Lab 2: Bioinformatics (annotation package)

June 4, 2003

Introduction

In this lab you will be introduced to some of the functionality provided by the annotate package and by the different meta-data packages that are produced through the Bioconductor project.

While the experimental data are very large, as we have noted in the lectures the biological meta-data are much larger. The biological meta-data are constantly evolving and improving as more experiments are done, reported and verified. Since most analyses are concentrated on data from a single chip we have decided to package the meta-data in per chip or per instrument data packages. