EMBO03 Lab 1: R and Bioconductor Basics

Sandrine Dudoit and Robert Gentleman

March 14, 2003

1 R and Bioconductor WWW resources

For software and documentation consult

- Main R project website: `www.R-project.org`.
- Comprehensive R Archive Network (CRAN): `cran.r-project.org`.
  Base system and contributed packages (Linux, MacOS, Windows), manuals, tutorials, R News.
  For Windows, there is an installer for the main R system. Contributed packages from CRAN can be installed using the "Install package from CRAN ..." item on the "Packages" menu. For other packages, you can use the "Install package from local zip file ..." item.
- Bioconductor website: `www.bioconductor.org`.
  Software packages, vignettes, datasets, short course materials.
  To install Bioconductor packages, use the install script (`getBioC.R`) provided under the "HowTo" link of the "Software" section of the website.

The labs for this short course are included in the `EMBO03` R package. They are executable documents that mix text and code, and are created and run using the Sweave system (`www.ci.tuwien.ac.at/~leisch/Sweave/`). We will use the `vExplorer` function from the `tkWidgets` package to step through the labs interactively. After loading the `EMBO03` package, using the command `library(EMBO03)`, type `vExplorer()` and select the `EMBO03` package using the widget.

2 Getting started

Some useful commands for getting help and sample scripts demonstrating software functionality

```
> help.start()
> apropos("mean")
> ? mean
> example("mean")
> demo()
> demo(image)
```

The functions `getwd` and `setwd` are used to get or set the working directory. In Windows, you can also use the "Change dir ..." item in the "File" menu.

```
> getwd()
```

```
[1] "/home/sandrine/Tmp"
```

To load packages, use the function `library` (or the "Load package ..." item on the "Packages" menu in Windows)

```
> library(Biobase)
> library(tkWidgets)
> library(genefilter)
```

The functions `save` and `save.image` are used to write external representations of R objects to a specified file. The objects can be read back from the file (`.RData`) at a later date by using the `load` function. `save.image()` is called when you answer "yes" when exiting R using the command `q()`. In Windows, you can use the "Load Workspace... ", "Save Workspace... ", and "Exit" items in the "File" menu.

Each Bioconductor package contains a *vignette*, which is an executable document providing a step-by-step overview of the package functionality. These documents are created using the `Sweave` function from the `tools` package. You can access the vignettes (`.Rnw` source and `.pdf` output) for a given package under the "Accompanying documentation" section of the HTML help page for the package. These files are also available in the `"doc"` subdirectory of the installed package. To view a list all available vignettes use `openVignette()`. Selecting a vignette ID will display the specified PDF file. The `vExplorer` function provides a graphical interface for viewing and executing code chunks from vignettes. We will use this function for the labs.

3 Bioconductor base package: `Biobase`

`Biobase` is the base package of Bioconductor. It contains the main *class/method* definitions for handling microarray and other types of genomic data. It also contains any reusable (or non-specific) functions needed by other packages.

The main class for (pre-processed) microarray data objects is `exprSet`. The S4 `methods` package has introduced substantial new capabilities into R. To obtain the manual pages for S4 classes you should use the following syntax `class?exprSet`. Consult the help file for a description of the *slots* of the `exprSet` class and basic methods which operate on objects of this class.

```
> slotNames("exprSet")
```

```
[1] "exprs"       "se.exprs"    "phenoData"   "description" "annotation"
[6] "notes"
```

An `exprSet` basically consists of the genes $\times$ arrays matrix of expression measures, optionally a set of standard errors for those estimates, the related experimental metadata (who did what, when, and to what), and the phenotypic data. Here, phenotype is interpreted quite broadly – it represents any characteristics of the target sample (e.g., for tumor mRNA samples, patient survival, age, sex, treatment).

The ALL/AML leukemia dataset of Golub et al. (1999) will be used as a case study. For more background on the experiments: `www-genome.wi.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43`. The data are available from the authors in an online repository (`www-genome.wi.mit.edu/mpr/data_set_ALL_AML.html`) and are included in an R data package, `golubEsets`, suitable for use in this lab. This dataset comes from a study of gene expression in two types of acute leukemias: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix high-density oligonucleotide arrays (HU6800 chip) containing probes for approximately $6,800$ human genes and ESTs. The chip actually contains 7,129 different probe sets; some of these map to the same genes and others are used for quality control purposes. The data comprise 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML. Samples are divided into a learning set with 38 observations and a test set of 34 observations. Here, we assume that the data have been suitably pre-processed, i.e., image analysis, normalization, and computation of expression measures were already performed. You are referred to the `affy` package from Bioconductor for relevant software and documentation for pre-processing Affymetrix data, and to the `marray` suite for pre-processing two-color spotted cDNA array data.

