Lecture 12

BIOF 339

December 5, 2016

Bioconductor

Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data, using R.

- 1296 packages
- Covers the bioinformatic pipeline

- Software
- Annotation
- Experiments

Explore Bioconductor website

Installing Bioconductor packages

This is different from the usual `install.packages`

```
source('http://bioconductor.org/biocLite.R')
biocLite('Biobase','limma','hgu95av2.db')
```

Data in Bioconductor

The basic structure in a Bioconductor pipeline is the `ExpressionSet`

```
library(Biobase)
str(sample.ExpressionSet)
```

```
#  Formal class 'ExpressionSet' [package "Biobase"] with 7 slots
#    ..@ experimentData   :Formal class 'MIAME' [package "Biobase"] with 13 slots
#    .. .. ..@ name              : chr "Pierre Fermat"
#    .. .. ..@ lab               : chr "Francis Galton Lab"
#    .. .. ..@ contact           : chr "pfermat@lab.not.exist"
#    .. .. ..@ title             : chr "Smoking-Cancer Experiment"
#    .. .. ..@ abstract          : chr "An example object of expression set (Expressi
#    .. .. ..@ url               : chr "www.lab.not.exist"
#    .. .. ..@ pubMedIds         : chr ""
#    .. .. ..@ samples           : list()
#    .. .. ..@ hybridizations    : list()
#    .. .. ..@ normControls      : list()
#    .. .. ..@ preprocessing     : list()
#    .. .. ..@ other             :List of 1
#    .. .. .. ..$ notes: chr "An example object of expression set (exprSet) class"
```

Differences with usual R

Instead of storing data in named lists, ExpressionSet objects store data in slots, and we can see what the slots are with `slotNames`:

```
slotNames(sample.ExpressionSet)
```

```
#  [1] "experimentData"    "assayData"          "phenoData"
#  [4] "featureData"       "annotation"         "protocolData"
#  [7] ".__classVersion__"
```

Differences with usual R

You can access these slots using `@`, instead of the usual `$`:

```
sample.ExpressionSet@phenoData
```

```
#  An object of class 'AnnotatedDataFrame'
#    sampleNames: A B ... Z (26 total)
#    varLabels: sex type score
#    varMetadata: labelDescription
```

Differences with usual R

However, it's much easier to go with the built-in functions

```
pData(sample.ExpressionSet)
```

```
#      sex    type score
# A Female Control  0.75
# B   Male    Case  0.40
# C   Male Control  0.73
# D   Male    Case  0.42
# E Female    Case  0.93
# F   Male Control  0.22
# G   Male    Case  0.96
# H   Male    Case  0.79
# I Female    Case  0.37
# J   Male Control  0.63
# K   Male    Case  0.26
# L Female Control  0.36
# M   Male    Case  0.41
# N   Male    Case  0.80
# O Female    Case  0.10
```

Differences with usual R

```
head(exprs(sample.ExpressionSet))
```

```
#                          A         B        C        D        E       F
#  AFFX-MurIL2_at  192.7420  85.75330 176.7570 135.5750 64.49390 76.3569
#  AFFX-MurIL10_at  97.1370 126.19600  77.9216  93.3713 24.39860 85.5088
#  AFFX-MurIL4_at   45.8192   8.83135  33.0632  28.7072  5.94492 28.2925
#  AFFX-MurFAS_at   22.5445   3.60093  14.6883  12.3397 36.86630 11.2568
#  AFFX-BioB-5_at   96.7875  30.43800  46.1271  70.9319 56.17440 42.6756
#  AFFX-BioB-M_at   89.0730  25.84610  57.2033  69.9766 49.58220 26.1262
#                          G       H       I         J        K       L
#  AFFX-MurIL2_at  160.5050 65.9631 56.9039 135.60800 63.44320 78.2126
#  AFFX-MurIL10_at  98.9086 81.6932 97.8015  90.48380 70.57330 94.5418
#  AFFX-MurIL4_at   30.9694 14.7923 14.2399  34.48740 20.35210 14.1554
#  AFFX-MurFAS_at   23.0034 16.2134 12.0375   4.54978  8.51782 27.2852
#  AFFX-BioB-5_at   86.5156 30.7927 19.7183  46.35200 39.13260 41.7698
#  AFFX-BioB-M_at   75.0083 42.3352 41.1207  91.53070 39.91360 49.8397
#                         M       N       O       P        Q        R
#  AFFX-MurIL2_at  83.0943 89.3372 91.0615 95.9377 179.8450 152.4670
#  AFFX-MurIL10_at 75.3455 68.5827 87.4050 84.4581  87.6806 108.0320
#  AFFX-MurIL4_at  20.6251 15.9231 20.1579 27.8139  32.7911  33.5292
```

