Handling probe sequence information: the package matchprobes (draft)

Wolfgang Huber and Robert Gentleman

October 28, 2003

Contents

1 Overview

This package allows to calculate with the sequences and related information of the probes on a microarray. The sequences themselves are stored in separate packages, one for each type of microarray. For some commonly used types, the packages can be downloaded from http://www.bioconductor.org. You can also create such packages yourself. This is described in section 4.

```
> library(matchprobes)
> library(affy)
> library(hgu95av2cdf)
> library(hgu95av2probe)
```

2 Using probe packages

The probe sequence data for the array type HG-U95Av2 (an Affymetrix GeneChip)
can be loaded with the command

```
> library(hgu95av2probe)
> help(hgu95av2probe)
```

2.1 Functions

Complementary sequence

```
> example(complementSeq)


cmplmS> seq <- c("AAACT", "GGGTT")

cmplmS> complementSeq(seq)
[1] "TTTGA" "CCCAA"

cmplmS> seq <- c("CGACTGAGACCAAGACCTACAACAG", "CCCGCATCATCTTTCCTGTGCTCTT")

cmplmS> complementSeq(seq, start = 13, stop = 13)
[1] "CGACTGAGACCATGACCTACAACAG" "CCCGCATCATCTATCCTGTGCTCTT"
```

Reverse sequence

```
> example(reverseSeq)


rvrsSq> w <- c("hey there", "you silly fool")

rvrsSq> reverseSeq(w)
[1] "ereht yeh"      "loof yllis uoy"

rvrsSq> w <- "able was I ere I saw Elba"
```

2

```
rvrsSq> reverseSeq(w)
[1] "ablE was I ere I saw elba"
```

Matching probes

```
> pm <- hgu95av2probe$sequence
> query <- hgu95av2probe$sequence[21:23]
> m <- matchprobes(query, pm)
> unlist(m)


match1 match2 match3
    21     22     23
```

2.2 Mismatch sequences for Affymetrix GeneChips

The mismatch sequences are not explicitely stored in the probe packages. They are implied by the match sequences, by flipping the middle base. This can be done with the complementSeq.

```
> mm <- complementSeq(hgu95av2probe$sequence, start = 13, stop = 13)
> cat(pm[1], mm[1], sep = "\n")


TCTCCTTTGCTGAGGCCTCCAGCTT
TCTCCTTTGCTGTGGCCTCCAGCTT
```

For Affymetrix GeneChips the length of the probe sequences is 25, and since we start counting at 1, the middle position is 13.

The function matchprobes also responds to this type of implied mismatch sequences, by multipying the matching indices with -1:

```
> m <- matchprobes(query, c(pm, mm))
> unlist(m)

 match1  match2  match3  match4  match5  match6
     21 -199105      22 -199106      23 -199107
```

Note: This may need to be set as an option, rather than as a standard behavior - I don't particularly like that the position 13 is hardcoded into matchprobes.

Best may be to drop the option altogether, since we can simply use complementSeq(query, start=13, stop=13) on the query string(s) and then call matchprobes with standard exact matching.

2.3   C-, G-content

The base content of the probes on an array can be obtained through the function
`basecontent`.

```
> bcpm <- basecontent(hgu95av2probe$sequence)
> as.matrix(bcpm[1:5, ])
```

```
 A T C  G
 2 9 9  5
 6 3 9  7
 5 4 9  7
 5 6 8  6
 5 7 3 10
```

```
> bcmm <- basecontent(complementSeq(hgu95av2probe$sequence, start = 13,
+     stop = 13))
> as.matrix(bcmm[1:5, ])
```

```
 A  T C G
 1 10 9 5
 7  2 9 7
 4  5 9 7
 5  6 7 7
 5  7 4 9
```

3   Combine

Load the data

```
x1 <- read.affybatch("118T1.cel")
x2 <- read.affybatch("CL2001032020AA.CEL")
```

Combine the data. For this to work it is required that the packages *hu6800probe*
and *hgu95av2probe* are installed.

```
res = combine(list(x1, x2), c("hu6800probe","hgu95av2probe"))
```

The function returns a list `res` with two elements: an `AffyBatch data` and a
`CDF enviroment newcdf`.

4

```
newcdf <- res$newcdf
z <- rma(res$data)
```

View the results.

```
x11(width=7, height=7)
par(mfrow=c(2,2))

plot(exprs(res$data), main="after combine", pch=".", log="xy")
plot(exprs(z),        main="after RMA", pch=".")

## the number of probes in each "new" probe set
prs = multiget(ls(newcdf), newcdf)
nrprobes = sapply(prs, function(x) nrow(x))
barplot(table(nrprobes), xlab="no probes in probeset", ylab="frequency")
```

4   Creating probe packages

Probe packages are a convenient way for distributing and storing the probe
sequences and related information.

4.1   Affymetrix genechips

In this section we see how a probe package can be created for Affymetrix
genechips from the tabulator-separated data files that can be obtained from the
vendor (at http://www.netaffx.com). As an example, the file HG-U95Av2_probe_tab.gz
is provided in the data subdirectory of the *matchprobes* package.

```
> filename <- file.path(.path.package("matchprobes"), "data", "HG-U95Av2_probe_tab.gz")
> outdir <- tempdir()
> me <- "Wolfgang Huber <w.huber@dkfz.de>"
> makeProbePackage("HG-U95Av2", datafile = gzfile(filename, open = "r"),
+     outdir = outdir, maintainer = me, version = "0.0.1", force = TRUE)


Importing the data.
Creating package in /tmp/Rtmp5144/hgu95av2probe
Writing the data.
Checking the package.
Building the package.
[1] "hgu95av2probe"
```

```
> dir(outdir)
```

```
[1] "hgu95av2probe"          "hgu95av2probe_0.0.1.tar.gz"
```

4.2   Other chiptypes

This requires writing your own `importfun`.

Input of an importfun:

- directory name of a package template directory, containing at least a file
  `DESCRIPTION`, a directory `man` with a file @PKGNAME@.Rd, a directory
  `data`, and possible other directories and files, conforming to the usual R
  package conventions.

Output of an importfun:

- an environment with a dataframe named @PKGNAME@ and a column
  'sequence'. The dataframe may have other columns, and the environment
  may contain other data objects. These will be saved as individual .rda
  files of the same name into the data directory of the produced package.
  Documentation needs to be provided for the columns, and for the other
  objects.

- Symbols:

  - ARRAYTYPE: name of the array
  - DATASOURCE: a textual description of how the data were obtained.
    It should contain the URL, or the name of the company / person.
  - FORMAT

For more details, please refer to the help files for the functions `makeProbePack-`
`age` and `getProbeDataAffy`. For an example, refer to the source code of `get-`
`ProbeDataAffy`.