The mammaPrintData package: annotated gene expression data from the Glas and Buyse breast cancer cohorts

Luigi Marchionni[1]

[1]The Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University School of Medicine

May 7, 2020

Contents

1 Overview

This vignette documents the R code used to retrieve from the original sources, annotate, and then assemble into separate RGList instances the gene expression data of the Glas and Buyse cohorts [3, 2]. Such data was used to translate the 70-gene breast cancer prognostic signature [7, 6] into the MammaPrint assays [3], and to independently validate it [2]. While the original studies by van't Veer and Van de Vijver and colleagues used the two-colors Agilent/Rosetta 24k array, the redeveloped assays is based on the 1.9k MammaPrint array. For a compete review of these breast cancer studies see Marchionni et al [4, 5].

Briefly, this document contains the complete R code used to:

1. Download the following gene expression data from ArrayExpress [1]:
   (a) The Glas data set (ArrayExpress E-TABM-115 series), which combined a large proportion of cases from the original van't Veer and Van de Vijver cohorts, and was used as training set;
   (b) The Buyse data set (ArrayExpress E-TABM-77), which analyzed a European multicenter cohort, and was used as validation set;
2. Pre-process and normalize all gene expression data;
3. Cross-reference the 70-gene list to the new MammaPrint microarray;
4. Create the corresponding limma package RGList instances.

All the files, among those mentioned above, needed to compile this vignette and create the mammaPrintData Rpackage are included inside the inst/extdata directory of the source package.

Finally, the R code chunk below was used todownload and install from Bioconductorthe packages required to prepared the present vignette. if the are not available (e.g gdata).

```
> ###Get the list of available packages
> installedPckgs <- installed.packages()[,"Package"]
> ###Define the list of desired libraries
> pckgListBIOC <- c("Biobase", "limma", "gdata")
> ###Install BiocManager if needed
> if (!requireNamespace("BiocManager", quietly=TRUE))
     install.packages("BiocManager")
> ###Load the packages, install them from Bioconductor if needed
> for (pckg in pckgListBIOC) {
         if (! pckg %in% installedPckgs) BiocManager::install(pckg)
         require(pckg, character.only=TRUE)
 }
```

2 The 70-gene signature

The list of genes originally identified in the van't Veer study [7] can be obtained from the supplementary data section of the original publication, at the following URL:

- http://www.nature.com/nature/journal/v415/n6871/extref/415530a-s9.xls

The Excel spreadsheet named `415530a-s9.xls` corresponds to Table S2, and contains the identifiers and annotation for the 231 genes othat proved to be associated with metastatic recurrence at 5 years. According to the information contained in the legend for Table S2 (see"Supplementary Methods" section) the optimal set of genes constituting the 70-gene signature can be identified from this table by selecting the top 70 features/reporters with highest absolute correlation coefficients.

Below is shown the code chunk used to download the `415530a-s9.xls` Excel file from the Nature website. This file was part of the Supplementary Information section associated with the original van't Veer's et manuscript. The following code chunk was not run to compile this vignette, which was prepared using the local copies of the data stored in the `inst/extdata` directory.

```
> ###Chunk not executed: files are already included
> ### Shown here just to show the source of the data
> dir.create("../extdata/seventyGenes", showWarnings = TRUE, recursive = TRUE)
> ###Define the url for Supplementary data on the Nature Website
> url <- "http://www.nature.com/nature/journal/v415/n6871/extref/415530a-s9.xls"
> ###Dowload the Excel file from Nature
> download.file(url, destfile="../extdata/seventyGenes/415530a-s9.xls",
               quiet = FALSE, mode = "w", cacheOK = TRUE)
```

Below is the `R` code chunk used to directly process the `415530a-s9.xls` Excel file and create the corresponding data.frame containing the features annotation.

```
> ###Load the library to process Excell files
> require(gdata)
> ###Read the Excel file with the 70-genes signature information for van't Veer dataset
> myFile <- system.file("./extdata/seventyGene", "415530a-s9.xls", package = "mammaPrintData")
> gns231 <- read.xls(myFile, skip=0, header=TRUE, stringsAsFactors=FALSE)
> ###Remove special characters in the colums header
> ###These are due to white spaces present in the Excel file
> colnames(gns231) <- gsub("\\.\\.", "", colnames(gns231))
> ###Remove GO annotation
> gns231 <- gns231[, -grep("sp_xref_keyword_list", colnames(gns231))]
> ###Check the structure of the data.frame
> str(gns231)
```

```
'data.frame':        231 obs. of  4 variables:
 $ accession  : chr  "NM_003748" "NM_003862" "Contig32125_RC" "U82987" ...
 $ correlation: num  0.421 0.411 0.409 0.407 0.402 ...
 $ gene.name  : chr  "ALDH4" "FGF18" "" "BBC3" ...
 $ description: chr  "aldehyde dehydrogenase 4 (glutamate gamma-semialdehyde dehydrogenase; pyrr
```

Below is the `R` code chunk used to generate a list of identifiers corresponding to the 70-gene and the 231-gene lists.

```
> ###Reorder the genes in decreasing order by absolute correlation
> gns231 <- gns231[order(abs(gns231$correlation), decreasing=TRUE),]
> ###Select the feature identifiers corresponding to the top 70 genes
> gns70 <- gns231[1:70,]
> ###Create a list of features and gene symbols for later use
> progSig <- list(gns231acc = unique(gns231$accession), gns231name = unique(gns231$gene.name),
```

```
                gns231any = unique(c(gns231$accession, gns231$gene.name)),
                gns70acc = unique(gns70$accession), gns70name = unique(gns70$gene.name),
                gns70any = unique(c(gns70$accession, gns70$gene.name)) )
> ###Remove empty elements
> progSig <- lapply(progSig, function(x) x[x!=""])
> ###Create a data.frame of correlations with combined accession
> ###and gene symbols as identifiers
> gns231Cors <- data.frame(stringsAsFactors=FALSE,
                           ID=c(gns231$gene.name[ gns231$gene.name %in% progSig$gns231any ],
                           gns231$accession),
                           gns231Cors=c(gns231$correlation[ gns231$gene.name %in% progSig$gns231a
                           gns231$correlation) )
> ####Keep unique only
> progSig$gns231Cors <- gns231Cors[!duplicated(gns231Cors$ID), ]
> progSig$gns231Cors <- gns231Cors[gns231Cors$ID != "", ]
> ###Check the structure of the list just created
> str(progSig)

List of 7
 $ gns231acc : chr [1:231] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231name: chr [1:161] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
 $ gns231any : chr [1:392] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70acc  : chr [1:70] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70name : chr [1:50] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
 $ gns70any  : chr [1:120] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231Cors:'data.frame':        394 obs. of  2 variables:
  ..$ ID        : chr [1:394] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
  ..$ gns231Cors: num [1:394] -0.425 0.421 0.411 0.407 0.402 ...
```

3 Glas and Buyse cohorts: `RGList` preparation

The original data sets used to redevelop [3] and validate [2] the MammaPrint assays can be obtained from the ArrayEpress database [1]:

- `http://www.ebi.ac.uk/arrayexpress/experiments/E-TABM-77`
- `http://www.ebi.ac.uk/arrayexpress/experiments/E-TABM-115`

These gene expression series account for clinical information, microarray feature annotation, and complete unprocessed raw gene expression data. In both studies a two-color design with dye-swap replication was applied, therefore two distinct `RGList` objects were prepared for each data set, and included in the `mammaPrintData` `R-Bioconductor` package, as detailed below. The pre-processed, normalized, and summarized values used by the authors in their original analyses were not included in the submission to ArrayExpress, hence data pre-processing and normalization of the `RGList` contained in this package is required prior any analysis.

3.1 Glas cohort - ArrayExpress E-TABM-115 Series

The Glas data set combines a large proportion of cases from the original van't Veer and Van de Vijver cohorts, and was used for the implementation of the original 70-gene signature into the MammaPrint assay [3].

