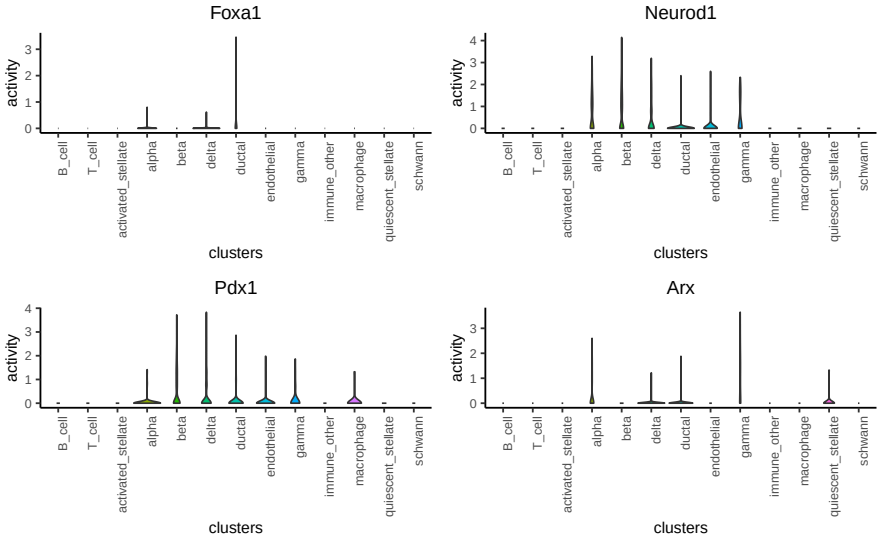
We can adapt the epiregulon package to plot gene expression. When compared against TF activity, gene expression of Foxa1 and Arx has noisy signals and high dropout rates. Epiregulon enhances the signal to noise ratio of TF activity and better resolves lineage differences.

```
# 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"),
```

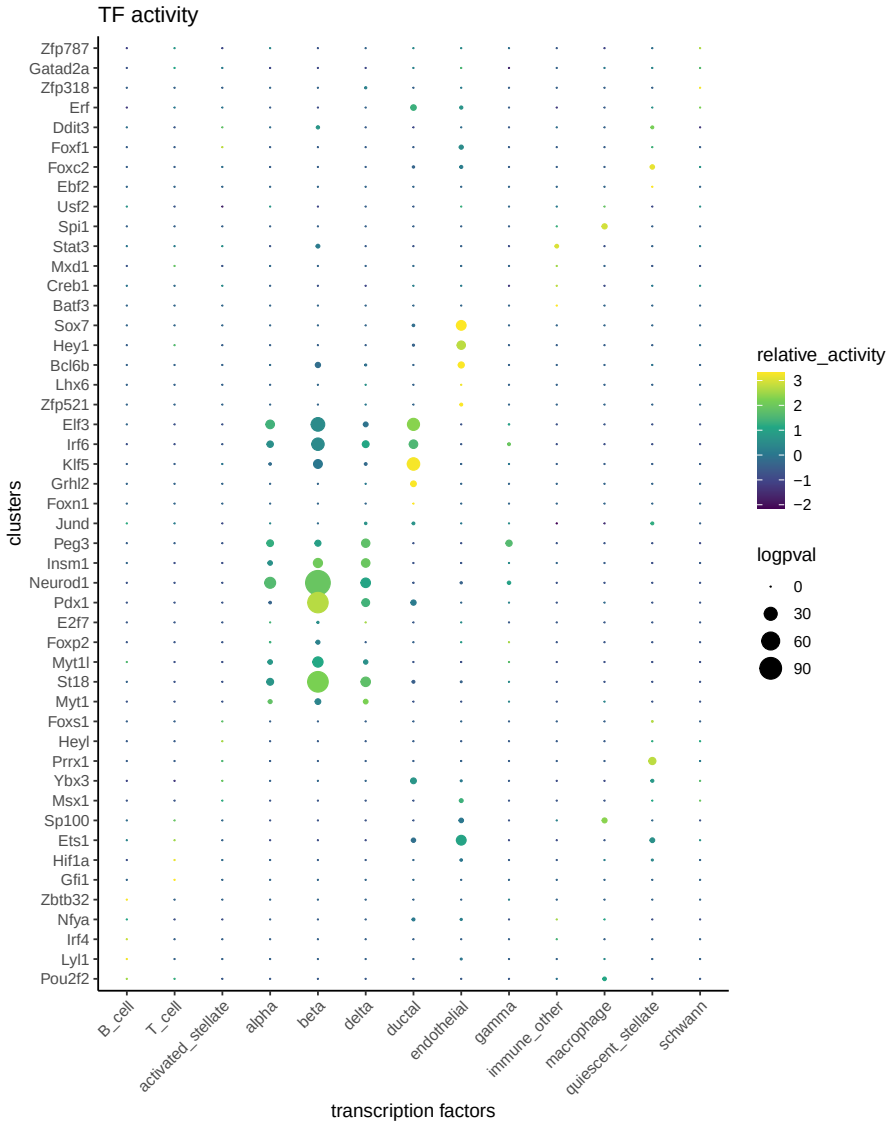
```
                    limit = c(0,2))
```




