Get Started

Kim-Anh Le Cao

13 September 2016

Contents

Read data

Affymetrix data

```
data = read.table('Data/resnorm_B46.txt', header = TRUE, sep = '\t', row.names = 1)
dim(data)
```

```
## [1] 32423    48
```

```
#colnames(data)
#data[1:5, 1:5]

# check normalisation: ok
# d = density(data[,1])
# plot(d)
# for(k in 2:ncol(data)){
#   d = density(data[,k])
#   lines(d)
# }

# filter out genes
var.genes = apply(data, 1, var)
#hist(var.genes)
keep.genes = names(sort(var.genes, decreasing = TRUE)[1:5000])

data.gene = t(data[keep.genes, ])
dim(data.gene)
```

```
## [1]   48 5000
```

```
# name of genes
name.gene = read.csv('Data/Table_S2.csv', row.names = 1)
dim(name.gene)
```

```
## [1] 32423    82
```

```
keep.name.genes = as.character(name.gene[keep.genes, "Name.Hu_AGI"])
# change empty to Hu_empty
keep.name.genes[keep.name.genes == 'EMPTY'] = 'Hu_empty'

#keep.name.genes can be used for example in the network vis / circos vis / plotVar to display the gene
```

Physio data

```
data.physio = read.table('Data/physio_B46.txt', header = TRUE, row.names = 1)
dim(data.physio)
```

```
## [1] 48 10
```

```
#data.physio[1:5,1:5]
```

Set up data for multivariate analyses

Summary of stress x genotype conditions:

```
# checking rownames match
# data.frame(rownames(data.gene), rownames(data.physio))

genotype = factor(substr(rownames(data.gene), 1, 5))
summary(genotype)
```

```
## Inedi Melod SF028 SF107 SF109 SF193 SF326 Tekny
##     6     6     6     6     6     6     6     6
```

```
trt = factor(c(rep(c('ctrl', 'stress'), 24)))
summary(trt)
```

```
##   ctrl stress
##     24     24
```

```
pch.trt = rep(16, length(trt))
pch.trt[trt == 'stress'] = 17
# checking
#data.frame(rownames(data.gene), trt)

table(genotype, trt)
```

```
##         trt
## genotype ctrl stress
##    Inedi    3      3
##    Melod    3      3
```

```
##    SF028    3      3
##    SF107    3      3
##    SF109    3      3
##    SF193    3      3
##    SF326    3      3
##    Tekny    3      3
```

Create separate legend for slides, then crop by hand

```
library(mixOmics)
pdf('Figures/legend.pdf')
plot(0, 0, type = 'n')
legend(-1, 0, legend = c(levels(trt), levels(genotype)), col = c('black', 'black', color.mixo(1:nlevels
dev.off()
```

```
## pdf
##   2
```

```
pdf('Figures/legend-trt.pdf')
plot(0, 0, type = 'n')
legend(-1, 0, legend = c(levels(trt)), col = c('black', 'black'), pch = c(16, 17), lwd = 2, cex = 2, nc
dev.off()
```

```
## pdf
##   2
```

The tuning for sPLS-DA is available for list.keepX = c(2:10, seq(20, 200, 10)) and ncomp = 1, nrepeat = 50, folds = 5 (takes ~ 10 min to run):

splsda.tune.gene = tune.splsda(data.gene, ncomp = 1, Y = trt, test.keepX = list.keepX, validation = 'Mfold', folds = 5, nrepeat = 50)