The following chunk of `R` code was used to download the all the files of the ArrayExpress the E-TABM-115 series. The files needed to compile this vignette and create the `mammaPrintData` Rpackage are included inside the `inst/extdata` directory.

```
> ###Chunk nor executed: files are already included in the package
> ###This chunk is just to show the source of the data
> ###Load the ArrayExpress library
> require(ArrayExpress)
> ###Create a working directory
> dir.create("../extdata/ETABM115", showWarnings = FALSE)
> ###Obtain the data package E-TABM-115 from ArrayExpress
> etabm115 <- getAE("E-TABM-115", type = "full", path="../extdata/ETABM115", extract=FALSE)
> ###Save
> save(etabm115, file="../extdata/ETABM115/etabm115.rda")
```

Below is the `R` code chunk used to process the phenotypic information retrieved form ArrayExpress and associated with the E-TABM-115 series on May 07, 2020

```
> ###Load the Biobase library
> require(Biobase)
> ###List the files obtained from ArrayExpress
> etabm155filesLoc <- system.file("extdata/ETABM115", package = "mammaPrintData")
> head(dir(etabm155filesLoc))
```

```
[1] "A-MEXP-318.adf.txt.gz"
[2] "E-TABM-115.sdrf.txt.gz"
[3] "US22502555_16011886011186_S01_1_1.txt.gz"
[4] "US22502555_16011886011186_S01_1_2.txt.gz"
[5] "US22502555_16011886011186_S01_1_3.txt.gz"
[6] "US22502555_16011886011186_S01_1_4.txt.gz"
```

```
> ###Read the phenotypic information from "E-TABM-115.sdrf.txt"
> targets <- read.table(dir(etabm155filesLoc, full.names=TRUE, pattern="E-TABM-115.sdrf.txt"),
                        comment.char="", sep="\t", header=TRUE, stringsAsFactors=FALSE)
> ###Check and process the phenotypic information: keep non-redundant information
> targets <- targets[, apply(targets, 2, function(x) length(unique(x)) != 1 )]
> ###Reorder by file name
> targets <- targets[order(targets$Array.Data.File, decreasing=TRUE),]
> ###Remove points in colnames
> colnames(targets) <- gsub("\\.$","",gsub("\\.\\.",".",colnames(targets)))
> ###Show the available phenotypic iformation
> str(targets)
```

```
'data.frame':        324 obs. of  16 variables:
 $ Source.Name                                                  : chr  "MRP" "9671397" "MRP" "9
```

```
$ Characteristics.EventDistantMetastases                          : int  NA 1 NA 0 NA 0 NA NA 0 0
$ Characteristics.EventDeath                                      : int  NA 1 NA 0 NA 0 NA NA 0 0
$ Characteristics.PeriodTillDevelopmentOfDistantMetastases.years: num  NA 4.77 NA NaN NA NaN NA
$ Characteristics.OverallSurvival                                 : chr  "  " "8.77 years" "  " "
$ Sample.Name                                                     : chr  "MRP" "9671397" "MRP" "9
$ Extract.Name                                                    : chr  "MRP_e1" "9671397_e1" "M
$ Labeled.Extract.Name                                            : chr  "MRP_e1_l1" "9671397_e1_
$ Label                                                           : chr  "Cy3" "Cy5" "Cy3" "Cy5"
$ Hybridization.Name                                              : chr  "agendia.com; ebi.co.uk:
$ Factor.Value.overall_survival                                   : chr  "reference pool" "8.77 y
$ Factor.Value.Event_Death                                        : chr  "reference pool" "1" "re
$ Factor.Value.Event_distant_metastases                           : chr  "reference pool" "1" "re
$ Factor.Value.Time_to_development_of_distant_metastases          : chr  "reference pool" "4.77 y
$ Scan.Name                                                       : chr  "US22502555_967_2_397" "
$ Array.Data.File                                                 : chr  "US22502555_967_2_397.tx
```

The Glas cohort accounts for 162 unique cases, which were analyzed using a dye-swap two-colors design, comparing each sample against the same reference RNA. According to ArrayEspress instruction for SDRF files (available at `http://tab2mage.sourceforge.net/docs/magetab_docs.html`), a complete SDRF annotation file consists of a table in which each hybridization channel is represented by an individual row. Therefore the complete SDRF annotation files for all hybridizations of the Glas cohort should for 648 total rows: 162 samples for 2 channels (cy5 and Cy3) for 2 replicated hybridizations (dye-swap). As shown below, the `E-TABM-115.sdrf.txt` file currently available from ArrayExpress (May 07, 2020) is incomplete, since it contains only 324 rows. An inspection of the information contained in the downloaded SDRF file reveales that it contains a row for each hybridization (n=324), rather than a row for each channel (n=648).

```
> ###Check targets data.frame dimentions
> dim(targets)

[1] 324  16

> ###Count the unique file names
> length(unique(targets$Array.Data.File))

[1] 324

> ###Count the number of rows associated with each channel
> table(CHANNEL=targets$Label)

CHANNEL
Cy3 Cy5
162 162

> ###Count the number of rows for the reference RNA (named "MRP") in each channel
> table(CHANNEL=targets$Label, REFERENCE=targets$Sample.Name == "MRP")

       REFERENCE
CHANNEL FALSE TRUE
    Cy3     0  162
    Cy5   162    0
```

```
> ###Count the number of distinct RNA analyzed (excluding the refernce RNA)
> sum(REFERENCE=targets$Sample.Name!="MRP")

[1] 162

> ###Count the number of metastatic events
> table(METASTASES=targets$Characteristics.EventDistantMetastases)

METASTASES
  0   1
106  56

> ###Count the number of rown for which metastatic events is missing
> table(MISSING=is.na(targets$Characteristics.EventDistantMetastases))

MISSING
FALSE  TRUE
  162   162

> ###Check if missing clinical information is exactly associated with
> ###hybridizations where the "MRP" reference RNA was used in the Cy5 channel
> table(MISSING=is.na(targets$Characteristics.EventDistantMetastases),
      REFERENCE=targets$Sample.Name!="MRP")

      REFERENCE
MISSING FALSE TRUE
  FALSE     0  162
  TRUE    162    0
```

Therefore the following conclusions can be drawn from the evaluation of the information contained in `E-TABM-115.sdrf.txt` file shown above:

- All hybridizations that were performed have a corresponding row in the SDRF table;
- Half of the rows of the SDRF table, for which the RNA samples described were associated with the Cy5 channel, contain a complete set of clinical information;
- The other half of the rows, for which the RNA samples described were associated with the Cy3 channel, report the information for the reference RNA, and the clinical information is missing;
- Nevertheless, given the dye swap design of the experiment, the SDRF table contains the complete clinical information associated with all the 162 breast cancer patients analyzed in the Glas study;
- It is possible to use the clinical information provided only with one set of dye-swap hybridizations, since for the other set ony the "MRP" reference RNA information was provided.

In conclusion, the complete data set can be analyzed only knowing which hybridizations constitute a dye-swap pair. This information can be obtained upon request from Dr. Glas, the corrisponding author of the of the original study.

Below is the `R` code chunk used to create the `RGList` objects using the raw gene expression data contained in the compressed archive `E-TABM-115.raw.1.zip` downloaded from ArrayExpress.

```
> ###List files for E-TABM-115 experiment
> etabm155filesLoc <- system.file("extdata/ETABM115", package = "mammaPrintData")
> etabm155files <- list.files(etabm155filesLoc,  pattern="^US")
```

```
> ###Check whether all available files correspond to the targets$Array.Data.File
> all(paste(targets$Array.Data.File, ".gz", sep="") %in% etabm155files)

[1] TRUE

> ###Check whether they are ordered in the same way
> all(paste(targets$Array.Data.File, ".gz", sep="") == etabm155files)

[1] FALSE

> ###Load the required library
> require(limma)
> ###Define the columns which will be read from the raw files selection
> colsList <- list(Rf="Feature Extraction Software:rMedianSignal",
                   Rb="Feature Extraction Software:rBGMedianSignal",
                   Gf="Feature Extraction Software:gMedianSignal",
                   Gb="Feature Extraction Software:gBGMedianSignal",
                   logRatio="Feature Extraction Software:LogRatio",
                   logRatioError="Feature Extraction Software:LogRatioError")
> ###Subset the targets data.frame for the hybridization in which the Reference RNA was labeled
> targetsCy3info <- targets[ targets$Source.Name == "MRP" & targets$Label == "Cy3", ]
> dim(targetsCy3info)

[1] 162  16

> ###Subset the targets data.frame for the hybridization in which the Reference RNA was labeled
> targetsCy5info <- targets[ targets$Source.Name != "MRP" & targets$Label == "Cy5", ]
> dim(targetsCy5info)

[1] 162  16

> ###The two sets of hybridizations above should be mutually exclusive
> all(targetsCy3info$Array.Data.File != targetsCy5info$Array.Data.File)

[1] TRUE

> ###Create an "RGList" using specific columns for the first set of swapped hybridizations:
> RGcy3 <- read.maimages(paste(targetsCy3info$Array.Data.File,  ".gz", sep=""),
                         source="generic",
                         columns=colsList, annotation="Reporter identifier",
                         path=etabm155filesLoc, verbose=FALSE)
> ###Add targets information
> RGcy3$targets <- targetsCy3info
> ###Format Reporter identifiers
> RGcy3$genes$ID <- gsub(".+A-MEXP-318\\.", "", RGcy3$genes[, "Reporter identifier"])
> ###Check dimensions
> dim(RGcy3)

[1] 1900  162

> ###Create an "RGList" using specific columns for the second set of swapped hybridizations:
> RGcy5 <- read.maimages(paste(targetsCy3info$Array.Data.File,  ".gz", sep=""),
                         source="generic",
                         columns=colsList, annotation="Reporter identifier",
```

```
                        path=etabm155filesLoc, verbose=FALSE)
> ###Add targets information
> RGcy5$targets <- targetsCy5info
> ###Format Reporter identifiers
> RGcy5$genes$ID <- gsub(".+A-MEXP-318\\.", "", RGcy5$genes[, "Reporter identifier"])
> ###Check dimensions
> dim(RGcy5)

[1] 1900  162
```

Below is the `R` code chunk used to check whether all data read is of the correct mode (i.e. numeric values are of mode "numeric", and so on).

```
> ###Check mode of read data for first set of hybridizations
> sapply(RGcy3, mode)

            R            Rb             G            Gb      logRatio
    "numeric"     "numeric"     "numeric"     "numeric"   "character"
logRatioError       targets         genes        source
    "numeric"        "list"        "list"   "character"

> ###Since RGc3$logRatio is character convert to numeric
> RGcy3$logRatio <- apply(RGcy3$logRatio, 2, as.numeric)
> ###Check mode of read data for second set of hybridizations
> sapply(RGcy5, mode)

            R            Rb             G            Gb      logRatio
    "numeric"     "numeric"     "numeric"     "numeric"   "character"
logRatioError       targets         genes        source
    "numeric"        "list"        "list"   "character"

> ###Since RGc3$logRatio is character convert to numeric
> RGcy5$logRatio <- apply(RGcy5$logRatio, 2, as.numeric)
```

Below is the `R` code chunk used to process the microarray feature annotation for the 1.9k MammaPrint array, as it is provided in the ArrayExpress database (file `A-MEXP-318.adf.txt`. We will then map the microarray features from thes new array to the 231-gene and the 70-gene prognostic signatures. We will finally add annotation information to the `RGList` objects created above.

```
> ###Read genes information for MammaPrint array from the A-MEXP-318.adf.txt
> genes <- read.table(dir(etabm155filesLoc, full.names=TRUE, pattern="A-MEXP-318.adf.txt"),
                      skip=21, sep="\t", header=TRUE, stringsAsFactors=FALSE)
> ###Remove special characters like points in the colums header
> colnames(genes) <- gsub("\\.$","",gsub("\\.\\.",".",colnames(genes)))
> ###Use the annotation information previously processed
> str(progSig)

List of 7
 $ gns231acc : chr [1:231] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231name: chr [1:161] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
 $ gns231any : chr [1:392] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70acc  : chr [1:70] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70name : chr [1:50] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
```

```
 $ gns70any  : chr [1:120] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231Cors:'data.frame':        394 obs. of  2 variables:
  ..$ ID        : chr [1:394] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
  ..$ gns231Cors: num [1:394] -0.425 0.421 0.411 0.407 0.402 ...
> ###Count the features mapped to the 231-gene prognostic signature by accession number
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231acc)
                  Block.Column                      Block.Row
                             0                              0
                        Column                            Row
                             0                              0
                  Reporter.Name          Comment.AEReporterName
                             0                             68
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             7                              0
            Reporter.Group.role                   Control.Type
                             0                              0
> ###Count the features mapped to the 231-gene prognostic signature by gene symbol
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231name)
                  Block.Column                      Block.Row
                             0                              0
                        Column                            Row
                             0                              0
                  Reporter.Name          Comment.AEReporterName
                             0                            161
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             0                              0
            Reporter.Group.role                   Control.Type
                             0                              0
> ###Count the features mapped to the 231-gene prognostic signature by gene symbol or accession
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231any)
                  Block.Column                      Block.Row
                             0                              0
                        Column                            Row
                             0                              0
                  Reporter.Name          Comment.AEReporterName
                             0                            229
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             7                              0
            Reporter.Group.role                   Control.Type
                             0                              0
> ###Add mapping information for 229 features using both gene symbols and accession numbers
> genes$genes231 <- genes$Comment.AEReporterName %in% progSig$gns231any
> ###Count the features mapped to the 70-gene prognostic signature by accession number
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70acc)
```

```
                  Block.Column                       Block.Row
                             0                               0
                        Column                             Row
                             0                               0
                  Reporter.Name          Comment.AEReporterName
                             0                              19
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             1                               0
            Reporter.Group.role                    Control.Type
                             0                               0
                       genes231
                             0
```

```
> ###Count the features mapped to the 70-gene prognostic signature by gene symbol
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70name)
```

```
                  Block.Column                       Block.Row
                             0                               0
                        Column                             Row
                             0                               0
                  Reporter.Name          Comment.AEReporterName
                             0                              50
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             0                               0
            Reporter.Group.role                    Control.Type
                             0                               0
                       genes231
                             0
```

```
> ###Count the features mapped to the 70-gene prognostic signature by gene symbol or accession n
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70any)
```

```
                  Block.Column                       Block.Row
                             0                               0
                        Column                             Row
                             0                               0
                  Reporter.Name          Comment.AEReporterName
                             0                              69
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             1                               0
            Reporter.Group.role                    Control.Type
                             0                               0
                       genes231
                             0
```

```
> ###Add mapping information for 69 features using both gene symbols and accession numbers
> genes$genes70 <- genes$Comment.AEReporterName %in% progSig$gns70any
> ###The resulting gene annotation file
> str(genes)
```

```
'data.frame':        1900 obs. of  12 variables:
```

```
 $ Block.Column                   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Block.Row                      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Column                         : int  1 2 13 20 40 42 48 49 50 1 ...
 $ Row                            : int  1 1 1 1 1 1 1 1 1 2 ...
 $ Reporter.Name                  : chr  "AGEN000001" "AGEN000002" "AGEN000013" "AGEN000020" ...
 $ Comment.AEReporterName         : chr  "Pro25G" "3xSLv1" "eQC" "eQC" ...
 $ Reporter.Database.Entry.embl   : chr  "" "" "" "" ...
 $ Reporter.Database.Entry.unigene: chr  "" "" "" "" ...
 $ Reporter.Group.role            : chr  "Control" "Control" "Control" "Control" ...
 $ Control.Type                   : chr  "control_biosequence" "control_biosequence" "control_bi
 $ genes231                       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ genes70                        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
> ###Add original correlation from the van't Veer study
> gns231Cors <- progSig$gns231Cors
> gns231Cors <- gns231Cors[gns231Cors$ID %in% genes$Comment.AEReporterName,]
> ###Merge and remove duplicates
> genes <- merge(genes, gns231Cors, all.x=TRUE, all.y=FALSE,
                 by.x="Comment.AEReporterName", by.y="ID")
> genes <- genes[!duplicated(genes$Reporter.Name),]
> ###Check genes
> str(genes)
```

```
'data.frame':        1900 obs. of  13 variables:
 $ Comment.AEReporterName         : chr  "3xSLv1" "3xSLv1" "3xSLv1" "3xSLv1" ...
 $ Block.Column                   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Block.Row                      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Column                         : int  2 48 1 2 48 49 50 1 4 7 ...
 $ Row                            : int  1 1 2 2 2 2 2 9 9 9 ...
 $ Reporter.Name                  : chr  "AGEN000002" "AGEN000048" "AGEN000051" "AGEN000052" ...
 $ Reporter.Database.Entry.embl   : chr  "" "" "" "" ...
 $ Reporter.Database.Entry.unigene: chr  "" "" "" "" ...
 $ Reporter.Group.role            : chr  "Control" "Control" "Control" "Control" ...
 $ Control.Type                   : chr  "control_biosequence" "control_biosequence" "control_bi
 $ genes231                       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ genes70                        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ gns231Cors                     : num  NA NA NA NA NA NA NA NA NA NA ...
```

Below is the `R` code chunk used to add the feature annotation the `RGList` objects containing the expression data (first set of hybridizations).

```
> ###Compare genes between platform and RGList instances: first set of hybridizations
> if  (nrow(genes)==nrow(RGcy3)) {
        ##Compare the identifiers' order
        if ( !all(genes$Reporter.Name == RGcy3$genes$ID) ) {
               ##Reorder if needed
               RGcy3 <- RGcy3[order(RGcy3$genes$ID),]
               genes <- genes[order(genes$Reporter.Name),]
               ##Check for order AGAIN
               if ( all(genes$Reporter.Name == RGcy3$genes$ID) ) {
```

```
                ##Substitute genes or stop
                RGcy3$genes <- genes
        } else {
                stop("Wrong gene order, check objects")
        }
    } else {
        print("Updating gene annotation")
        RGcy3$genes <- genes
    }
} else {
    stop("Wrong number of features, check objects")
}
> ###Rename the object
> glasRGcy3 <- RGcy3
```

Below is the `R` code chunk used to add the feature annotation the `RGList` objects containing the expression data (second set of hybridizations).

```
> ###Compare genes between paltform and RGList instances: second set of hybridizations
> if (nrow(genes)==nrow(RGcy5)) {
    ##Compare the identifiers' order
    if ( !all(genes$Reporter.Name == RGcy5$genes$ID) ) {
        ##Reorder if needed
        RGcy5 <- RGcy5[order(RGcy5$genes$ID),]
        genes <- genes[order(genes$Reporter.Name),]
        ##Check for order AGAIN
        if ( all(genes$Reporter.Name == RGcy5$genes$ID) ) {
            ##Substitute genes or stop
            RGcy5$genes <- genes
        } else {
            stop("Wrong gene order, check objects")
        }
    } else {
        print("Updating gene annotation")
        RGcy5$genes <- genes
    }
} else {
    stop("Wrong number of features, check objects")
}
> ###Rename the object
> glasRGcy5 <- RGcy5
```

3.2 Buyse cohort - ArrayExpress E-TABM-77 Series

The Buyse data set (ArrayExpress E-TABM-77) accounts for the large European multicenter breast cancer cohort that was used to validate the MammaPrint assay [2].

The following chunk of `R` code was used to download the all the files of the ArrayExpress the E-TABM-77 series. The files needed to compile this vignette and create the `mammaPrintData` Rpackage are included

inside the `inst/extdata` directory.

```
> ###Chunk not executed: files are already included in the package
> ###This chunk is just to show the source of the data
> dir.create("../extdata/ETABM77", showWarnings = FALSE)
> ##################################################
> ###Obtain the data package E-TABM-115 from ArrayExpress
> etabm77 <- getAE("E-TABM-77", type = "full", path="../extdata/ETABM77", extract=FALSE)
> ##################################################
> ###Save
> save(etabm77,file="../extdata/ETABM77/etabm77.rda")
```

Below is the `R` code chunk used to process the phenotypic information retrieved form ArrayExpress and associated with the E-TABM-77 series on May 07, 2020.

```
> ###List the files obtained from ArrayExpress
> etabm77filesLoc <- system.file("extdata/ETABM77", package = "mammaPrintData")
> head(dir(etabm77filesLoc))

[1] "A-MEXP-318.adf.txt.gz"
[2] "E-TABM-77.sdrf.txt.gz"
[3] "US22502555_251188611675_S01_1_1_04000344_1_MRP.txt.gz"
[4] "US22502555_251188611675_S01_1_2_MRP_04000344_1.txt.gz"
[5] "US22502555_251188611675_S01_2_1_04000346_1_MRP.txt.gz"
[6] "US22502555_251188611675_S01_2_2_MRP_04000346_1.txt.gz"

> ###Read the phenotypic information from "E-TABM-77.sdrf.txt"
> targets <- read.table(dir(etabm77filesLoc, full.names=TRUE, pattern="E-TABM-77.sdrf.txt"),
                        comment.char="", sep="\t", header=TRUE, stringsAsFactors=FALSE)
> ###Check and process the phenotypic information: keep non-redundant information
> targets <- targets[, apply(targets, 2, function(x) length(unique(x)) != 1 )]
> ###Reorder by file name
> targets <- targets[order(targets$Array.Data.File, decreasing=TRUE),]
> ###Remove points in colnames
> colnames(targets) <- gsub("\\.$","",gsub("\\.\\.",".",colnames(targets)))
> ###Show the available phenotypic iformation
> str(targets)

'data.frame':        1228 obs. of  30 variables:
 $ Source.Name                                    : chr  "MRP" "513" "513" "MRP" ...
 $ Characteristics.Age                            : chr  "  " "47 years at diagnosis" "47 years
 $ Characteristics.RiskStatus                     : chr  "  " "MammaPrint result:high risk" "Mam
 $ Characteristics.TumorSize                      : chr  "  " "38 mm diameter" "38 mm diameter"
 $ Characteristics.TumorGrading                   : chr  "  " "2" "2" "  " ...
 $ Characteristics.BioSourceProvider              : chr  "Agendia" "JRH" "JRH" "Agendia" ...
 $ Characteristics.SurvivalProbability            : chr  "  " "64.5%" "64.5%" "  " ...
 $ Characteristics.Disease.Free.Survival          : chr  "  " "3.1 years" "3.1 years" "  " ...
 $ Characteristics.TestResult                     : chr  "  " "Estrogen Receptor Negative" "Estr
 $ Characteristics.DistantMetastasis.Free.Survival: chr  "  " "5.3 years plus" "5.3 years plus"
 $ Characteristics.SurvivalTime                   : chr  "  " "5.3 years" "5.3 years" "  " ...
 $ Characteristics.RiskStatus.1                   : chr  "  " "Adjuvant online:high clinical ris
```

```
$ Characteristics.RiskStatus.2                  : chr  "  " "NPI:high clinical risk" "NPI:high
$ Characteristics.RiskStatus.3                  : chr  "  " "St Gallen category:high clinical
$ Protocol.REF                                  : chr  "P-TABM-213" "  " "  " "P-TABM-213" ...
$ Sample.Name                                   : chr  "MRP" "4000702" "4000702" "MRP" ...
$ Extract.Name                                  : chr  "MRP_e1" "4000702_e1" "4000702_e1" "MRP
$ Labeled.Extract.Name                          : chr  "MRP_e1_l1Cy5" "4000702_e1_l1Cy3" "4000
$ Label                                         : chr  "Cy5" "Cy3" "Cy5" "Cy3" ...
$ Hybridization.Name                            : chr  "TABM:Tab2MAGE:E-TABM-77.2517.Hybridiza
$ Factor.Value.NPI.Risk.Status                  : chr  "NPI:high clinical risk" "NPI:high clin
$ Factor.Value.Adjuvant.online.risk.status      : chr  "Adjuvant online:high clinical risk" "A
$ Factor.Value.MammaPrint.prediction            : chr  "MammaPrint result:high risk" "MammaPri
$ Factor.Value.Disease.Free.Survival            : chr  "3.1 years" "3.1 years" "3.1 years" "3.
$ Factor.Value.DistantMetastasis.Free.Survival  : chr  "5.3 years plus" "5.3 years plus" "5.3
$ Factor.Value.St.Gallen.Risk.Status            : chr  "St Gallen category:high clinical risk'
$ Factor.Value.SurvivalTime                     : chr  "5.3 years" "5.3 years" "5.3 years" "5.
$ Factor.Value.ER.status                        : chr  "Estrogen Receptor Negative" "Estrogen
$ Scan.Name                                     : chr  "US22502555_251188613295_S01_2_4_MRP1_8
$ Array.Data.File                               : chr  "US22502555_251188613295_S01_2_4_MRP1_8
```

The Buyse cohort accounts for 307 unique cases, which were analyzed using a dye-swap two-colors design, comparing each sample against the same reference RNA. According to ArrayEspress instruction for SDRF files (available at `http://tab2mage.sourceforge.net/docs/magetab_docs.html`), a complete SDRF annotation file consists of a table in which each hybridization channel is represented by an individual row. Therefore the complete SDRF annotation files for all hybridizations of the Buse cohort should for 1228 total rows: 307 samples for 2 channels (cy5 and Cy3) for 2 replicated hybridizations (dye-swap). As shown below, the `E-TABM-77.sdrf.txt` file currently available from ArrayExpress (May 07, 2020) is complete and contains all nececessary information.

An inspection of the information contained in the downloaded SDRF file reveales that it contains two row for each hybridization (n=1228) as expected.

```
> ###Check targets data.frame dimentions
> dim(targets)

[1] 1228   30

> ###Count the unique file names
> length(unique(targets$Array.Data.File))

[1] 614

> ###Count the number of rows associated with each channel
> table(CHANNEL=targets$Label)

CHANNEL
Cy3 Cy5
614 614

> ###Count the number of rows for the reference RNA (named "MRP") in each channel
> table(CHANNEL=targets$Label, REFERENCE=targets$Sample.Name == "MRP")

       REFERENCE
CHANNEL FALSE TRUE
```

```
    Cy3   307  307
    Cy5   307  307
> ###Count the number of distinct RNA analyzed (excluding the refernce RNA)
> sum(REFERENCE=targets$Sample.Name!="MRP")

[1] 614

> ###Count the MammaPrint predictions
> table(MAMMAPRINT=targets$Factor.Value.MammaPrint.prediction)

MAMMAPRINT
MammaPrint result:high risk  MammaPrint result:low risk
                        776                         452
```

The `E-TABM-77.sdrf.txt` file is therefore valid and can be used to assemble the two `RGList` objects corresponding to the two sets of dye-swap hybridization that were performed, as shown in the `R` code below.

```
> ###List files for E-TABM-77 experiment
> etabm77filesLoc <- system.file("extdata/ETABM77", package = "mammaPrintData")
> etabm77files <- list.files(etabm77filesLoc,  pattern="^US")
> ###Check whether all available files correspond to the targets$Array.Data.File
> all(paste(targets$Array.Data.File, ".gz", sep="") %in% etabm77files)

[1] TRUE

> ###Check whether they are ordered in the same way
> all(paste(targets$Array.Data.File, ".gz", sep="") == etabm77files)

[1] FALSE

> ####Subset targets table extracting the rows for CANCER and the separate by channel
> caList <- !targets$Sample.Name=="MRP"
> targetsCa <- targets[caList,]
> ###Cy5 channel
> targetsCa.ch1 <- targetsCa[targetsCa$Label=="Cy5",]
> colnames(targetsCa.ch1) <- paste(colnames(targetsCa.ch1),"Cy5",sep=".")
> ###Cy3 channel
> targetsCa.ch2 <- targetsCa[targetsCa$Label=="Cy3",]
> colnames(targetsCa.ch2) <- paste(colnames(targetsCa.ch2),"Cy3", sep=".")
> ####Subset targets table extracting the rows for REFERENCE and the separate by channel
> refList <- targets$Sample.Name=="MRP"
> targetsRef <- targets[refList,]
> ###Cy5 channel
> targetsRef.ch1 <- targetsRef[targetsRef$Label=="Cy5",]
> colnames(targetsRef.ch1) <- paste(colnames(targetsRef.ch1),"Cy5",sep=".")
> ###Cy3 channel
> targetsRef.ch2 <- targetsRef[targetsRef$Label=="Cy3",]
> colnames(targetsRef.ch2) <- paste(colnames(targetsRef.ch2),"Cy3", sep=".")
```

Below is the `R` code chunk used to assemble the phenotypic information associated with the first set of hybridizations.

```
> ###Now combine channel information to create the new targets object
> ###Here is for the FIRST set of swapped hybridization (keep only useful columns)
> if ( all(targetsCa.ch1$Array.Data.File.Cy5==targetsRef.ch2$Array.Data.File.Cy3) ) {
        colsSel <- apply(targetsCa.ch1==targetsRef.ch2,2,function(x) sum(x) != length(x) )
        targetsSwap1 <- cbind(targetsCa.ch1,targetsRef.ch2[,colsSel],stringsAsFactors=FALSE)
        targetsSwap1 <- targetsSwap1[,apply(targetsSwap1,2,function(x) length(unique(x)) != 1 )
        ##add Cy5 and Cy3 columns
        targetsSwap1$Cy5 <- targetsSwap1$Sample.Name
        targetsSwap1$Cy3 <- "Ref"
        targetsSwap1 <- targetsSwap1[,order(colnames(targetsSwap1))]
        ##remove dye extension to Scan name
        colnames(targetsSwap1) <- gsub("\\.Cy5$","",colnames(targetsSwap1))
        ##reorder by sample name in Cy5
        targetsSwap1 <-         targetsSwap1[order(targetsSwap1$Cy5),]
        print("Assembling the targets object for the first set of swapped hybridizations")
 } else {
        stop("Check objects")
 }
[1] "Assembling the targets object for the first set of swapped hybridizations"
```

Below is the R code chunk used to assemble the phenotypic information associated with the second set of hybridizations.

```
> ###create new target objects for SECOND set of swapped hybs and keep useful columns
> if ( all(targetsCa.ch2$Array.Data.File.Cy3==targetsRef.ch1$Array.Data.File.Cy5) ) {
        colsSel <- apply(targetsCa.ch2==targetsRef.ch1,2,function(x) sum(x) != length(x) )
        targetsSwap2 <- cbind(targetsCa.ch2,targetsRef.ch1[,colsSel],stringsAsFactors=FALSE)
        targetsSwap2 <- targetsSwap2[,apply(targetsSwap2,2,function(x) length(unique(x)) != 1 )
        ##add Cy5 and Cy3 columns
        targetsSwap2$Cy5 <- "Ref"
        targetsSwap2$Cy3 <- targetsSwap2$Sample.Name
        targetsSwap2 <- targetsSwap2[,order(colnames(targetsSwap2))]
        ##remove dye extension to Scan name
        colnames(targetsSwap2) <- gsub("\\.Cy3$","",colnames(targetsSwap2))
        ##reorder by sample name in Cy3
        targetsSwap2 <-         targetsSwap2[order(targetsSwap2$Cy3),]
        print("Assembling the targets object for the second set of swapped hybridizations")
 } else {
        stop("Check objects")
 }
[1] "Assembling the targets object for the second set of swapped hybridizations"
```

Below is the R code chunk used to assemble the RGlist object for the first set of hybridizations of the ETABM-77 series.

```
> ##Define the columns which will be read from the raw files selection
> colsList <- list(Rf="Feature Extraction Software:rMedianSignal",
                   Rb="Feature Extraction Software:rBGMedianSignal",
                   Gf="Feature Extraction Software:gMedianSignal",
```

```
                    Gb="Feature Extraction Software:gBGMedianSignal",
                    logRatio="Feature Extraction Software:LogRatio",
                    logRatioError="Feature Extraction Software:LogRatioError")
> ###Swap set 1 has Reference RNA in Cy3 channel
> table(targetsSwap1$Cy3)

Ref
307

> ###This creates an instance of class "RGList", selecting specific columns from the raw data:
> RGcy3 <- read.maimages(paste(targetsSwap1$Array.Data.File, ".gz",  sep=""),
                         source="generic",
                         columns=colsList, annotation="Reporter identifier",
                         path=etabm77filesLoc, verbose=FALSE)
> ###Add targets information
> RGcy3$targets <- targetsSwap1
> ###Add sample names as column names with prefix
> colnames(RGcy3) <- paste("BC",targetsSwap1$Sample.Name,sep=".")
> ###Format Reporter identifiers
> RGcy3$genes$ID <- gsub(".+A-MEXP-318\\.", "", RGcy3$genes[, "Reporter identifier"])
> ###Check dimensions
> dim(RGcy3)

[1] 1900  307
```

Below is the `R` code chunk used to assemble the `RGlist` object for the second set of hybridizations of the `ETABM-77` series.

```
> ###This creates an instance of class "RGList", selecting specific columns from the raw data:
> RGcy5 <- read.maimages(paste(targetsSwap1$Array.Data.File, ".gz",  sep=""),
                         source="generic",
                         columns=colsList, annotation="Reporter identifier",
                         path=etabm77filesLoc, verbose=FALSE)
> ###Swap set 2 has Reference RNA in Cy5 channel
> table(targetsSwap2$Cy5)

Ref
307

> ###Add targets information
> RGcy5$targets <- targetsSwap2
> ###Format Reporter identifiers
> RGcy5$genes$ID <- gsub(".+A-MEXP-318\\.", "", RGcy5$genes[, "Reporter identifier"])
> ###Add sample names as column names with prefix
> colnames(RGcy5) <- paste("BC",targetsSwap2$Sample.Name,sep=".")
> ###Check dimensions
> dim(RGcy5)

[1] 1900  307
```

Below is the `R` code to check whether all data read is of teh correct mode (i.e. numeric values are of mode "numeric", and so on)

```
> ###Check mode of read data for first set of hybridizations
> sapply(RGcy3, mode)
```

```
           R            Rb             G            Gb      logRatio
    "numeric"     "numeric"     "numeric"     "numeric"     "numeric"
logRatioError       targets         genes        source
    "numeric"        "list"        "list"   "character"
```

```
> ###Since RGc3$logRatio is character convert to numeric
> RGcy3$logRatio <- apply(RGcy3$logRatio, 2, as.numeric)
> ###Check mode of read data for second set of hybridizations
> sapply(RGcy5, mode)
```

```
           R            Rb             G            Gb      logRatio
    "numeric"     "numeric"     "numeric"     "numeric"     "numeric"
logRatioError       targets         genes        source
    "numeric"        "list"        "list"   "character"
```

```
> ###Since RGc3$logRatio is character convert to numeric
> RGcy5$logRatio <- apply(RGcy5$logRatio, 2, as.numeric)
```

Below is the `R` code chunkto process the microarray feature annotation for the 1.9k MammaPrint array, as it is provided in the ArrayExpress database (file `A-MEXP-318.adf.txt`. We will then map the microarray features from thes new array to the 231-gene and the 70-gene prognostic signatures. We will finally add annotation information to the `RGList` objects created above.

```
> ###Read genes information for MammaPrint array from the A-MEXP-318.adf.txt
> genes <- read.table(dir(etabm77filesLoc, full.names=TRUE, pattern="A-MEXP-318.adf.txt"),
                      comment.char="", skip=21, sep="\t", header=TRUE, stringsAsFactors=FALSE)
> ###Remove special characters like points in the colums header
> colnames(genes) <- gsub("\\.$","",gsub("\\.\\.",".",colnames(genes)))
> ###Use the annotation information previously processed
> str(progSig)
```

```
List of 7
 $ gns231acc : chr [1:231] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231name: chr [1:161] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
 $ gns231any : chr [1:392] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70acc  : chr [1:70] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns70name : chr [1:50] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
 $ gns70any  : chr [1:120] "AL080059" "Contig63649_RC" "Contig46218_RC" "NM_016359" ...
 $ gns231Cors:'data.frame':        394 obs. of  2 variables:
  ..$ ID        : chr [1:394] "LOC51203" "ALDH4" "FGF18" "BBC3" ...
  ..$ gns231Cors: num [1:394] -0.425 0.421 0.411 0.407 0.402 ...
```

```
> ###Count the features mapped to the 231-gene prognostic signature by accession number
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231acc)
```

```
  Block.Column                     Block.Row
             0                             0
        Column                           Row
             0                             0
```

```
                Reporter.Name          Comment.AEReporterName
                            0                              68
  Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                            7                               0
           Reporter.Group.role                    Control.Type
                            0                               0
```

```
> ###Count the features mapped to the 231-gene prognostic signature by gene symbol
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231name)
```

```
                 Block.Column                      Block.Row
                            0                              0
                       Column                         Row
                            0                              0
                Reporter.Name          Comment.AEReporterName
                            0                             161
  Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                            0                               0
           Reporter.Group.role                    Control.Type
                            0                               0
```

```
> ###Count the features mapped to the 231-gene prognostic signature by gene symbol or accession
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns231any)
```

```
                 Block.Column                      Block.Row
                            0                              0
                       Column                         Row
                            0                              0
                Reporter.Name          Comment.AEReporterName
                            0                             229
  Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                            7                               0
           Reporter.Group.role                    Control.Type
                            0                               0
```

```
> ###Add mapping information for 229 features using both gene symbols and accession numbers
> genes$genes231 <- genes$Comment.AEReporterName %in% progSig$gns231any
> ###Count the features mapped to the 70-gene prognostic signature by accession number
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70acc)
```

```
                 Block.Column                      Block.Row
                            0                              0
                       Column                         Row
                            0                              0
                Reporter.Name          Comment.AEReporterName
                            0                              19
  Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                            1                               0
           Reporter.Group.role                    Control.Type
                            0                               0
                     genes231
```

```
                            0

> ###Count the features mapped to the 70-gene prognostic signature by gene symbol
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70name)

                  Block.Column                      Block.Row
                             0                              0
                        Column                         Row
                             0                              0
                  Reporter.Name          Comment.AEReporterName
                             0                             50
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             0                              0
            Reporter.Group.role                   Control.Type
                             0                              0
                      genes231
                             0

> ###Count the features mapped to the 70-gene prognostic signature by gene symbol or accession r
> apply(genes, 2, function(x, y) { sum(unique(x) %in% y) }, y = progSig$gns70any)

                  Block.Column                      Block.Row
                             0                              0
                        Column                         Row
                             0                              0
                  Reporter.Name          Comment.AEReporterName
                             0                             69
   Reporter.Database.Entry.embl Reporter.Database.Entry.unigene
                             1                              0
            Reporter.Group.role                   Control.Type
                             0                              0
                      genes231
                             0

> ###Add mapping information for 229 features using both gene symbols and accession numbers
> genes$genes70 <- genes$Comment.AEReporterName %in% progSig$gns70any
> ###The resulting gene annotation file
> str(genes)

'data.frame':        1900 obs. of  12 variables:
 $ Block.Column                  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Block.Row                     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Column                        : int  1 2 13 20 40 42 48 49 50 1 ...
 $ Row                           : int  1 1 1 1 1 1 1 1 1 2 ...
 $ Reporter.Name                 : chr  "AGEN000001" "AGEN000002" "AGEN000013" "AGEN000020" ...
 $ Comment.AEReporterName        : chr  "Pro25G" "3xSLv1" "eQC" "eQC" ...
 $ Reporter.Database.Entry.embl  : chr  "" "" "" "" ...
 $ Reporter.Database.Entry.unigene: chr  "" "" "" "" ...
 $ Reporter.Group.role           : chr  "Control" "Control" "Control" "Control" ...
 $ Control.Type                  : chr  "control_biosequence" "control_biosequence" "control_bi
 $ genes231                      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
 $ genes70                        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
> ###Add original correlation from the van't Veer study
> gns231Cors <- progSig$gns231Cors
> gns231Cors <- gns231Cors[gns231Cors$ID %in% genes$Comment.AEReporterName,]
> ###Merge and remove duplicates
> genes <- merge(genes, gns231Cors, all.x=TRUE, all.y=FALSE,
                 by.x="Comment.AEReporterName", by.y="ID")
> genes <- genes[!duplicated(genes$Reporter.Name),]
> ###Check genes
> str(genes)
```

```
'data.frame':        1900 obs. of  13 variables:
 $ Comment.AEReporterName         : chr  "3xSLv1" "3xSLv1" "3xSLv1" "3xSLv1" ...
 $ Block.Column                   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Block.Row                      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Column                         : int  2 48 1 2 48 49 50 1 4 7 ...
 $ Row                            : int  1 1 2 2 2 2 2 9 9 9 ...
 $ Reporter.Name                  : chr  "AGEN000002" "AGEN000048" "AGEN000051" "AGEN000052" ...
 $ Reporter.Database.Entry.embl   : chr  "" "" "" "" ...
 $ Reporter.Database.Entry.unigene: chr  "" "" "" "" ...
 $ Reporter.Group.role            : chr  "Control" "Control" "Control" "Control" ...
 $ Control.Type                   : chr  "control_biosequence" "control_biosequence" "control_bi
 $ genes231                       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ genes70                        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ gns231Cors                     : num  NA NA NA NA NA NA NA NA NA NA ...
```

Below is the `R` code chunk used to add the feature annotation the `RGList` objects containing the expression data (first set of hybridizations).

```
> ###Compare genes between paltform and RGList instances: first set of hybridizations
> if (nrow(genes)==nrow(RGcy3)) {
        ##Compare the identifiers' order
        if ( !all(genes$Reporter.Name == RGcy3$genes$ID) ) {
                ##Reorder if needed
                RGcy3 <- RGcy3[order(RGcy3$genes$ID),]
                genes <- genes[order(genes$Reporter.Name),]
                ##check for order AGAIN
                if ( all(genes$Reporter.Name == RGcy3$genes$ID) ) {
                        ##Substitute genes or stop
                        RGcy3$genes <- genes
                } else {
                        stop("Wrong gene order, check objects")
                }
        } else {
                print("Updating gene annotation")
                RGcy3$genes <- genes
        }
 } else {
        stop("Wrong number of features, check objects")
```

```
 }
> ###Rename
> buyseRGcy3 <- RGcy3
```

Below is the R code chunk used to add the feature annotation the RGList objects containing the expression data (second set of hybridizations).

```
> ###Compare genes between paltform and RGList instances: second set of hybridizations
> if (nrow(genes)==nrow(RGcy5)) {
        ##Compare the identifiers' order
        if ( !all(genes$Reporter.Name == RGcy5$genes$ID) ) {
                ##Reorder if needed
                RGcy5 <- RGcy5[order(RGcy5$genes$ID),]
                genes <- genes[order(genes$Reporter.Name),]
                ##check for order AGAIN
                if ( all(genes$Reporter.Name == RGcy5$genes$ID) ) {
                        ##Substitute genes or stop
                        RGcy5$genes <- genes
                } else {
                        stop("Wrong gene order, check objects")
                }
        } else {
                print("Updating gene annotation")
                RGcy5$genes <- genes
        }
 } else {
        stop("Wrong number of features, check objects")
 }
> ###Rename
> buyseRGcy5 <- RGcy5
```

4 Phenotypic information processing

4.1 Glas cohort: patients' phenotype curation

The chunk of R code below was used to process the clinical information associated with the 162 patients of the ArrayExpress "E-TABM-115" series, and define the final "good" and "bad" prognosis patients groups, as well as other end-points that were investigated in the our study. To this end we first identified the putative 78 patients from the van't Veer cohort [7], and the 84 cases from the Van de Vijver cohort [6] that were combined to assemble the final Glas data set. Since this information was not provided with the ArrayExpress "E-TABM-115" series, we analyzed the raw data hybridization file names, and used the 3-digits suffix code present in exactly 84 files, as follows:

```
> ###Retrieve the phenotypic information from one of the RGList intances
> phenoGlasCy3 <- glasRGcy3$targets
> phenoGlasCy5 <- glasRGcy5$targets
> ###Identification of the 78 samples from van't Veer and the 84 from VanDeVijver
> table(gsub(".+_", "", phenoGlasCy3$Scan.Name) == gsub(".+_", "", phenoGlasCy5$Scan.Name))
```

```
FALSE  TRUE
   78    84
```

```
> ##################################################
> ###There are 78 hybridizations with 1-digit suffixes, and 84 of them with 3-digit suffixes
> table(nchar(gsub(".+_", "", phenoGlasCy5$Scan.Name)))
```

```
 1  3
78 84
```

```
> table(nchar(gsub(".+_", "", phenoGlasCy3$Scan.Name)))
```

```
 1  3
78 84
```

```
> ###Use the hybridization file names suffixes to define the putative cohort of origin: FIRST SE
> phenoGlasCy5$putativeCohort <- factor(nchar(gsub(".+_", "", phenoGlasCy5$Scan.Name)))
> levels(phenoGlasCy5$putativeCohort) <- c("putativeVantVeer", "putativeVanDeVijver")
> ###Use the hybridization file names suffixes to define the putative cohort of origin: SECOND S
> phenoGlasCy3$putativeCohort <- factor(nchar(gsub(".+_", "", phenoGlasCy3$Scan.Name)))
> levels(phenoGlasCy3$putativeCohort) <- c("putativeVantVeer", "putativeVanDeVijver")
> ###Show counts FIRST SET
> table(phenoGlasCy5$putativeCohort)
```

```
  putativeVantVeer putativeVanDeVijver
                78                   84
```

```
> ##################################################
> ###Show counts SECOND SET
> table(phenoGlasCy3$putativeCohort)
```

```
  putativeVantVeer putativeVanDeVijver
                78                   84
```

We subsequently processed the "Factor.Value.overall_survival" character strings contained in the "E-TABM-115" SDRF table to define the actual overall survival (OS) time, the OS event status, and the 10-year survival status for each patient, as shown below. This was possible only for the set of hybridization for which the clinical information was provided in ArrayExpress (see `glasRGcy5$targets` below).

```
> ###Process the overall survival (OS) character strings  and make numeric
> OS <- gsub("\\s.+", "", phenoGlasCy5$Factor.Value.overall_survival)
> phenoGlasCy5$OS <- as.numeric(OS)
> ####Process OS event and make numeric: missing data, labeled with "n/a" strings are turned int
> phenoGlasCy5$OSevent <- as.numeric(phenoGlasCy5$Factor.Value.Event_Death)
> ###Count missing information by counting the number of NA
> sum(is.na(phenoGlasCy5$OSevent))
```

```
[1] 17
```

```
> ####Create binary OS at 10 years groups
> phenoGlasCy5$TenYearSurv <- phenoGlasCy5$OS < 10 & phenoGlasCy5$OSevent == 1
> phenoGlasCy5$TenYearSurv[is.na(phenoGlasCy5$OSevent)] <- NA
```

The overall survival information summary for the Glas cohort is shown below:

```
> ###Count OS events
> table(phenoGlasCy5$OSevent)

  0   1
104  41

> ###Count survival at 10 years status
> table(phenoGlasCy5$TenYearSurv)

FALSE  TRUE
  106    39
```

We subsequently processed the "Factor.Value.Time_to_development_of_distant_metastases" character strings contained in the "E-TABM-115" SDRF table to define the actual time to metastasis (TTM) time, the TTM event status, and the 5-year distant recurrence status for each patient, as shown below. This was possible only for the set of hybridization for which the clinical information was provided in ArrayExpress (see `glasRGcy5$targets` below).

```
> ###Process time metastases (TTM) character strings and make numeric
> ###Note that missing data, labeled with "n/a" strings are turned into NA
> TTM <- phenoGlasCy5$Factor.Value.Time_to_development_of_distant_metastases
> phenoGlasCy5$TTM <- as.numeric(gsub("\\s.+", "", TTM))
> ####Process TTM event and make numeric
> phenoGlasCy5$TTMevent <- as.numeric(phenoGlasCy5$Factor.Value.Event_distant_metastases)
> ###Use follow-up time from OS as TTM for patients withouth metastasis event
> phenoGlasCy5$TTM[is.na(phenoGlasCy5$TTM)] <- phenoGlasCy5$OS[is.na(phenoGlasCy5$TTM)]
> ####Create binary TTM at 5 years groups
> phenoGlasCy5$FiveYearMetastasis <- phenoGlasCy5$TTM < 5 & phenoGlasCy5$TTMevent == 1
```

The time to metastasis information summary, and the five year distant recurrence status, which corresponds to the "good" and "bad" prognosis groups used in the Glas study, are shown below.

```
> ###Count TTM events
> table(phenoGlasCy5$TTMevent)

  0   1
106  56

> ###Count metastasis events by putative cohort of origin
> table(phenoGlasCy5$FiveYearMetastasis, phenoGlasCy5$putativeCohort)

       putativeVantVeer putativeVanDeVijver
  FALSE               44                  72
  TRUE                34                  12
```

We finally replaced the processed phenotypic information in the corresponding `RGList-class` instances:

```
> ###Substitute in the RGLists
> glasRGcy3$targets <- phenoGlasCy3
> glasRGcy5$targets <- phenoGlasCy5
```

Below is the `R` code chunk used to save the final `RGList` objects.

```
> ###Save the final ExpressionSet object: not run
> dataDirLoc <- system.file("data", package = "mammaPrintData")
> save(glasRGcy5, glasRGcy3, file=paste(dataDirLoc, "/glasRG.rda", sep=""))
```

4.2 Buyse cohort: patients' phenotype curation

The chunk of R code below was used to process the clinical information associated with the 307 analyzed patients that were retrieved from the ArrayExpress "E-TABM-77" series, to define the final "good" and "bad" prognosis patients groups, as well as other end-points that were investigated in the present study. We hence processed the "Factor.Value.SurvivalTime" character strings contained in the "E-TABM-77" SDRF table to define the actual overall survival (OS) time, the OS event status, and the 10-year survival group for each patient, as shown below

```
> ###Retrieve the phenotypic information from one of the RGList intances
> phenoBuyseCy3 <- buyseRGcy3$targets
> phenoBuyseCy5 <- buyseRGcy5$targets
> ###Process overall survival (OS) character strings and make numeric: BOTH SETS
> OScy5 <- phenoBuyseCy5$Factor.Value.SurvivalTime
> phenoBuyseCy5$OS <- as.numeric(gsub("\\s.+", "", OScy5))
> OScy3 <- phenoBuyseCy3$Factor.Value.SurvivalTime
> phenoBuyseCy3$OS <- as.numeric(gsub("\\s.+", "", OScy3))
> ###Create the OS event by processing the OS character string with grep
> ###and the "plus" pattern making it a numeric vector: BOTH SETS
> phenoBuyseCy5$OSevent <- 1*( ! 1:nrow(phenoBuyseCy5) %in% grep("plus", OScy5))
> phenoBuyseCy3$OSevent <- 1*( ! 1:nrow(phenoBuyseCy3) %in% grep("plus", OScy3))
> ####Create binary OS at 10 years groups: BOTH SETS
> phenoBuyseCy5$TenYearSurv <- phenoBuyseCy5$OS < 10 & phenoBuyseCy5$OSevent == 1
> phenoBuyseCy3$TenYearSurv <- phenoBuyseCy3$OS < 10 & phenoBuyseCy3$OSevent == 1
```

The overall survival information summary for the Buyse cohort is shown below:

```
> ###Count OS survival events: BOTH SETS
> table(phenoBuyseCy5$OSevent)

  0   1
225  82

> table(phenoBuyseCy3$OSevent)

  0   1
225  82

> ###Count survival at 10 years status: BOTH SETS
> table(phenoBuyseCy5$TenYearSurv)

FALSE  TRUE
  242    65

> table(phenoBuyseCy3$TenYearSurv)

FALSE  TRUE
  242    65
```

We subsequently processed the "Factor.Value.Disease.Free.Survival" character strings contained in the "E-TABM-77" SDRF table to define the actual disease free survival (DFS) time, and the DFS event status for each patient, as shown below.

```
> ###Process disease free survival (DFS) character strings and make numeric: BOTH SETS
> DFScy5 <- phenoBuyseCy5$Factor.Value.Disease.Free.Survival
> phenoBuyseCy5$DFS <- as.numeric(gsub("\\s.+", "", DFScy5))
> DFScy3 <- phenoBuyseCy3$Factor.Value.Disease.Free.Survival
> phenoBuyseCy3$DFS <- as.numeric(gsub("\\s.+", "", DFScy3))
> ###Create the DFS event by processing the DFS character string with grep
> ###and the "plus" pattern making it a numeric vector: BOTH SETS
> phenoBuyseCy5$DFSevent <- 1*( ! 1:nrow(phenoBuyseCy5) %in% grep("plus", DFScy5))
> phenoBuyseCy3$DFSevent <- 1*( ! 1:nrow(phenoBuyseCy3) %in% grep("plus", DFScy3))
> ####Create binary DFS at 5 years groups:BOTH SETS
> phenoBuyseCy5$FiveYearDiseaseFree <- (phenoBuyseCy5$DFS < 5
                                        & phenoBuyseCy5$DFSevent == 1)
> phenoBuyseCy3$FiveYearDiseaseFree <- (phenoBuyseCy3$DFS < 5
                                        & phenoBuyseCy3$DFSevent == 1)
```

The disease free survival information summary for the Buyse cohort is shown below:

```
> ###Count DFS survival events: BOTH SETS
> table(phenoBuyseCy5$DFSevent)

  0   1
168 139

> table(phenoBuyseCy3$DFSevent)

  0   1
168 139

> ###Count recurrence at 5 years status: BOTH SETS
> table(phenoBuyseCy5$FiveYearDiseaseFree)

FALSE  TRUE
  226    81

> table(phenoBuyseCy3$FiveYearDiseaseFree)

FALSE  TRUE
  226    81
```

We subsequently processed the "Factor.Value.DistantMetastasis.Free.Survival" character strings contained in the "E-TABM-77" SDRF table to define the actual time to metastasis (TTM) time, the TTM event status, and the 5-year recurrence status for each patient, as shown below.

```
> ###Process time metastases (TTM) character strings and make numeric: BOTH SETS
> TTMcy5 <- phenoBuyseCy5$Factor.Value.DistantMetastasis.Free.Survival
> phenoBuyseCy5$TTM <- as.numeric(gsub("\\s.+", "", TTMcy5))
> TTMcy3 <- phenoBuyseCy3$Factor.Value.DistantMetastasis.Free.Survival
> phenoBuyseCy3$TTM <- as.numeric(gsub("\\s.+", "", TTMcy3))
> ###Create the TTM event by processing the TTM character string with grep
> ###and the "plus" pattern making it a numeric vector: BOTH SETS
```

```
> phenoBuyseCy5$TTMevent <- 1*( ! 1:nrow(phenoBuyseCy5) %in% grep("plus", TTMcy5))
> phenoBuyseCy3$TTMevent <- 1*( ! 1:nrow(phenoBuyseCy3) %in% grep("plus", TTMcy3))
> ####Create binary TTM at 5 years groups: BOTH SETS
> phenoBuyseCy5$FiveYearRecurrence <- (phenoBuyseCy5$TTM < 5
                                       & phenoBuyseCy5$TTMevent == 1)
> phenoBuyseCy3$FiveYearRecurrence <- (phenoBuyseCy3$TTM < 5
                                       & phenoBuyseCy3$TTMevent == 1)
```

The time to metastasis information summary, and the five year distant recurrence status, which corresponds to the "good" and "bad" prognosis groups used in the Buyse study, are shown below.

```
> ###Count TTM survival events: BOTH SETS
> table(phenoBuyseCy5$TTMevent)

  0   1
230  77

> table(phenoBuyseCy3$TTMevent)

  0   1
230  77

> ###Count recurrence at 5 years status: BOTH SETS
> table(phenoBuyseCy5$FiveYearRecurrence)

FALSE  TRUE
  260    47

> table(phenoBuyseCy3$FiveYearRecurrence)

FALSE  TRUE
  260    47

> ###Count metastasis events by European center: BOTH SETS
> table(phenoBuyseCy5$FiveYearRecurrence, phenoBuyseCy5$Characteristics.BioSourceProvider)

       CRH GUY IGR JRH KAR
  FALSE  54  37  84  32  53
  TRUE    5  16  12   7   7

> table(phenoBuyseCy3$FiveYearRecurrence, phenoBuyseCy3$Characteristics.BioSourceProvider)

       CRH GUY IGR JRH KAR
  FALSE  54  37  84  32  53
  TRUE    5  16  12   7   7
```

We finally identified the patients that were excluded from the analysis in the original study by Buyse and colleagues [2] beacuse of missing clinical information.

```
> ###Exclude patients with missing ER status: BOTH SETSXS
> phenoBuyseCy5$toExclude <- phenoBuyseCy5$Factor.Value.ER.status=="unknown"
> phenoBuyseCy3$toExclude <- phenoBuyseCy3$Factor.Value.ER.status=="unknown"
```

We finally replaced the processed phenotypic information in the corresponding `RGList-class` instances:

```
> ###Substitute in the RGLists
> buyseRGcy3$targets <- phenoBuyseCy3
> buyseRGcy5$targets <- phenoBuyseCy5
```

Below is the R code chunk used to save the final RGList objects.

```
> ###Save RGList instances for later use: not run
> dataDirLoc <- system.file("data", package = "mammaPrintData")
> save(buyseRGcy5, buyseRGcy3, file=paste(dataDirLoc, "/buyseRG.rda", sep=""))
```

5 System Information

Session information:

```
> toLatex(sessionInfo())
```

- R version 4.0.0 (2020-04-24), x86_64-pc-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=C, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Running under: Ubuntu 18.04.4 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.11-bioc/R/lib/libRblas.so
- LAPACK: /home/biocbuild/bbs-3.11-bioc/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: Biobase 2.48.0, BiocGenerics 0.34.0, gdata 2.18.0, limma 3.44.1
- Loaded via a namespace (and not attached): BiocManager 1.30.10, compiler 4.0.0, gtools 3.8.2, tools 4.0.0

6 References

References

[1] A. Brazma, H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G. G. Lara, A. Oezcimen, P. Rocca-Serra, and S. A. Sansone. ArrayExpress–a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res*, 31(1):68–71, 2003. 1362-4962 (Electronic) Journal Article.

[2] M. Buyse, S. Loi, L. van't Veer, G. Viale, M. Delorenzi, A. M. Glas, M. S. d'Assignies, J. Bergh, R. Lidereau, P. Ellis, A. Harris, J. Bogaerts, P. Therasse, A. Floore, M. Amakrane, F. Piette, E. Rutgers, C. Sotiriou, F. Cardoso, and M. J. Piccart. Validation and clinical utility of a 70-gene prognostic signature for women with node-negative breast cancer. *J Natl Cancer Inst*, 98(17):1183–92, 2006. 1460-2105 (Electronic) Journal Article Multicenter Study Research Support, Non-U.S. Gov't Validation Studies.

[3] A. M. Glas, A. Floore, L. J. Delahaye, A. T. Witteveen, R. C. Pover, N. Bakx, J. S. Lahti-Domenici, T. J. Bruinsma, M. O. Warmoes, R. Bernards, L. F. Wessels, and L. J. Van't Veer. Converting a breast cancer microarray signature into a high-throughput diagnostic test. *BMC Genomics*, 7:278, 2006. 1471-2164 (Electronic) Journal Article Research Support, Non-U.S. Gov't.

[4] Luigi Marchionni, Renee F Wilson, Spyridon S Marinopoulos, Antonio C Wolff, Giovanni Parmigiani, Eric B Bass, and Steven N Goodman. Impact of gene expression profiling tests on breast cancer outcomes. *Evid Rep Technol Assess (Full Rep)*, (160):1–105, Dec 2007.

[5] Luigi Marchionni, Renee F Wilson, Antonio C Wolff, Spyridon Marinopoulos, Giovanni Parmigiani, Eric B Bass, and Steven N Goodman. Systematic review: gene expression profiling assays in early-stage breast cancer. *Ann Intern Med*, 148(5):358–369, Mar 2008.

[6] M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, 347(25):1999–2009, 2002. 1533-4406 (Electronic) Evaluation Studies Journal Article.

[7] L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–6, 2002. 0028-0836 (Print) Journal Article.