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

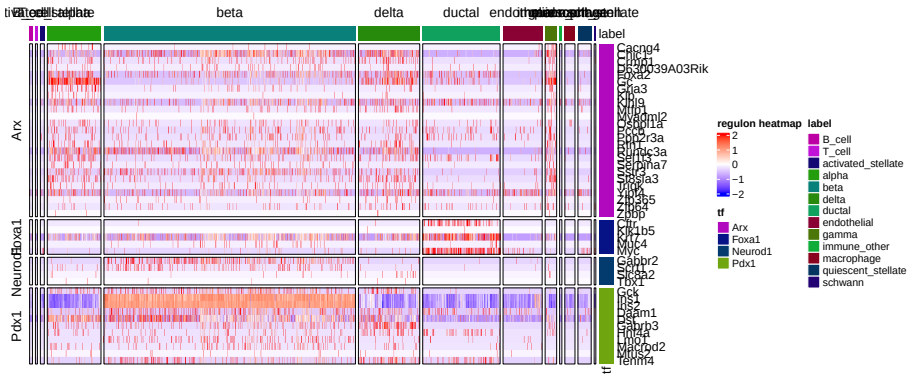



```
# plot Bubble plot
plotBubble(logcounts(sce),
           tf = markers.sig$tf,
           clusters = sce$label)
```

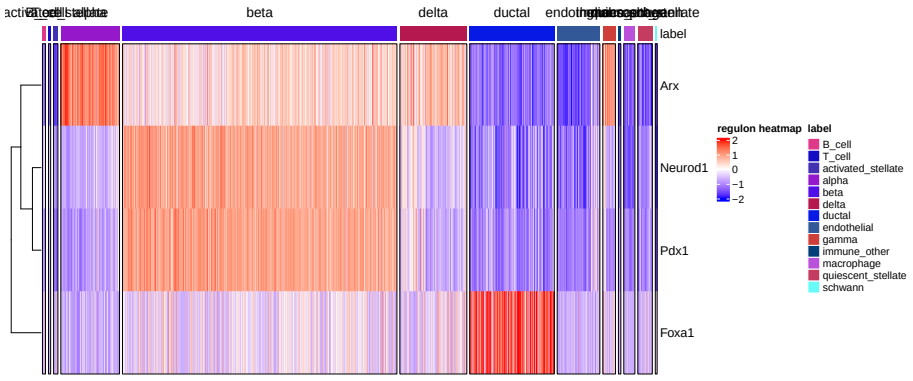


We can visualize the target genes for transcription factors of interest

```
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")
```




```
plotHeatmapActivity(activity_matrix = score.combine,
                    sce=sce,
