

Dorothea regulon

Xiaosai Yao

April 29, 2022

Epiregulon also supports transcription factor activity inference when users only have scRNA-seq. After all, multiome or scATAC-seq data is still considered a rarity. In this case, 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.

Load regulon

Dorothea assigns confidence level to its regulon 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", "Nkx3-1", "Trp63", "Sox2")
regulon <- regulon[regulon$tf %in% genes_to_plot, ]
```

Load scRNA-seq data

We load a previously analyzed scRNA-seq dataset from resultsDB. Dimension reduction and clustering were already performed. We also need to rename the rownames of the scRNA-seq as gene symbols to be compatible with the regulons.

```
library(ResultsDB)
library(epiregulon)
file.id <- packID("GPA505",
                   "scbasic-results/sce",
                   "PUBLISHED-1")

sce <- getPublishedResult(file.id)

## Warning in .generalCacher(FUN, SAVEFUN = SAVEFUN, LOADFUN = LOADFUN, id = id, : reacquiring resource
## 'https://cerberus.genomics.roche.com/v1/files/DS000007200%3ARES000062229_1_MAE.hdf5%401/metadata' wi
## missing file in cache

## Warning in .generalCacher(FUN, SAVEFUN = SAVEFUN, LOADFUN = LOADFUN, id = id, : reacquiring resource
## 'https://cerberus.genomics.roche.com/v1/files/DS000007200%3ARES000062229_1_MAE.hdf5%401/metadata' wi
## missing file in cache

#rename rownames as gene symbol
rownames(sce) <- rowData(sce)$symbol
```

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="cluster",
                           BPPARAM=BiocParallel::MulticoreParam())

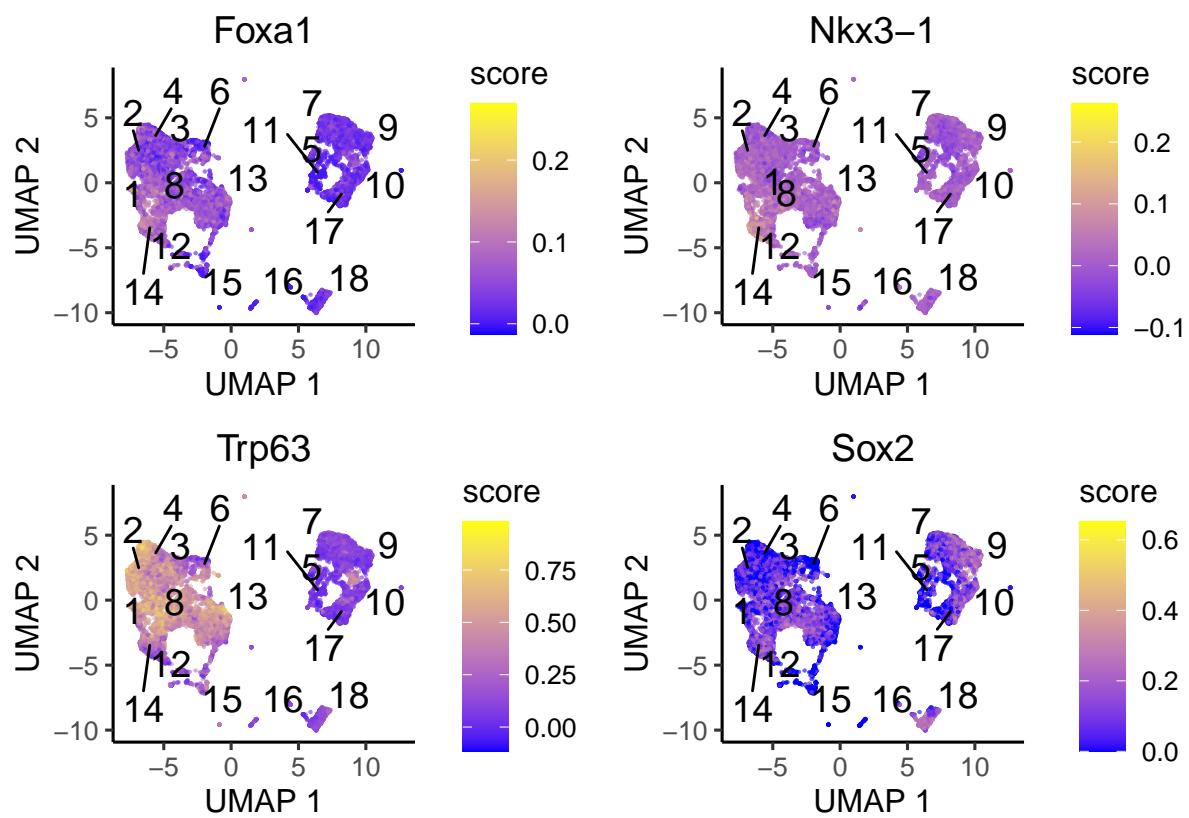
## calculating average expression across clusters...
## computing correlation of the regulon...
##   |
#Calculate activity
score.combine <- calculateActivity(sce, regulon.ms, "weight", method="weightedMean")

## converting DelayedMatrix to dgCMatrix
##   |
```

