RSCORE_Tutorial_for_HECA

Code ▾

Please notice that, we use an updated BioGRID PPI (Version: 3.5.174, provided, previous version: 3.5.173), thus the result maybe a little different, but the conclusions are consistent with our study.

load the HECA data

Hide

```
# change the directory to yours
setwd("F:/Github/HECA/RSCORE_Tutorial_for_HECA/")
Cortex <- readRDS("../Cortex_UMI_counts_filtered.rds")
FGC <- readRDS("../FGC_UMI_counts_filtered.rds")
Heart <- readRDS("../Heart_UMI_counts_filtered.rds")
Kidney <- readRDS("../Kidney_UMI_counts_filtered.rds")
PFC <- readRDS("../PFC_UMI_counts_filtered.rds")
#combine all the five datasets
HECA <- cbind(Cortex,FGC,Heart,Kidney,PFC)
#check the data size
dim(HECA)
```

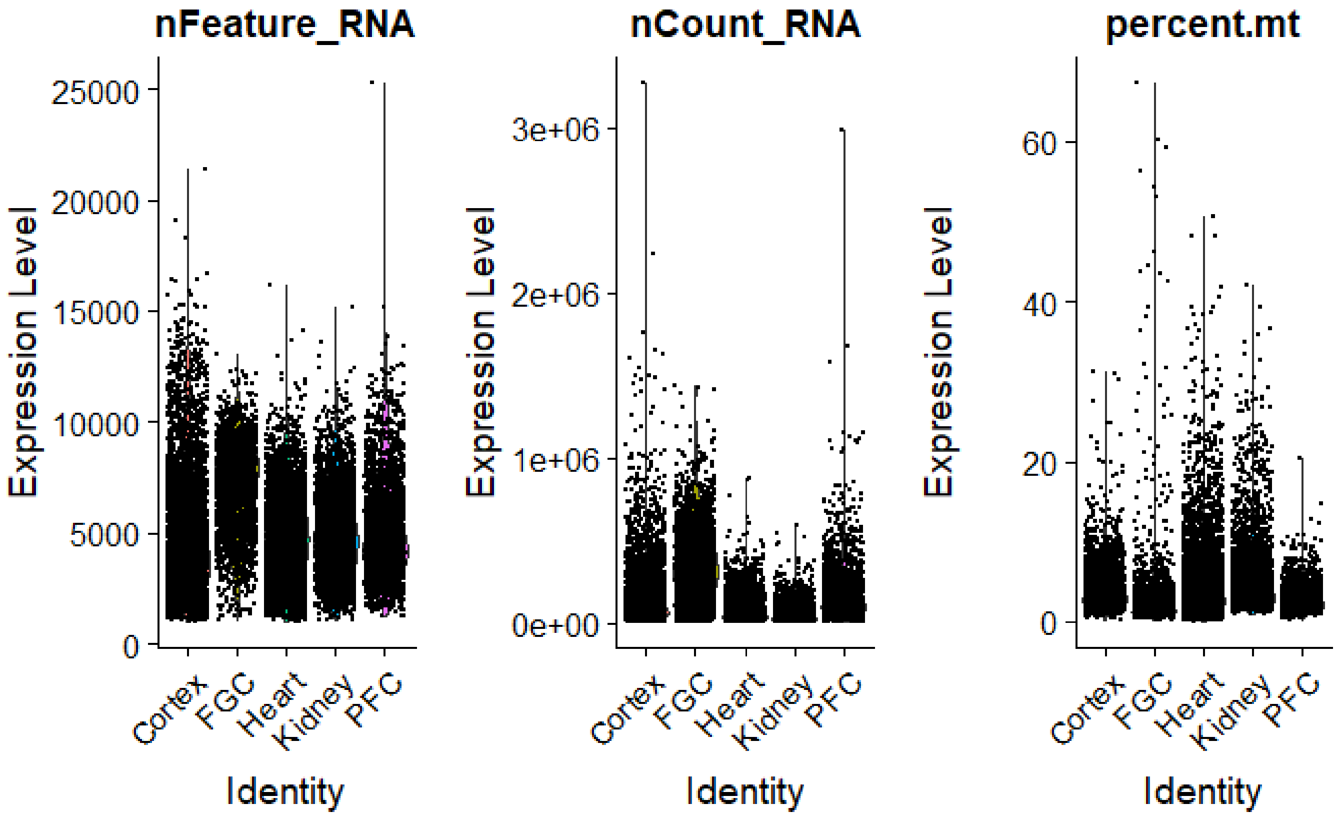
```
[1] 33694 17010
```

Construct a Seurat V3 object

Hide

```
HECA_seurat <- CreateSeuratObject(counts = as.matrix(HECA), min.cells = 3, min.features = 1000,
                                  names.field = 2, names.delim = '_',
                                  assay = 'RNA', project = 'HECA')
```

Check the data quality



Quality control

Hide

```
HECA_seurat <- subset(HECA_seurat, subset = nFeature_RNA > 1000 & nCount_RNA > 10000 & nCount_RNA < 15
00000 & percent.mt < 20)
```

Since we use the UMI count, we need to nomalize the data. We use 100,000 because our data is high-quality rather than 10X data (where you should use 10,000). If you load an already normalized dataset, please skip this step

Hide

```
HECA_seurat <- NormalizeData(HECA_seurat, normalization.method = "LogNormalize", scale.factor = 100000
)
```

```
Performing log-normalization
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

You can use either method to remove low-quality genes,mean.var.plot or vst. Please keep enough genes for the downstream network inference For SMART-seq2 based scRNA-seq, we recommends ~8000 genes, while 10X based data, ~6000 genes. You can also try different gene number to obtain optimal results.

Hide

```
HECA_seurat <- FindVariableFeatures(HECA_seurat, selection.method = "mean.var.plot",
                                    mean.cutoff = c(0.1,Inf), dispersion.cutoff = c(0.1,Inf))
```

```
Calculating gene means
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating gene variance to mean ratios