                    tfs=genes_to_plot,
                    downsample=1000,
                    cell_attributes="label",
                    col_gap="label",
                    name="regulon heatmap")
```



6 Pathway enrichment

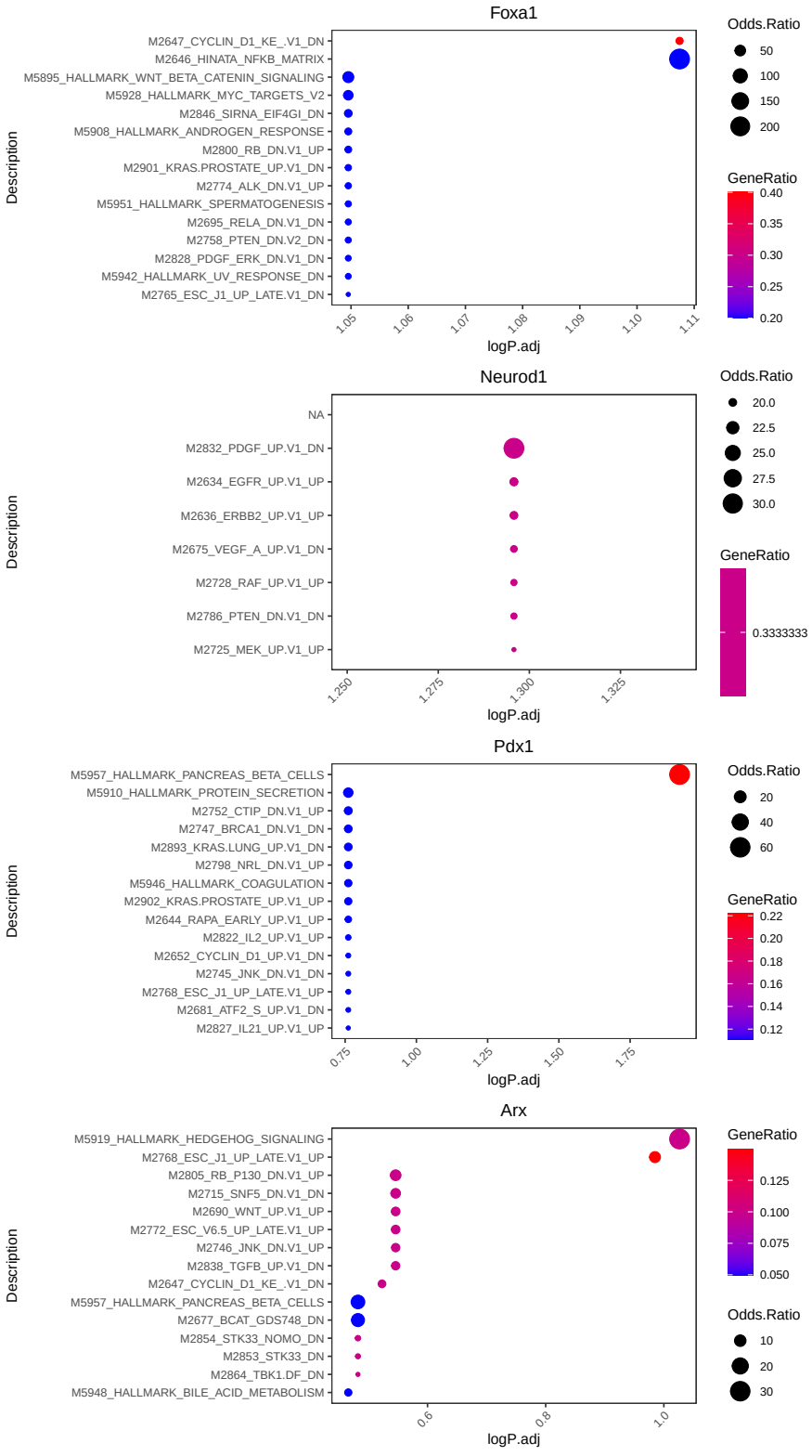
Sometimes it is useful to understand what pathways are enriched in the regulons. We take the highly correlated target genes of a regulon and perform geneset enrichment using the enricher function from clusterProfiler.