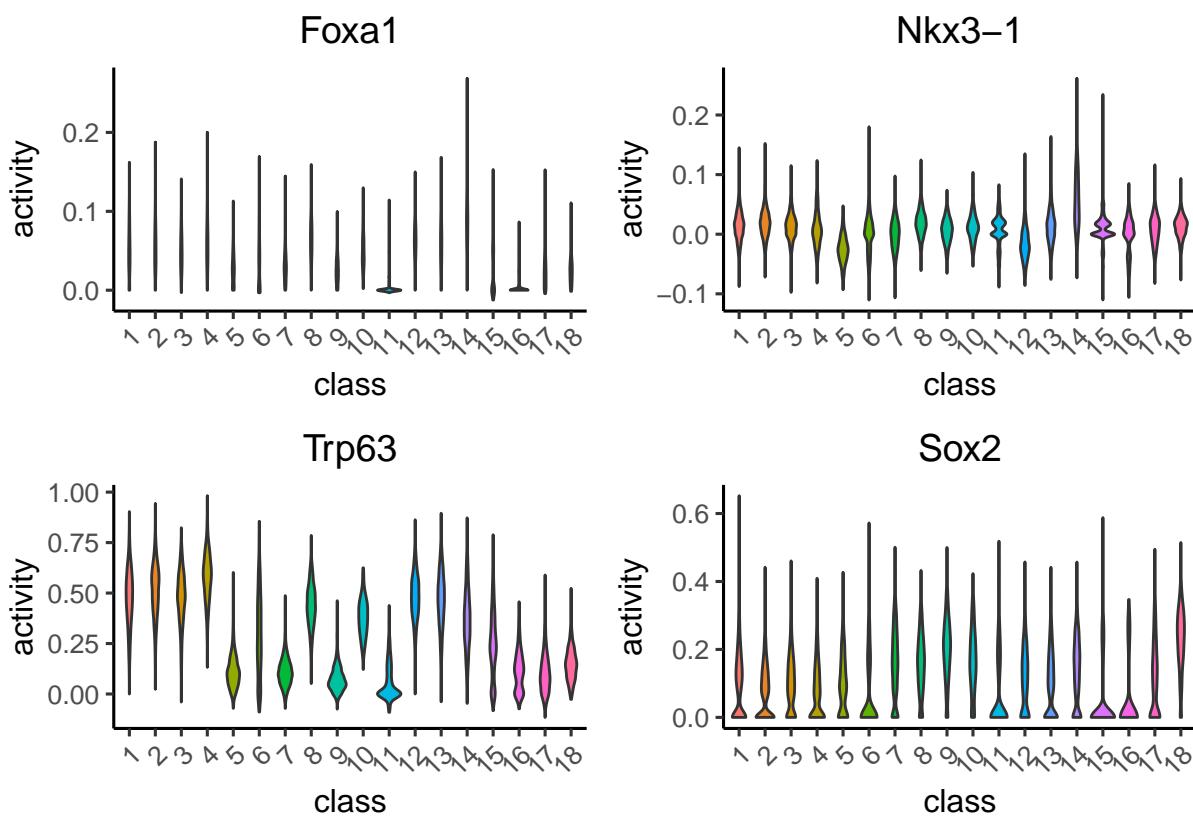
Visualize activity

Finally we finalize the TF activity by either UMAP, violin plots or bubble plots

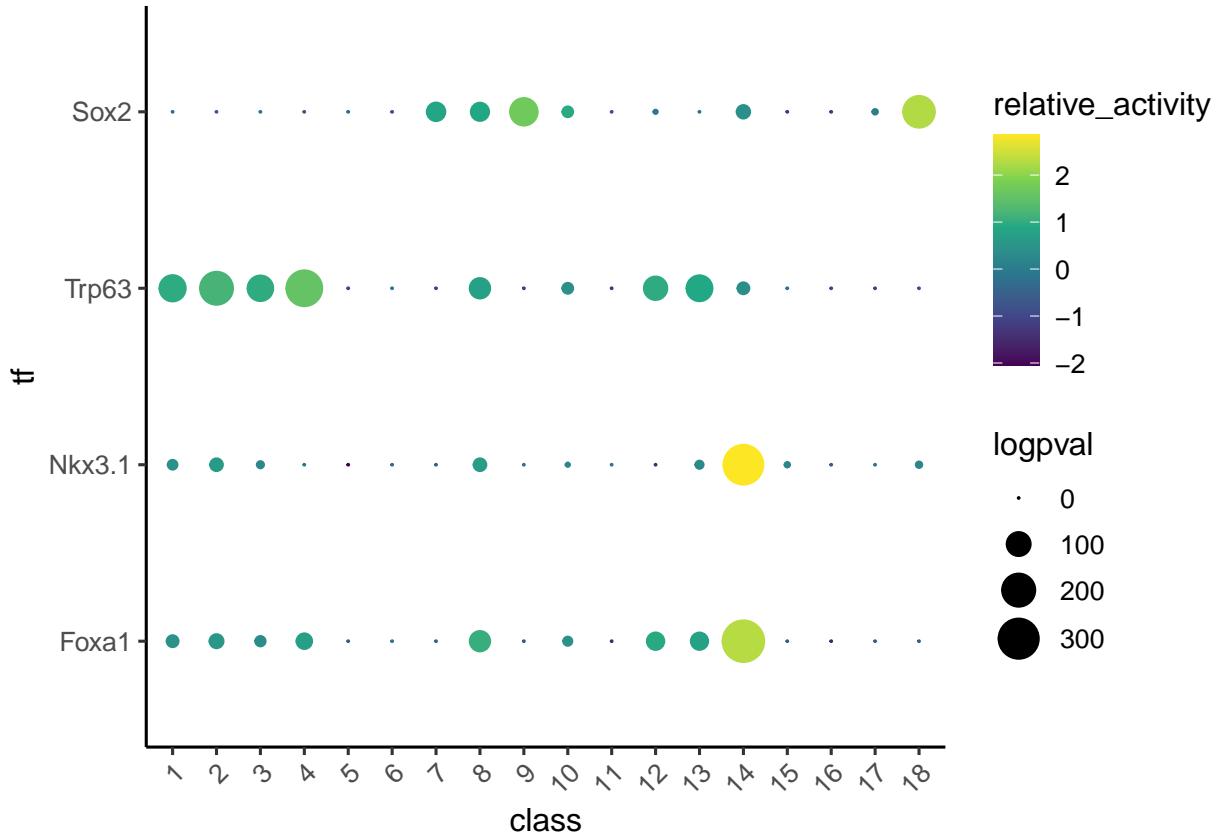
```
# plot umap
plotActivityDim(sce = sce,
                 activity_matrix = score.combine,
                 tf = genes_to_plot,
                 legend.label = "score",
                 point_size = 0.1,
                 dimtype = "UMAP",
                 label = "cluster",
                 combine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



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



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



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 2022-09-03 17:46:34
C6 <- EnrichmentBrowser::getGenesets(org = "mmu", db = "msigdb", cat = "C6", gene.id.type = "SYMBOL" )

## Using cached version from 2022-09-03 17:46:36
#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)
```