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

Hide

```
#HECA_seurat <- FindVariableFeatures(object = HECA_seurat, selection.method = 'vst', nfeatures = 1000
0)
#check the number of selected genes
length(VariableFeatures(HECA_seurat))
```

```
[1] 8844
```

Scale the data

Hide

```
HECA_seurat <- ScaleData(object = HECA_seurat)
```

```
Centering and scaling data matrix

  |
  |
|   0%
  |
  |===============
|  11%
  |
  |==============================
|  22%
  |
  |=============================================
|  33%
  |
  |============================================================
|  44%
  |
  |===========================================================================
|  56%
  |
  |==========================================================================================
|  67%
  |
  |===================================================================================================
======                              |  78%
  |
  |===================================================================================================
=====================               |  89%
  |
  |===================================================================================================
====================================| 100%
```

PPI data is necessary. You can provide the adjacent matrix of PPI network by yourself or use our function to build a PPI network easily.

Hide

```
# change the directory to yours
hs_network <- as.matrix(readRDS("hs_ppi_matrix_BioGRID-3.5.174.Rda"))
#build a BioGRID homo sapien PPI, the PPI matrix is saved in the working directory
#hs_network <- as.matrix(getPPI_Biogrid())
```

Now, we can use SCORE to analyze our data

Hide

```
DefaultAssay(HECA_seurat) <- "RNA"
HECA_seurat <- R.SCORE(Data = HECA_seurat, PPI = hs_network, max_step = 10, nCores = 4)
```

The result is saved in 'Net' assay of RCA_seurat (it has been set as default assay). You can plot the UMAP

Hide

```
HECA_seurat <- RunPCA(HECA_seurat, features = rownames(HECA_seurat), assay = "Net", npcs = 50,
                      reduction.name = "NetPCA", reduction.key = "NetPCA_", verbose = F)
HECA_seurat <- RunUMAP(HECA_seurat, reduction = "NetPCA", dims = 1:50,
                       reduction.name = "NetUMAP",  reduction.key = "NetUMAP_")
```