Making a heatmap

Heatmaps

There are several ways of doing heatmaps in R:

- http://sebastianraschka.com/Articles/heatmaps_in_r.html
- https://plot.ly/r/heatmaps/
- http://moderndata.plot.ly/interactive-heat-maps-for-r/
- http://www.siliconcreek.net/r/simple-heatmap-in-r-with-ggplot2
- https://rud.is/b/2016/02/14/making-faceted-heatmaps-with-ggplot2/

Some example data

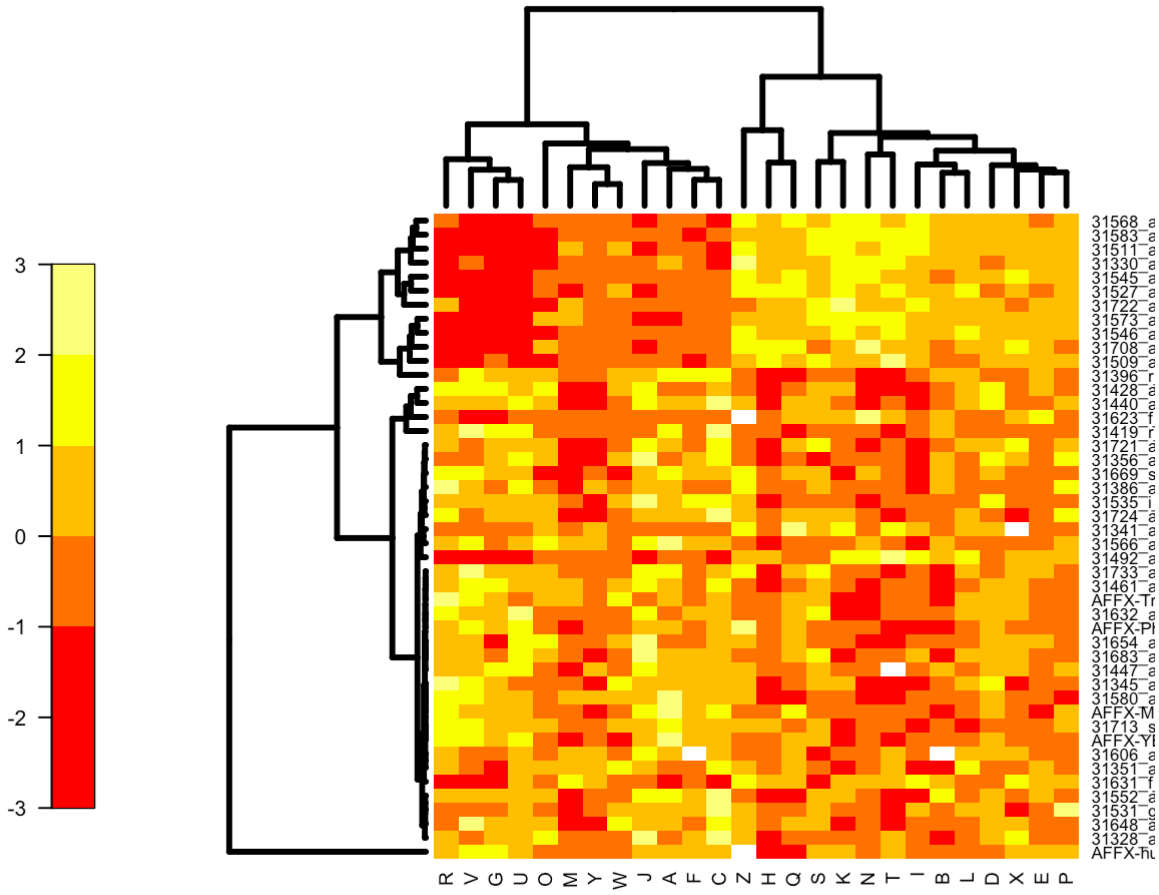
```
library(Biobase)
data(sample.ExpressionSet)
exdat <-  sample.ExpressionSet
library(limma)
design1 <- model.matrix(~type, data=pData(exdat))
lm1 <- lmFit(exprs(exdat), design1)
lm1 <- eBayes(lm1) # compute linear model for each probeset
geneID <- rownames(topTable(lm1, coef=2, num=100, adjust='none',p.value=0.05))
exdat2 <- exdat[geneID,] # Keep features with p-values < 0.05
exdat2
```

```
#  ExpressionSet (storageMode: lockedEnvironment)
#  assayData: 46 features, 26 samples
#    element names: exprs, se.exprs
#  protocolData: none
#  phenoData
#    sampleNames: A B ... Z (26 total)
#    varLabels: sex type score
#    varMetadata: labelDescription
#  featureData: none
```

