STAT 5570 / 6570: Statistical Bioinformatics
Notes 1.3: Getting Started with Bioconductor

1 Bioconductor Packages

There are many "automatic" functions that come with R; however, most of what we'll use in this class will require specialized functions dealing with genomic and sequence data. These functions are included in separate "packages" that must be downloaded and installed. Often these packages include data sets that we will use in this course.

With R installed, you will need to get the packages necessary to do certain analyses. You can get them one at a time as needed during the course. (This is why you'd want to use a rewritable CD if you're going that route for installing R, as suggested in Notes 1.2.) For example, suppose you know you need the R package called `affy` (either because you know what it does or because you tried to do something in R and R stopped and told you it needed that package). There are many ways to get this package, and the most straightforward is to use the drop-down menus in R. Using the drop-down menus, select `Packages` → `Select repositories...`, and highlight all repositories (for now; this tells R where to look for packages online). Next, using the drop-down menus, select `Packages` → `Install package(s)...`, and after choosing one of the USA mirrors, highlight `affy`.

Alternatively, run the following code:

```
> setRepositories(ind=c(1:9))  # select desired repositories
> chooseCRANmirror(graphics=F) # select a USA mirror, like 75 - USA (CA 1)
> install.packages("affy") # request package
```

As another alternative, if you know for sure that it's a Bioconductor package that you want to install, you can use the following code:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("affy")
```

The `source` command runs the code in the specified file (local or on the web) in the current R session. The file here defines the `biocLite` function, which will download (from the Seattle mirror) the specified package. This will only work if the package is hosted by the Bioconductor repository.

Once a package has been downloaded and installed, it will only be used in an R session if you explicitly request that it be loaded (to save on memory space).

```
> library(affy)
```

2 Bioconductor tools for CEL files

(See p. 10 of Notes 1.1 regarding bioinformatic technologies. We're focusing on recurring statistical issues, not a specific technology.)

Each spot in the image of the microarray corresponds to a probe (either PM or MM), and the "average" pixel intensity of each spot is summarized in a .CEL file. Each .CEL file corresponds to one microarray (i.e., one sample).

To see which *CEL* files are available in a directory:

```
> library(affy)
> cels <- list.celfiles("C:\\folder")
> cels
```

```
[1] "apoEdef1.CEL" "apoEdef2.CEL" "apoEdef3.CEL" "wt1.CEL"      "wt2.CEL"
[6] "wt3.CEL"
```

These .CEL files come from a study of aorta tissue samples in mice. Two strains were considered: wild type and apolipoprotein E deficient.

To read in all *CEL* files in a folder:

```
> data <- ReadAffy(celfile.path="C:\\folder")
> data
```

```
AffyBatch object
size of arrays=1002x1002 features (11 kb)
cdf=Mouse430_2 (45101 affyids)
number of samples=6
number of genes=45101
annotation=mouse4302
notes=
```

This creates an *AffyBatch* object in R. See what this class of object is:

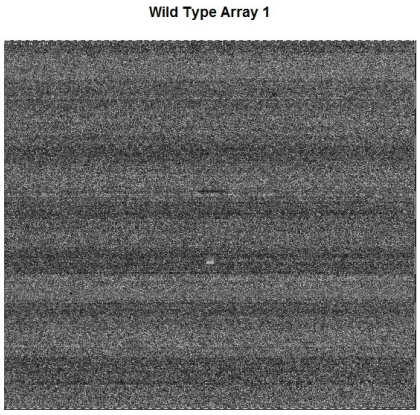
```
> ?AffyBatch
```

To select a specific array from the *AffyBatch* object:

```
> wt1 <- data[,4]
```

To see the image of an array:

```
> image(data[,4], main="Wild Type Array 1")
```



We can access the perfect match intensities of the arrays:

```
> pm.matrix <- pm(data)
> pm.vector <- pm(data[,1])
> head(pm.matrix)
```

	apoEdef1.CEL	apoEdef2.CEL	apoEdef3.CEL	wt1.CEL	wt2.CEL	wt3.CEL
754776	1334	2401	2197	1997	1555	1807
558601	2607	3160	3220	3248	2487	3028
604987	3216	5074	4456	5067	4022	4520
901278	222	470	358	359	241	304
376404	979	1449	1646	1351	956	1213
476021	1169	1639	1493	1664	1206	1347

```
> head(pm.vector)
```

	apoEdef1.CEL
754776	1334
558601	2607
604987	3216
901278	222
376404	979
476021	1169

Then `pm.matrix` is a matrix and `pm.vector` is a vector containing perfect match (PM) intensities. (Use mm function similarly to get mismatch intensities.)

How do we know which intensities correspond to which genes (or probesets)? Probe information is available in *CDF* files. R will automatically download and install the necessary packages when it encounters a *CEL* file. (Alternatively, you can download and install the packages yourself, by hand.) You can also create your own *CDF* file for custom arrays.

To look at gene names (probe set names):

```
> gn <- geneNames(data)
> head(gn)
```

```
[1] "1415670_at"   "1415671_at"   "1415672_at"   "1415673_at"   "1415674_a_at"
[6] "1415675_at"
```

To look at intensities for specific probe set(s):

```
> pm.gn1 <- pm(data,gn[1])
> mm.gn3 <- mm(data,gn[3])
> head(mm.gn3)
```

	apoEdef1.CEL	apoEdef2.CEL	apoEdef3.CEL	wt1.CEL	wt2.CEL	wt3.CEL
1415672_at1	1707	2616	2212	2550	2319	2449
1415672_at2	689	1184	1033	1009	574	757
1415672_at3	217	216	354	398	288	268
1415672_at4	312	454	435	445	439	324
1415672_at5	1356	2086	2262	2214	1372	1788
1415672_at6	1398	2295	2291	2201	1365	1676

3 Preprocessing CEL files

"Preprocessing" refers to the steps necessary to go from this raw (original probe-level) data to usable gene-level data in each sample. Notes 1.4 discuss the relevant issues and a commonly-used procedure.