Now we can do clustering, you can change the parameters to obtan optimal results

Hide

```
HECA_seurat <- FindNeighbors(HECA_seurat, reduction = "NetPCA",dims = 1:50,k.param = 10)
HECA_seurat <- FindClusters(HECA_seurat, resolution = 0.8)
```

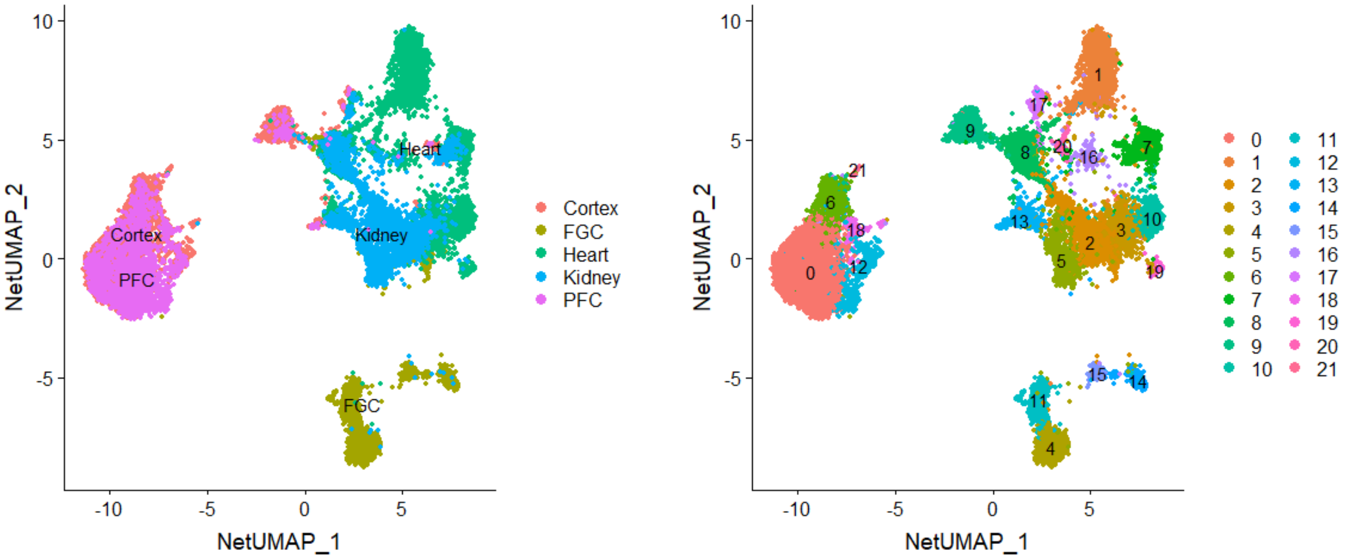
```
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 16773
Number of edges: 339109

Running Louvain algorithm...
```

```
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

