ISA Bi-Clustering

Uwe Schwartz

13 8 2017

Dependencies

```
library(eisa)
library(Biobase)
library(genefilter)
library(org.Hs.eg.db)
```

Read RPKM-count tables

First we load the RPKM-count tables obtained from Moffa et al. In this study RNA-expression levels were quantified after single gene knock-down of WNT-signaling pathway components in HCT116 cells. Each table represents one biological replicate. In this study our focus lies on APC, CTNNB1 and EVI knock-downs and the control sample.

```
rna1<-read.table("data/RNA-seq/SA00011.seq0005.rpkm", header=T, row.names=1)
rna2<-read.table("data/RNA-seq/SA00011.seq0006.rpkm", header=T, row.names=1)

# mark second replicate
colnames(rna2)<- paste(colnames(rna2), "2", sep="_")
# select samples of interest and convert them to an ExpressionSet object
rnaseq<-cbind(rna1[,c(1,4,5,6)],rna2[,c(1,4,5,6)])
ma.rna<-as.matrix(rnaseq)
exp.set<-new("ExpressionSet", exprs=ma.rna, annotation="org.Hs.eg")
```

Non-specific filtering

Next, we remove not expressed genes and genes exhibiting a low variability across all samples.

```
# gene has at least in 2 samples RPKM-values >3
kLimit<-2
ALimit<-3
# gene has at least a variance of 0.5
varLimit<-0.5
# filter ExpressionSet
flist <- filterfun(function(x) var(x)>varLimit, kOverA(kLimit,ALimit))
exp.set.fil <- exp.set[genefilter(exp.set, flist), ]
```

Data normalization

Before clutsering the expression matrix is scaled and centered. Two normalized matrices are generated: the gene-wise (row-wise) normalized and the sample-wise (column-wise) normalized expression matrix.

```
norm.ISA<-ISANormalize(exp.set.fil)
```

Iterative Bi-Clustering

The Algorithm starts with a random set of genes and will refine iteratively this input set. First those conditions are identified where the gene set exhibits a high correlation, then genes are removed/included that are well/poorly co-expressed within the selected conditions. This process is iterated until the gene expression signature is stable and does not change anymore in this process. This procedure is controled by two user-defined thresholds thr.feat and thr.sample, which define how closely related are the genes and the conditions respectively. We start with 300 randomly generated gene sets. Next ISA is applied and co-regulated modules are collected. Finally duplicated modules are removed.

```
random.seed <- generate.seeds(length=nrow(norm.ISA), count=3)
#Bi-Clustering
module_run <- ISAIterate(norm.ISA, feature.seeds=random.seed,
                         thr.feat=1.75, thr.samp=0.45, convergence="cor")
# Remove duplicated modules
modules.unique <- ISAUnique(norm.ISA, module_run)
```

Session info

```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.12.6 (Sierra)
##
## locale:
## [1] de_DE.UTF-8/de_DE.UTF-8/de_DE.UTF-8/C/de_DE.UTF-8/de_DE.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
## [1] org.Hs.eg.db_3.3.0   genefilter_1.54.2    eisa_1.24.0
## [4] AnnotationDbi_1.34.4 IRanges_2.6.1        S4Vectors_0.10.3
## [7] Biobase_2.32.0       BiocGenerics_0.18.0  isa2_0.3.5
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.12    bitops_1.0-6    tools_3.3.1     digest_0.6.12
##  [5] bit_1.1-12      annotate_1.50.1 RSQLite_2.0     evaluate_0.10.1
##  [9] memoise_1.1.0   tibble_1.3.3    lattice_0.20-35 pkgconfig_2.0.1
## [13] rlang_0.1.2     Matrix_1.2-11   graph_1.50.0    DBI_0.7
## [17] Category_2.38.0 yaml_2.1.14     stringr_1.2.0   knitr_1.17
## [21] rprojroot_1.2   bit64_0.9-7     grid_3.3.1      GSEABase_1.34.1
## [25] survival_2.41-3 XML_3.98-1.9    RBGL_1.48.1     rmarkdown_1.6
## [29] blob_1.1.0      magrittr_1.5    splines_3.3.1   backports_1.1.0
## [33] htmltools_0.3.6 xtable_1.8-2    stringi_1.1.5   RCurl_1.95-4.8
```