Introduction to scLink

Wei Vivian Li, Rutgers Department of Biostatistics and Epidemiology

2020-07-17

A system-level understanding of the regulation and coordination mechanisms of gene expression is essential to understanding the complexity of biological processes in health and disease. With the rapid development of single-cell RNA sequencing technologies, it is now possible to investigate gene interactions in a cell-type-specific manner. Here we introduce the `scLink` package, which uses statistical network modeling to understand the co-expression relationships among genes and to construct sparse gene co-expression networks from single-cell gene expression data.

Here we demonstrate the functionality of `scLink` using the example data stored in the package.

```
library(scLink)
```

```
## Loading required package: parallel
```

```
count = readRDS(system.file("extdata", "example.rds", package = "scLink"))
genes = readRDS(system.file("extdata", "genes.rds", package = "scLink"))
```

The example raw count matrix `count` has 793 **rows** representing different cells and 23,341 **columns** representing different genes. `genes` is a character vector of 500 genes of interest.

sclink_norm

We use the function `sclink_norm` to process single cell read count for application of the sclink method. The code below will normalize the read count matrix with a library size of 10^6 and only keep the 500 genes in `genes` for downstream analysis. Note that the normalized count matrix `count.norm` is on the $\log_{10}$ scale.

```
count.norm = sclink_norm(count, scale.factor = 1e6, filter.genes = FALSE, gene.names = genes)
```

If users do not have a particular gene list for network inference, they can set `filter.genes=TRUE` to filter for the top n genes with largest average expression values. For example:

```
count.norm = sclink_norm(count, scale.factor = 1e6, filter.genes = TRUE, n = 500)
```

sclink_net

After the pre-processing step, we use the function `sclink_net` to calculate the robust correlation matrix and identifed sparse co-expression network of scLink. `expr` is the normalized count matrix output by `sclink_norm` or supplied by the users. `lda` is the candidate regularization parameters used in scLink's graphical model. The users can set `ncores` to take advantage of parallel computation.

```
networks = sclink_net(expr = count.norm, ncores = 1, lda = seq(0.5, 0.1, -0.05))
```

`sclink_net` returns a list of results. The scLink's robust correlation matrix can be retrieved from the `cor` element:

```
networks$cor[1:3,1:3]
```

```
##              Rn45s      Eef1a1      Malat1
## Rn45s   1.00000000 -0.27604002 -0.08561265
## Eef1a1 -0.27604002  1.00000000 -0.05138179
## Malat1 -0.08561265 -0.05138179  1.00000000
```

The gene co-expression networks and summary statistics can be retrieved from the `summary` element, which is a list with the same length as `lda`: each element corresponds to one regularization parameter.

```
net1 = networks$summary[[1]]
names(net1)
```

```
## [1] "adj"    "Sigma"  "nedge"  "bic"    "lambda"
```

```
### adjacency matrix
net1$adj[1:3,1:3]
```

```
##        Rn45s Eef1a1 Malat1
## Rn45s      1      0      0
## Eef1a1     0      1      0
## Malat1     0      0      1
```

```
### concentration matrix
net1$Sigma[1:3,1:3]
```

```
##           Rn45s   Eef1a1   Malat1
## Rn45s  1.696584 0.000000 0.000000
## Eef1a1 0.000000 1.665809 0.000000
## Malat1 0.000000 0.000000 1.578271
```

```
### BIC
net1$bic
```

```
## [1] 1255450
```

```
### number of edges
net1$nedge
```

```
## [1] 127
```

```
### regularization parameter lambda
net1$lambda
```

```
## [1] 0.5
```

sclink_cor

Since it is very difficult to infer co-expression relationships for lowly expressed genes in single-cell data, we suggest the filtering step as used in `sclink_norm` to select genes. This also reduces the computational burden. However, if the users would like to infer gene networks for a large gene list (e.g., > 5000 genes), we suggest that the users first use `sclink_cor` to investigate the correlation structures among these genes.

```
corr = sclink_cor(expr = count.norm, ncores = 1)
```

If the correlation matrix suggests obvious gene modules, then the users can apply `sclink_net` separately on these modules to reduce computation time and increase overall accuracy.