```
Maximum modularity in 10 random starts: 0.8916
Number of communities: 22
Elapsed time: 1 seconds
```

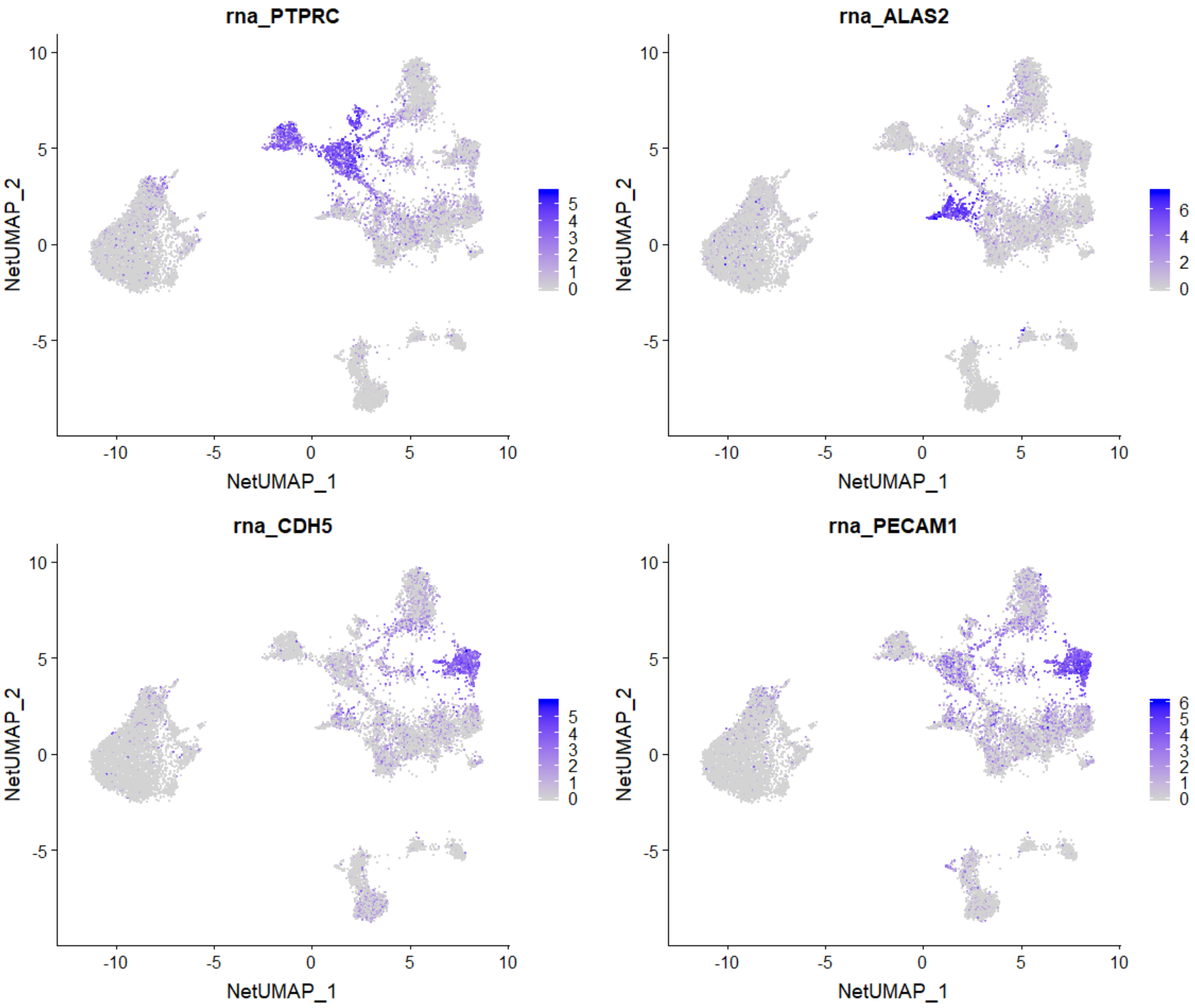
Hide

```
Net_umap_cluster <- DimPlot(HECA_seurat, reduction = "NetUMAP", pt.size = 1,label = T)
Net_umap_origin <- DimPlot(HECA_seurat, reduction = "NetUMAP", pt.size = 1, group.by = 'orig.ident',la
bel = T)
CombinePlots(plots = list(Net_umap_origin, Net_umap_cluster))
```



Hide

```
FeaturePlot(HECA_seurat, reduction = "NetUMAP", c('PTPRC','ALAS2','CDH5','PECAM1'))
```



Hide

```
#Identify the DEGs using genesorteR
SCORE_DEGs_list <- Find_Markers(object = HECA_seurat, assay = 'RNA', FoldChange = 1.5)
table(SCORE_DEGs_list$Markers$Cluster)
```

