Micorarrays - Marray, sourcing

Source: https://github.com/lgatto/S4-tutorial

Using the script at `./R/Marray.R`.

```
# Create a matrix and two dataframes

n <- 10
m <- 6

marray <- matrix(rnorm(n * m, 10, 5), ncol = m)
rownames(marray) <- paste0("probe", 1:n)
colnames(marray) <- LETTERS[1:m]

# build the annotations dataframe for the probes
fmeta <- data.frame(geneId = 1:n, pathway = sample(LETTERS, n, replace = TRUE))
rownames(fmeta) <- rownames(marray)

pmeta <- data.frame(sampleId = 1:m, condition = rep(c("WT", "MUT"), each = 3))
rownames(pmeta) <- colnames(marray)

maexp <- list(marray = marray,
              fmeta = fmeta,
              pmeta = pmeta)
```

```
source("./R/Marray.R")        # load the class

# we use the previous list
ma <- MArray(marray = maexp[[1]],       # load the data
             pmeta = maexp[["pmeta"]],
             fmeta = maexp[["fmeta"]]
             )
ma
```

```
## An object of class MArray
##  10 features by 6 samples.
```

```
# show the array
marray(ma)
```

```
##                 A         B         C         D         E         F
## probe1  12.265834 14.556787 19.020727  8.038573 12.272549 13.319418
## probe2  19.733729 21.331377 10.630566  5.129016  5.089732 10.757785
## probe3   6.717919 11.343487  4.897526  9.150322 12.628082 12.314539
## probe4  14.910788 14.919327  8.484904  5.216278  9.475401 12.866106
## probe5  12.323055 16.555086 12.706192 15.691746 13.795273 17.061138
## probe6  15.449977 11.349991  9.254074 -1.854662  5.892223  8.518018
## probe7  12.748571 13.050343 11.175370 12.536445 10.308160 13.786911
## probe8  13.911874 19.578958  3.959002 11.929915 -2.513701 15.108463
## probe9   7.383131 11.541271 17.759639  9.225504  4.818971 21.836223
## probe10  9.367797  2.046491  7.051425  6.665827  6.035357  2.264786
```

```
# show the dataframes
fmeta(ma)
```

```
##         geneId pathway
## probe1        1       X
## probe2        2       Q
## probe3        3       N
## probe4        4       H
## probe5        5       B
## probe6        6       I
## probe7        7       F
## probe8        8       N
## probe9        9       Z
## probe10      10       T
```

```
pmeta(ma)
```

```
##   sampleId condition
## A        1        WT
## B        2        WT
## C        3        WT
## D        4       MUT
## E        5       MUT
## F        6       MUT
```

```
# subsetting
ma[1:5, 1:3]
```

```
## An object of class MArray
##  5 features by 3 samples.
```

```
# a summary for sample "D"
summary(ma, "D")
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.855   5.579   8.594   8.173  11.254  15.692
```

```
# for the condition "WT" in the sample annotations,
# get the array values for probe8
wt <- maexp$pmeta[, "condition"] == "WT"
maexp$marray["probe8", wt]
```

```
##         A         B         C
## 13.911874 19.578958  3.959002
```

```
# same but shorter
getValuesAtProbeForCondition(ma, "probe3", "WT")
```

```
##         A         B         C
##  6.717919 11.343487  4.897526
```

```
# getValuesAtSampleForPathway
getValuesAtSampleForPathway(ma, "C", "A")    # one case
```

```
## numeric(0)
```

```
getValuesAtSampleForPathway(ma, "C", "X")    # two cases
```

```
## [1] 19.02073
```

```
getSampleConditions(ma)
```

```
## [1] WT  MUT
## Levels: MUT WT
```

```
getProbePathways(ma)
```

```
## [1] X Q N H B I F Z T
## Levels: B F H I N Q T X Z
```

Introspection

```
slotNames(ma)
```

```
## [1] "marray" "fmeta"  "pmeta"
```

```
getClass("MArray")
```

```
## Class "MArray" [in ".GlobalEnv"]
## 
## Slots:
## 
## Name:      marray      fmeta      pmeta
## Class:     matrix data.frame data.frame
```

```
showMethods(classes = "MArray")
```

```
## 
## Function ".DollarNames":
##  <not an S4 generic function>
## Function: [ (package base)
## x="MArray"
## 
## 
## Function "complete":
##  <not an S4 generic function>
## Function: fmeta (package .GlobalEnv)
## object="MArray"
## 
## 
## Function "formals<-":
##  <not an S4 generic function>
## 
## Function "functions":
##  <not an S4 generic function>
## Function: getProbePathways (package .GlobalEnv)
## object="MArray"
## 
## Function: getSampleConditions (package .GlobalEnv)
## object="MArray"
## 
## Function: getValuesAtProbeForCondition (package .GlobalEnv)
## object="MArray"
## 
## Function: getValuesAtSampleForPathway (package .GlobalEnv)
## object="MArray"
## 
## Function: initialize (package methods)
## .Object="MArray"
##     (inherited from: .Object="ANY")
##
## Function: marray (package .GlobalEnv)
## object="MArray"
##
## Function: pmeta (package .GlobalEnv)
## object="MArray"
##
##
## Function "prompt":
##  <not an S4 generic function>
## Function: show (package methods)
## object="MArray"
##
## Function: summary (package base)
## object="MArray"
```