matchprobes

October 5, 2010

---

combineAffyBatch   *A function to combine data from different Affymetrix genechip types into one AffyBatch.*

Description

The function takes a list of AffyBatches and their respective probe sequence information and merges the intensities from the matching probes only into one AffyBatch.

Usage

```
combineAffyBatch(batch, probepkg, newcdf, verbose=TRUE)
```

Arguments

batch	A list of AffyBatches.
probepkg	A character vector of the same length as batch, containing the names of the probe sequences libraries (for example, hgu133aprobe) that go with the Affy-Batches.
newcdf	Character. The name of the name CDF environment that is to be created.
verbose	Logical. If TRUE, messages are printed to the console.

Details

Note that batch effects may play an important role, e.g. because of slightly but systematically different reagents or sample treatments in the experiments at led to the data in the different AffyBatches. Thus, please interpret the results of this function with caution.

Probe to probe-set mapping: after the common probes between all AffyBatches have been identified, their mapping into probe sets (=the new CDF environment) is obtained from the restriction of the mapping of the first element of batch (see variable REFCHIP in the code).

If you want to use this function, please have a look at its code and if necessary, check intermediate results.

Value

A list with two elements: dat, an AffyBatch, and cdf, an environment that contains the probe-set to probe mapping.

The phenoData slot of dat will be empty. This is because in most cases this requires some manual interference. You will need to construct a combined phenoData slot by yourself.

Author(s)

R. Gentleman, Wolfgang Huber

Examples

```
## Not run:
## WARNING: This example is broken because the 118T1.cel file doesn't
## contain the ScanDate info required by read.affybatch() since affy
## 1.23.2.

## In this example, we show how to combine the data from two different
## array types whose reporter sequences are partially the same.

library("affy")
library("hu6800cdf")
library("hu6800probe")

## Load the data:
f1 <- system.file("extdata", "118T1.cel", package="matchprobes")
f2 <- system.file("extdata", "CL2001032020AA.cel", package="matchprobes")
pd1 <- new("AnnotatedDataFrame", data=data.frame(fromFile=I(f1), row.names="f1"))
pd2 <- new("AnnotatedDataFrame", data=data.frame(fromFile=I(f2), row.names="f2"))
x1 <- read.affybatch(filenames=f1, compress=TRUE, phenoData=pd1)
x2 <- read.affybatch(filenames=f2, compress=TRUE, phenoData=pd2)

## 'f1' and 'f2' contain the filenames of the CEL files. In practice,
## they will be vectors with many filenames per array type, here,
## for demonstration, we only use one single CEL file per array type,
## which are provided in the 'extdata' subdirectory of the package.

## Combine the data. For this to work it is required that the packages
## hu6800probe and hgu95av2probe are installed:
res <- combineAffyBatch(list(x1, x2), c("hu6800probe","hgu95av2probe"), newcdf="comb")

## The function returns a list 'res' with two elements: an AffyBatch
## object (dat) and a CDF enviroment (cdf):
comb <- res$cdf
z    <- rma(res$dat)

## First, let us look at the distribution of the number of common
## probes per probeset:
prs <- mget(ls(comb), comb, ifnotfound=NA)
nrprobes <- sapply(prs, function(x) nrow(x))
barplot(table(nrprobes),
   xlab="number of probes per probeset",
   ylab="frequency")

## Let us also make scatterplots of the two combined arrays, first on
## the probe and then at the probe set level. This plot shows the
## comparison of the intensity data from common probes (left panel)
## and probesets (right panel) of the two array platforms:
png("matchprobes-scp.png", width=900, height=480)
par(mfrow=c(1,2))
plot(exprs(res$dat), main="after combine",
  pch=".", asp=1, xlab="f1", ylab="f2", log="xy")
plot(exprs(z), main="after RMA",
```

```
  pch=".", asp=1, xlab="f1", ylab="f2")
dev.off()

## Here, we explicitely used the png() device to write this plot to.
## PDF, for example, can become cumbersomely large in size for
## feature-rich plots such as this one.

## End(Not run)
```

Index