```

   0    1   10   11   12   13   14   15   17   18   19    2   20   21    3    4    5    6    7    8    
9 
 505  334  205  765   28   58   41   53   55   11   11  201   30   32  222 1400  165 1223  215  326   2
98 
```

Hide

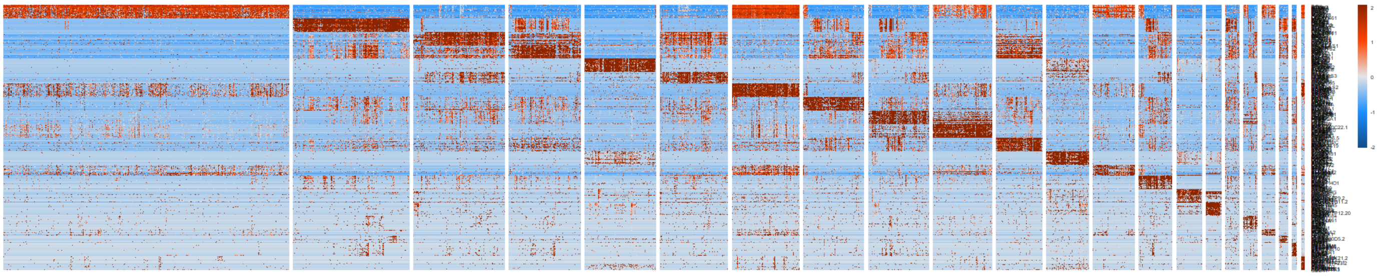
```
SCORE_DAMs_list <- Find_Markers(object = HECA_seurat, assay = 'Net', FoldChange = 1.5)
table(SCORE_DAMs_list$Markers$Cluster)
```

```

  0   1  10  11  12  13  14  15  17   2   3   4   5   6   7   8   9 
614 205 129  48  34  59  36  38  48 155 135 104  66 252 109 171 148 
```

Select the top n marker genes of each cluster

Hide

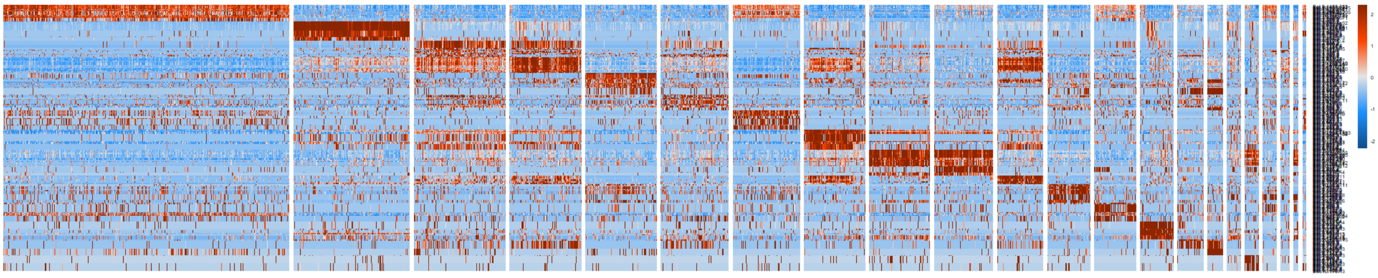
```
top10_DEGs <- SCORE_DEGs_list$Markers %>% group_by(Cluster) %>% top_n(n = 20, wt = Gene.Score)
#genesorteR plotMarkerHeat function
plotMarkerHeat(exp = SCORE_DEGs_list$GeneSort$inputMat,
               classes = SCORE_DEGs_list$GeneSort$inputClass,
               markers = top10_DEGs$Marker,
               clusterGenes = FALSE,
               averageCells = 1)
```



Select the top n marker modules of each cluster

Hide