For a description of the Golub et al. (1999) dataset type `? golubTrain` and to load these data

```
> library(golubEsets)
> data(golubTrain)
> data(golubTest)
> data(golubMerge)
```

The microarray expression measures and clinical variables corresponding to each of the 38 training mRNA samples are stored in an object of class `exprSet`. For information on the training data `golubTrain`

```
> class(golubTrain)

[1] "exprSet"

> slotNames(golubTrain)

[1] "exprs"       "se.exprs"    "phenoData"   "description" "annotation"
[6] "notes"

> golubTrain

Expression Set (exprSet) with
        7129 genes
        38 samples
                 phenoData object with 11 variables and 38 cases
         varLabels
                Samples: Samples
                ALL.AML: ALL.AML
                BM.PB: BM.PB
                T.B.cell: T.B.cell
                FAB: FAB
                Date: Date
                Gender: Gender
                pctBlasts: pctBlasts
                Treatment: Treatment
                PS: PS
                Source: Source
```

The phenotypic data are stored in a separate, but linked, object of class `phenoData`. An object of class `phenoData` is a combination of a data frame containing a number of variables for each array and a list that explains what each variable represents. This information is usually relegated to a help page but we felt that it was important to keep it more closely associated with the expression measures. You can obtain and manipulate the `phenoData` object corresponding to a particular `exprSet` object using specific methods, as described in the manual page. To extract only the sample level variables

```
> phenoTrain <- phenoData(golubTrain)
> class(phenoTrain)

[1] "phenoData"
```

```
> slotNames(phenoTrain)

[1] "pData"     "varLabels"

> varLabels(phenoTrain)

$Samples
[1] "Samples"

$ALL.AML
[1] "ALL.AML"

$BM.PB
[1] "BM.PB"

$T.B.cell
[1] "T.B.cell"

$FAB
[1] "FAB"

$Date
[1] "Date"

$Gender
[1] "Gender"

$pctBlasts
[1] "pctBlasts"

$Treatment
[1] "Treatment"

$PS
[1] "PS"

$Source
[1] "Source"

> pData(phenoTrain)

   Samples ALL.AML BM.PB T.B.cell  FAB       Date Gender pctBlasts Treatment
1        1     ALL    BM   B-cell <NA>   9/4/1996      M        NA      <NA>
```

```
2        2     ALL    BM   T-cell <NA>       <NA>      M        NA      <NA>
3        3     ALL    BM   T-cell <NA>       <NA>      M        NA      <NA>
4        4     ALL    BM   B-cell <NA>       <NA>   <NA>        NA      <NA>
5        5     ALL    BM   B-cell <NA>       <NA>   <NA>        NA      <NA>
6        6     ALL    BM   T-cell <NA>       <NA>      M        NA      <NA>
7        7     ALL    BM   B-cell <NA>  3/25/1983      F        NA      <NA>
8        8     ALL    BM   B-cell <NA>       <NA>      F        NA      <NA>
9        9     ALL    BM   T-cell <NA>       <NA>      M        NA      <NA>
10      10     ALL    BM   T-cell <NA>  7/23/1987      M        NA      <NA>
11      11     ALL    BM   T-cell <NA>  6/25/1985      M        NA      <NA>
12      12     ALL    BM   B-cell <NA>  9/17/1985      F        NA      <NA>
13      13     ALL    BM   B-cell <NA>  7/27/1988      F        NA      <NA>
14      14     ALL    BM   T-cell <NA> 11/27/1987      M        NA      <NA>
15      15     ALL    BM   B-cell <NA>  3/25/1989      F        NA      <NA>
16      16     ALL    BM   B-cell <NA>  2/12/1990      M        NA      <NA>
17      17     ALL    BM   B-cell <NA>  9/26/1990      M        NA      <NA>
18      18     ALL    BM   B-cell <NA>       <NA>      F        NA      <NA>
19      19     ALL    BM   B-cell <NA>       <NA>   <NA>        NA      <NA>
20      20     ALL    BM   B-cell <NA>       <NA>   <NA>        NA      <NA>
21      21     ALL    BM   B-cell <NA>  1/24/1984      M        NA      <NA>
22      22     ALL    BM   B-cell <NA>  5/27/1988      M        NA      <NA>
23      23     ALL    BM   T-cell <NA>   7/9/1991      M        NA      <NA>
24      24     ALL    BM   B-cell <NA>  5/19/1981      M        NA      <NA>
25      25     ALL    BM   B-cell <NA>  2/18/1982      M        NA      <NA>
26      26     ALL    BM   B-cell <NA>       <NA>      F        NA      <NA>
27      27     ALL    BM   B-cell <NA>       <NA>      F        NA      <NA>
34      34     AML    BM     <NA>   M2       <NA>   <NA>        77   Success
35      35     AML    BM     <NA>   M1       <NA>   <NA>        67   Success
36      36     AML    BM     <NA>   M5       <NA>   <NA>        76   Success
37      37     AML    BM     <NA>   M2       <NA>   <NA>        44   Success
38      38     AML    BM     <NA>   M1       <NA>   <NA>        80   Success
28      28     AML    BM     <NA>   M2       <NA>   <NA>        79   Failure
29      29     AML    BM     <NA>   M2       <NA>   <NA>        34   Failure
30      30     AML    BM     <NA>   M5       <NA>   <NA>        93   Failure
31      31     AML    BM     <NA>   M4       <NA>   <NA>        77   Failure
32      32     AML    BM     <NA>   M1       <NA>   <NA>        86   Failure
33      33     AML    BM     <NA>   M2       <NA>   <NA>        70   Failure
     PS Source
1  1.00   DFCI
2  0.41   DFCI
3  0.87   DFCI
4  0.91   DFCI
```

```
5  0.89   DFCI
6  0.76   DFCI
7  0.78   DFCI
8  0.77   DFCI
9  0.89   DFCI
10 0.56   DFCI
11 0.74   DFCI
12 0.20   DFCI
13 1.00   DFCI
14 0.73   DFCI
15 0.98   DFCI
16 0.95   DFCI
17 0.49   DFCI
18 0.59   DFCI
19 0.80   DFCI
20 0.90   DFCI
21 0.76   DFCI
22 0.37   DFCI
23 0.77   DFCI
24 0.92   DFCI
25 0.43   DFCI
26 0.89   DFCI
27 0.82   DFCI
34 0.64  CALGB
35 0.21  CALGB
36 0.94  CALGB
37 0.95  CALGB
38 0.73  CALGB
28 0.44  CALGB
29 0.74  CALGB
30 0.80  CALGB
31 0.61  CALGB
32 0.47  CALGB
33 0.89  CALGB
```