Heatmaps using **`Heatplus`**

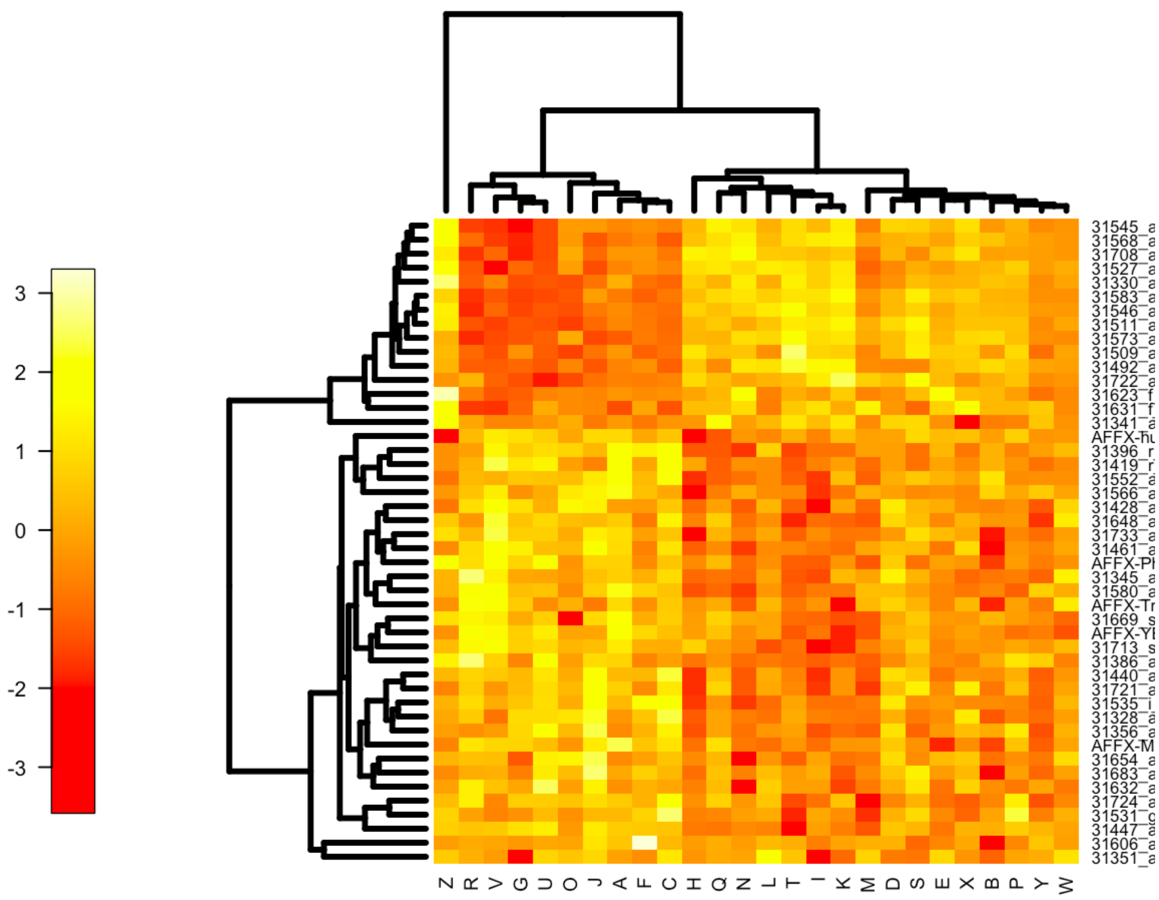
```
source('http://bioconductor.org/biocLite.R')
biocLite('Heatplus')
```

```
library(Heatplus)
reg1 <- regHeatmap(exprs(exdat2), legend=2, col=heat.colors,
                   breaks=-3:3)
plot(reg1)
```