```
top10_DAMs <- SCORE_DAMs_list$Markers %>% group_by(Cluster) %>% top_n(n = 20, wt = Gene.Score)
plotMarkerHeat(exp = SCORE_DAMs_list$GeneSort$inputMat,
               classes = SCORE_DAMs_list$GeneSort$inputClass,
               markers = top10_DAMs$Marker,
               clusterGenes = FALSE,
               averageCells = 1)
```

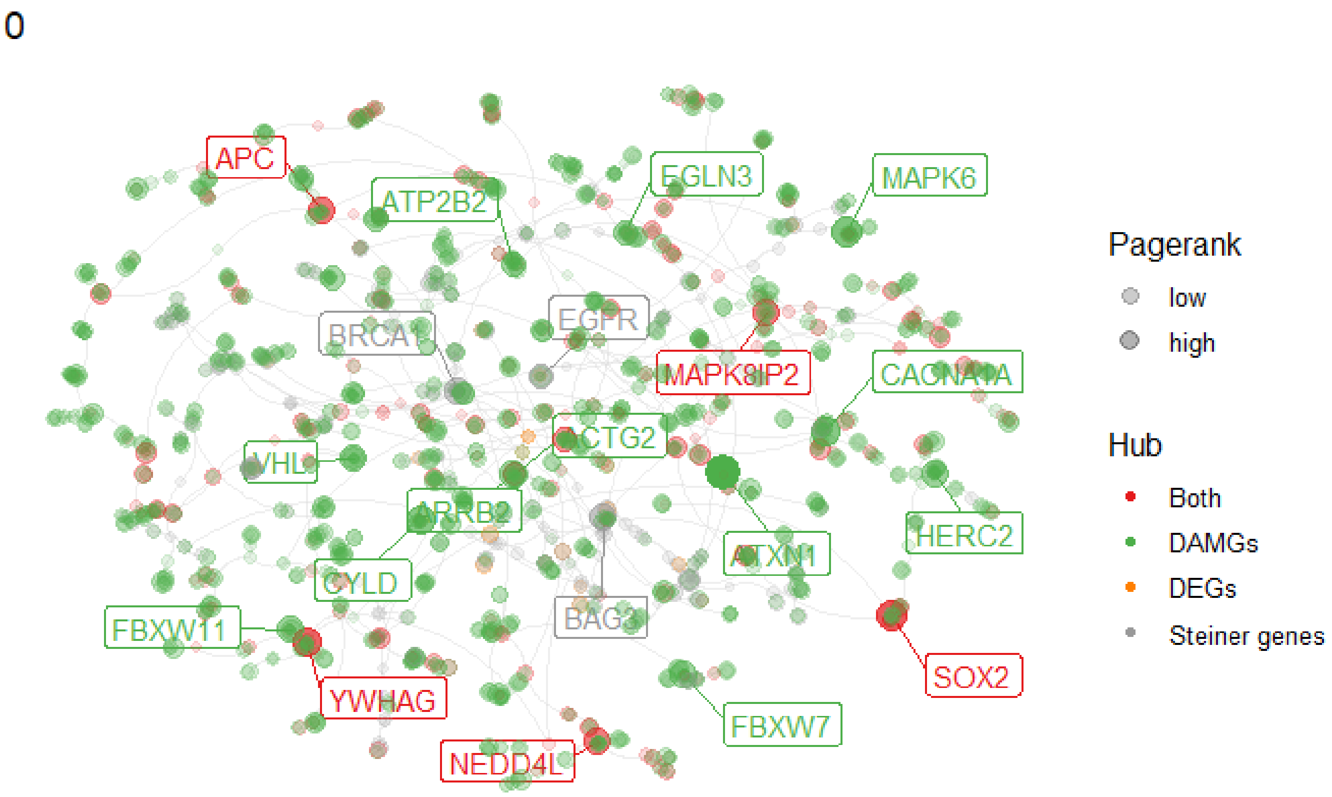


You can also show steiner tree of given cluster

Hide

```
PlotSteinertree(HECA_seurat, ident = '0')
```

```
calculate tree
```


0
APC
ATP2B2
EGLN3
MAPK6
BRCA1
EGFR
MAPK8IP2
CACNA1A
VHL
ACTG2
ARRB2
ATXN1
HERC2
CYLD
BAG3
FBXW11
YWHAG
SOX2
FBXW7
NEDD4L
Pagerank
low
high
Hub
Both
DAMGs
DEGs
Steiner genes