```
#retrieve genesets
H <- EnrichmentBrowser::getGenesets(org = "mmu", db = "msigdb", cat = "H", gene.id.type = "SYMBOL" )
## Using cached version from 2022-12-18 22:21:45
C6 <- EnrichmentBrowser::getGenesets(org = "mmu", db = "msigdb", cat = "C6", gene.id.type = "SYMBOL" )
## Using cached version from 2022-12-18 22:21:49

#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)
## Foxa1
## Neurod1
## Pdx1
## Arx

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


Foxa1
M2647_CYCLIN_D1_KE_.V1_DN
M2646_HINATA_NFKB_MATRIX
M5895_HALLMARK_WNT_BETA_CATENIN_SIGNALING
M5928_HALLMARK_MYC_TARGETS_V2
M2846_SIRNA_EIF4GI_DN
M5908_HALLMARK_ANDROGEN_RESPONSE
M2800_RB_DN.V1_UP
M2901_KRAS.PROSTATE_UP.V1_DN
M2774_ALK_DN.V1_UP
M5951_HALLMARK_SPERMATOGENESIS
M2695_RELA_DN.V1_DN
M2758_PTEN_DN.V2_DN
M2828_PDGF_ERK_DN.V1_DN
M5942_HALLMARK_UV_RESPONSE_DN
M2765_ESC_J1_UP_LATE.V1_DN
Description
1.05
1.06
1.07
1.08
1.09
1.10
1.11
logP.adj
Odds.Ratio
50
100
150
200
GeneRatio
0.40
0.35
0.30
0.25
0.20
Neurod1
NA
M2832_PDGF_UP.V1_DN
M2634_EGFR_UP.V1_UP
M2636_ERBB2_UP.V1_UP
M2675_VEGF_A_UP.V1_DN
M2728_RAF_UP.V1_UP
M2786_PTEN_DN.V1_DN
M2725_MEK_UP.V1_UP
Description
1.250
1.275
1.300
1.325
logP.adj
Odds.Ratio
20.0
22.5
25.0
27.5
30.0
GeneRatio
0.3333333
Pdx1
M5957_HALLMARK_PANCREAS_BETA_CELLS
M5910_HALLMARK_PROTEIN_SECRETION
M2752_CTIP_DN.V1_UP
M2747_BRCA1_DN.V1_DN
M2893_KRAS.LUNG_UP.V1_DN
M2798_NRL_DN.V1_UP
M5946_HALLMARK_COAGULATION
M2902_KRAS.PROSTATE_UP.V1_UP
M2644_RAPA_EARLY_UP.V1_UP
M2822_IL2_UP.V1_UP
M2652_CYCLIN_D1_UP.V1_DN
M2745_JNK_DN.V1_DN
M2768_ESC_J1_UP_LATE.V1_UP
M2681_ATF2_S_UP.V1_DN
M2827_IL21_UP.V1_UP
Description
0.75
1.00
1.25
1.50
1.75
logP.adj
Odds.Ratio
20
40
60
GeneRatio
0.22
0.20
0.18
0.16
0.14
0.12
Arx
M5919_HALLMARK_HEDGEHOG_SIGNALING
M2768_ESC_J1_UP_LATE.V1_UP
M2805_RB_P130_DN.V1_UP
M2715_SNF5_DN.V1_DN
M2690_WNT_UP.V1_UP
M2772_ESC_V6.5_UP_LATE.V1_UP
M2746_JNK_DN.V1_UP
M2838_TGFB_UP.V1_DN
M2647_CYCLIN_D1_KE_.V1_DN
M5957_HALLMARK_PANCREAS_BETA_CELLS
M2677_BCAT_GDS748_DN
M2854_STK33_NOMO_DN
M2853_STK33_DN
M2864_TBK1.DF_DN
M5948_HALLMARK_BILE_ACID_METABOLISM
Description
0.6
0.8
1.0
logP.adj
GeneRatio
0.125
0.100
0.075
0.050
Odds.Ratio
10
20
30

7 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         LC_TIME=C            LC_COLLATE=C
##  [5] LC_MONETARY=C        LC_MESSAGES=C        LC_PAPER=C           LC_NAME=C
##  [9] LC_ADDRESS=C         LC_TELEPHONE=C       LC_MEASUREMENT=C     LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] org.Mm.eg.db_3.16.0         AnnotationDbi_1.60.0        msigdbr_7.5.1
##  [4] BiocStyle_2.26.0            epiregulon_1.0.22           scater_1.26.1
##  [7] ggplot2_3.4.0               scuttle_1.8.3               scRNAseq_2.12.0
## [10] SingleCellExperiment_1.20.0 SummarizedExperiment_1.29.1 Biobase_2.58.0
## [13] GenomicRanges_1.50.2        GenomeInfoDb_1.34.4         IRanges_2.32.0