3
2
1
0
-1
-2
-3
31568_a
31583¯a
31511¯a
31330¯a
31545¯a
31527¯a
31722¯a
31573¯a
31546¯a
31708¯a
31509¯a
31396¯r
31428¯a
31440¯a
31623¯f
31419¯r
31721¯a
31356¯a
31669¯s
31386¯a
31535¯i
31724¯a
31341¯a
31566¯a
31492¯a
31733¯a
31461¯a
AFFX-Tr
31632_a
AFFX-Pl
31654_a
31683¯a
31447¯a
31345¯a
31580¯a
AFFX-M
31713_s
AFFX-Yl
31606_a
31351¯a
31631¯f
31552¯a
31531¯g
31648¯a
31328¯a
AFFX-hu
R
V
G
U
O
M
Y
W
J
A
F
C
Z
H
Q
S
K
N
T
I
B
L
D
X
E
P

```r
library(Heatplus)
reg1 <- regHeatmap(exprs(exdat2), legend=2, col=heat.colors,
                   breaks=-3:3)
plot(reg1)
```

```r
corrdist <- function(x) as.dist(1-cor(x))
hclust.avl <-  function(x) hclust(x, method='average')
reg2 <- regHeatmap(exprs(exdat2), legend=2, col=heat.colors,
                   breaks=-3:3,
                   dendrogram = list(clustfun=hclust.avl, distfun=corrdist))
plot(reg2)
```


3
2
1
0
-1
-2
-3
31545_a
31568_a
31708_a
31527_a
31330_a
31583_a
31546_a
31511_a
31573_a
31509_a
31492_a
31722_a
31623_f
31631_f
31341_a
AFFX-hu
31396_r
31419_r
31552_a
31566_a
31428_a
31648_a
31733_a
31461_a
AFFX-Pl
31345_a
31580_a
AFFX-Tr
31669_s
AFFX-YI
31713_s
31386_a
31440_a
31721_a
31535_i
31328_a
31356_a
AFFX-M
31654_a
31683_a
31632_a
31724_a
31531_g
31447_a
31606_a
31351_a
Z
R
V
G
U
O
J
A
F
C
H
Q
N
L
T
I
K
M
D
S
E
X
B
P
Y
W

```
ann1 <- annHeatmap(exprs(exdat2), ann=pData(exdat2))
plot(ann1)
```

```
ann1 <- annHeatmap(exprs(exdat2), ann=pData(exdat2))
plot(ann1)
```

```
ann2 <- annHeatmap(exprs(exdat2), ann=pData(exdat2),
                   cluster = list(cuth=7500,
                                  label=c('Control-like','Case-like')))
plot(ann3)
```