Cell Type Specific Differential Expression Gene Test

Weixu Wang

2022/4/21

In this tutorial, we illustrate how to identify the cell type-specific expression gene through ENIGMA, the example datasets could be downloaded from https://github.com/WWXkenmo/ENIGMA/tree/main/ENIGMA_analysis

load datasets and create ENIGMA object

We first load the datasets and create enigma object, the reference data is provided, so ref_type = "aggre", onething we need to note that both our Bulk and Ref datasets are simulated according to the simulation model which is described in supplementary note of our manuscript. And the Bulk and Ref datasets are simulated from gaussian distribution, so we no longer need to perform additional preprocessing.

```
library(ENIGMA)
load("exampleDatasets.Rdata")

egm = create_ENIGMA(bulk = Bulk,ref = Ref,ref_type = "aggre")
```

```
## Thu Apr 21 11:39:08 2022 Reference from aggregated FACS/Sort bulk RNA-seq/microarray/scRNA-seq.
```

```
## Warning in create_ENIGMA(bulk = Bulk, ref = Ref, ref_type = "aggre"): Reference
## matrix seems to be in log space. Please check if the reference matrix is
## normalized count data.
```

```
egm = get_cell_proportion(egm,method = "RLR")
```

```
## Thu Apr 21 11:39:08 2022 Calculating cell type proportion of bulk samples...
## Using Robust Linear Regression...
```

Perform deconvolution

According our benchmark results, the ENIGMA_trace is perform better than ENIGMA maximum L2 norm model on cell type specific DEG (CTS-DEG) identification, therefore, we suggested user to use ENIGMA_trace_norm to perform deconvolution when they need to know the CTS-DEG.

```
egm = ENIGMA_trace_norm(egm, solver = "proximalpoint",pos=FALSE,preprocess = "none",alpha=0.1)
```

```
## Loading required package: corpcor
```

```
## Using proximalpoint solver...
## Normalization...
## Converge in 129 steps
```

```
## the Bulk has negative value so we set pos=FALSE,
## in real senario we suggested user to set pos=TRUE,
```

```
## if there has no specific normalization procedure like Z-score,
## which would introduce negative value.
```

Perform CTS-DE analysis

Now we could perform CTS-DE analysis, through input the enigma object and phenotype vector.(1:case,0:control)

```
p <- 100
y <- as.numeric(gl(2, p/2)) - 1
DEG = FindCSE_DEG(egm,y)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Using Linear Regression To Infer Probability of Expression...
## Bootstrapping...
## Refining...
```

The output is the list object contain the DEG results,

```
head(DEG$CellType1)
```

```
##          ExpressionDifference       pvalue       qvalue sig_gene sig_gene_p
## gene1041            2.0398226 0.0006262799 7.145539e-14        1      0.001
## gene1669            1.3198390 0.0015409446 2.257215e-09        1      0.034
## gene30              1.2909966 0.0003718263 1.350724e-07        1      0.011
## gene53              1.2453592 0.0026005305 2.277815e-06        1      0.019
## gene1183            0.8724573 0.0068823824 4.043939e-04        1      0.032
## gene1289            0.5502680 0.0081641946 1.425293e-01        1      0.031
```

The first column indicates the expression difference (mean expression of each gene in case - mean expression of each gene in normal), user could return fold change through DEG = FindCSE_DEG(egm,y,FoldChange=TRUE). The second and third column is the pvalue and FDR adjusted pvalue for DEG analysis. The forth column indicates if this gene is the gene which has high likelihood to be accurated estimated within this cell type (1:yes, 0:no). And the fiveth column represents the pvalue of statistical significance of gene expression in the cell type.