## [16] S4Vectors_0.36.1            BiocGenerics_0.44.0         MatrixGenerics_1.10.0
## [19] matrixStats_0.63.0          dorothea_1.10.0
##
## loaded via a namespace (and not attached):
##   [1] rappdirs_0.3.3               rtracklayer_1.58.0
##   [3] tidyr_1.2.1                  bit64_4.0.5
##   [5] knitr_1.40                   irlba_2.3.5.1
##   [7] DelayedArray_0.24.0          data.table_1.14.6
##   [9] KEGGREST_1.38.0              RCurl_1.98-1.9
##  [11] AnnotationFilter_1.22.0      doParallel_1.0.17
##  [13] generics_0.1.3               GenomicFeatures_1.50.3
##  [15] ScaledMatrix_1.6.0           callr_3.7.0
##  [17] cowplot_1.1.1                usethis_2.1.6
##  [19] RSQLite_2.2.19               shadowtext_0.1.2
##  [21] enrichplot_1.18.3            bit_4.0.5
##  [23] xml2_1.3.3                   httpuv_1.6.7
##  [25] assertthat_0.2.1             viridis_0.6.2
##  [27] xfun_0.31                    hms_1.1.2
##  [29] jquerylib_0.1.4              babelgene_22.9
##  [31] evaluate_0.19                promises_1.2.0.1
##  [33] fansi_1.0.3                  restfulr_0.0.15
##  [35] progress_1.2.2               dbplyr_2.2.1
##  [37] Rgraphviz_2.42.0             igraph_1.3.5
##  [39] DBI_1.1.3                    purrr_0.3.5
##  [41] ellipsis_0.3.2               dplyr_1.0.10
```

```
##  [43] backports_1.4.1              bookdown_0.30
##  [45] annotate_1.76.0              biomaRt_2.54.0
##  [47] sparseMatrixStats_1.10.0     vctrs_0.5.1
##  [49] remotes_2.4.2                ensembldb_2.22.0
##  [51] Cairo_1.6-0                  cachem_1.0.6
##  [53] withr_2.5.0                  ggforce_0.4.1
##  [55] HDO.db_0.99.0                checkmate_2.1.0
##  [57] treeio_1.22.0                GenomicAlignments_1.34.0
##  [59] prettyunits_1.1.1            scran_1.26.1
##  [61] DOSE_3.23.3                  cluster_2.1.3
##  [63] ExperimentHub_2.6.0          ape_5.6-2
##  [65] lazyeval_0.2.2               crayon_1.5.1
##  [67] edgeR_3.40.1                 pkgconfig_2.0.3
##  [69] labeling_0.4.2               tweenr_2.0.2
##  [71] nlme_3.1-161                 vipor_0.4.5
##  [73] pkgload_1.2.4                ProtGenerics_1.30.0
##  [75] devtools_2.4.3               rlang_1.0.6
##  [77] lifecycle_1.0.3              downloader_0.4
##  [79] filelock_1.0.2               BiocFileCache_2.6.0
##  [81] rsvd_1.0.5                   AnnotationHub_3.6.0
##  [83] polyclip_1.10-4              rprojroot_2.0.3
##  [85] GSVA_1.46.0                  graph_1.76.0
##  [87] aplot_0.1.9                  Matrix_1.5-3
##  [89] Rhdf5lib_1.20.0              beeswarm_0.4.0
##  [91] GlobalOptions_0.1.2          processx_3.5.3
##  [93] png_0.1-8                    viridisLite_0.4.1
##  [95] rjson_0.2.21                 bitops_1.0-7
##  [97] gson_0.0.9                   rhdf5filters_1.10.0
##  [99] EnrichmentBrowser_2.28.0     Biostrings_2.66.0
## [101] blob_1.2.3                   DelayedMatrixStats_1.20.0
## [103] shape_1.4.6                  qvalue_2.30.0
## [105] stringr_1.4.0                gridGraphics_0.5-1
## [107] beachmat_2.14.0              scales_1.2.1
## [109] memoise_2.0.1                GSEABase_1.60.0
## [111] magrittr_2.0.3               plyr_1.8.8
## [113] zlibbioc_1.44.0              scatterpie_0.1.8
## [115] compiler_4.2.0               dqrng_0.3.0
## [117] BiocIO_1.8.0                 RColorBrewer_1.1-3
## [119] clue_0.3-63                  KEGGgraph_1.58.2
## [121] Rsamtools_2.14.0             cli_3.4.1
## [123] XVector_0.38.0               patchwork_1.1.2
## [125] ps_1.7.0                     MASS_7.3-58.1
## [127] tidyselect_1.2.0             stringi_1.7.6
## [129] highr_0.9                    GOSemSim_2.24.0
## [131] yaml_2.3.5                   BiocSingular_1.14.0
## [133] locfit_1.5-9.6               ggrepel_0.9.2
## [135] grid_4.2.0                   sass_0.4.4
## [137] bcellViper_1.34.0            fastmatch_1.1-3
## [139] tools_4.2.0                  parallel_4.2.0
## [141] circlize_0.4.15              rstudioapi_0.13
## [143] bluster_1.8.0                foreach_1.5.2
```

```
## [145] metapod_1.6.0                gridExtra_2.3
## [147] farver_2.1.1                 ggraph_2.1.0
## [149] digest_0.6.29                BiocManager_1.30.19
## [151] FNN_1.1.3.1                  shiny_1.7.4
## [153] Rcpp_1.0.9                   BiocVersion_3.16.0
## [155] later_1.3.0                  httr_1.4.3
## [157] ComplexHeatmap_2.14.0        colorspace_2.0-3
## [159] brio_1.1.3                   XML_3.99-0.13
## [161] fs_1.5.2                     splines_4.2.0
## [163] yulab.utils_0.0.5            uwot_0.1.14
## [165] statmod_1.4.37               tidytree_0.4.1
## [167] graphlayouts_0.8.4           ArchR_1.0.2
## [169] ggplotify_0.1.0              sessioninfo_1.2.2
## [171] xtable_1.8-4                 ggtree_3.6.2
## [173] jsonlite_1.8.4               tidygraph_1.2.2
## [175] ggfun_0.0.9                  testthat_3.1.4
## [177] R6_2.5.1                     pillar_1.8.1
## [179] htmltools_0.5.4              mime_0.12
## [181] glue_1.6.2                   fastmap_1.1.0
## [183] clusterProfiler_4.6.0        BiocParallel_1.32.4
## [185] BiocNeighbors_1.16.0         interactiveDisplayBase_1.36.0
## [187] codetools_0.2-18             fgsea_1.24.0
## [189] pkgbuild_1.3.1               utf8_1.2.2
## [191] lattice_0.20-45              bslib_0.4.2
## [193] tibble_3.1.8                 curl_4.3.2
## [195] ggbeeswarm_0.7.1             magick_2.7.3
## [197] GO.db_3.16.0                 limma_3.54.0
## [199] rmarkdown_2.18               desc_1.4.1
## [201] munsell_0.5.0                GetoptLong_1.0.5
## [203] rhdf5_2.42.0                 GenomeInfoDbData_1.2.9
## [205] iterators_1.0.14             HDF5Array_1.26.0
## [207] reshape2_1.4.4               gtable_0.3.1
```