Install and load packages and workflow

All information can be found here: https://www.bioconductor.org/help/workflows/arrays/#sample-workflow

At the beginning we need to install and load the packages that we need for our Microarray analysis.

- **affy** for Affymetrix pre-processing
- **limma** for two-color pre-processing; differential expression

To install those packages use:

```r
## try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite(c("affy", "limma"))
```

Now we can load these packages:

```r
library(affy)
library(limma)
```

To install the whole workflow type:

```r
source("http://bioconductor.org/workflows.R")
workflowInstall("arrays")
```

This installs all data that is needed by this workflow in your home: e.g. this could be `~/lib/R/library/arrays/extdata` or sth similar.

We would like to play with some sample data. So, first we import “phenotype” data, describing the experimental design

```r
phenoData <- read.AnnotatedDataFrame(system.file("extdata", "pdata.txt", package="arrays"))
```

```r
# An object of class 'AnnotatedDataFrame'
# rowNames: sample01.CEL sample02.CEL ... sample12.CEL (12 total)
# varLabels: IVT Sensitivity
# varMetadata: labelDescription
```

**Task 01:**

- have a look at the file `pdata.txt` to see what you read in
- examine `phenoData` to see what you got in R

**Read data and create an Expression Set (RMA normalization)**

The Affymetrix data are originally stored in files of type `.CEL` and `.CDF.
We use the function `justRMA` to read such CEL files and compute an expression measure into an R data structure `ExpressionSet`, integrating our phenotype data from `phenoData`. 
**Task 02:**

- have a look at the directory `celfiles` to see how many files (samples) you read in
- examine `eset` to see what you got in R

### Differential Expression Analysis

First, combine the two phenotype data columns into the single one `combn` and use it as the design of the model to be fit.

```r
combn <- factor(paste(pData(phenoData)[,1], pData(phenoData)[,2], sep = "_"))
design <- model.matrix(~combn)
```

Now, fit a linear model for each gene given a series of arrays (`eset`). Use this fit to compute moderated t-statistics, moderated F-statistic, and log-odds of differential expression by empirical Bayes moderation of the standard errors towards a common value.

```r
fit <- lmFit(eset, design)  # fit each probeset to model
efit <- eBayes(fit)          # empirical Bayes adjustment
```

**Task 03:**

- examine `fit` describe what you see
- examine `efit` what is different/equal to `fit`?

Now, extract a table of the (10) top-ranked genes w.r.t. differential expression from the linear model fit.

```r
topTable(efit, coef=2)  # table of differentially expressed probesets
```

```
## logFC AveExpr t    P.Value      adj.P.Val
## 204582_s_at  3.468416 10.150533 39.03471 1.969915e-14 1.732146e-10
## 211548_s_at -2.325670  7.178610 -22.73165 1.541158e-11 6.775701e-08
## 216598_s_at  1.936306  7.692822  21.73818 2.658881e-11 7.793180e-08
## 211110_s_at  3.157766  7.909391  21.19204 3.625216e-11 7.969130e-08
## 206001_at   -1.590732 12.402722 -18.64398 1.715422e-10 3.016740e-07
## 202409_at   -1.125780  6.704989 -17.72512 1.692923e-10 3.621657e-07
## 221019_s_at  2.315730  7.304912  16.34552 8.352833e-10 1.034929e-06
## 213048_s_at  1.183051  7.125307  14.75281 2.834343e-09 3.165297e-06
## 209288_s_at -1.226421  7.603917 -13.32681 9.401074e-09 7.784531e-06
## B
```

<table>
<thead>
<tr>
<th>gene</th>
<th>logFC</th>
<th>AveExpr</th>
<th>t</th>
<th>P.Value</th>
<th>adj.P.Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>204582_s_at</td>
<td>3.468416</td>
<td>10.150533</td>
<td>39.03471</td>
<td>1.969915e-14</td>
<td>1.732146e-10</td>
</tr>
<tr>
<td>211548_s_at</td>
<td>-2.325670</td>
<td>7.178610</td>
<td>-22.73165</td>
<td>1.541158e-11</td>
<td>6.775701e-08</td>
</tr>
<tr>
<td>216598_s_at</td>
<td>1.936306</td>
<td>7.692822</td>
<td>21.73818</td>
<td>2.658881e-11</td>
<td>7.793180e-08</td>
</tr>
<tr>
<td>211110_s_at</td>
<td>3.157766</td>
<td>7.909391</td>
<td>21.19204</td>
<td>3.625216e-11</td>
<td>7.969130e-08</td>
</tr>
<tr>
<td>206001_at</td>
<td>-1.590732</td>
<td>12.402722</td>
<td>-18.64398</td>
<td>1.715422e-10</td>
<td>3.016740e-07</td>
</tr>
<tr>
<td>202409_at</td>
<td>-1.125780</td>
<td>6.704989</td>
<td>-17.72512</td>
<td>1.692923e-10</td>
<td>3.621657e-07</td>
</tr>
<tr>
<td>221019_s_at</td>
<td>2.315730</td>
<td>7.304912</td>
<td>16.34552</td>
<td>8.352833e-10</td>
<td>1.034929e-06</td>
</tr>
<tr>
<td>213048_s_at</td>
<td>1.183051</td>
<td>7.125307</td>
<td>14.75281</td>
<td>2.834343e-09</td>
<td>3.165297e-06</td>
</tr>
<tr>
<td>209288_s_at</td>
<td>-1.226421</td>
<td>7.603917</td>
<td>-13.32681</td>
<td>9.401074e-09</td>
<td>7.784531e-06</td>
</tr>
</tbody>
</table>

2
The p-values for the coefficient/contrast of interest are adjusted for multiple testing by a call to `p.adjust`. The “BH” method, which controls the expected false discovery rate (FDR) below the specified value, is the default adjustment method because it is the most likely to be appropriate for microarray studies. Note that the adjusted p-values from this method are bounds on the FDR rather than p-values in the usual sense. Because they relate to FDRs rather than rejection probabilities, they are sometimes called q-values. See `help("p.adjust")` for more information.

```
sessionInfo()
```