The `$` operator can be used to extract particular variables from an object of class `phenoData`. It also can be used directly on the `exprSet` instance.

```
> table(phenoTrain$ALL.AML)
```

```
ALL AML
 27  11
```

```
> table(golubTest$ALL.AML)
```

```
ALL AML
 20  14
```

Data on only the first 10 genes in the first 3 chips can be obtained using the subsetting operator "["

```
> golubTrain[1:10, 1:3]

Expression Set (exprSet) with
        10 genes
        3 samples
                 phenoData object with 11 variables and 3 cases
         varLabels
                Samples: Samples
                ALL.AML: ALL.AML
                BM.PB: BM.PB
                T.B.cell: T.B.cell
                FAB: FAB
                Date: Date
                Gender: Gender
                pctBlasts: pctBlasts
                Treatment: Treatment
                PS: PS
                Source: Source
```

Notice that when subsetting, we have arranged it so that the *rows* correspond to genes and the *columns* correspond to samples.

4 Gene filtering package: `genefilter`

The `genefilter` package provides functions for sequentially applying filters to the rows (genes) of a matrix or of an instance of the `exprSet` class. For instance, with the Golub et al. (1999) dataset, the following identifies the genes with absolute intensity greater than 10000 in at least 10 of the 38 arrays in the training set.

```
> fg <- kOverA(k = 10, A = 10000)
> flist <- filterfun(fg)
> ans <- genefilter(golubTrain, flist)
> sum(ans)

[1] 131
```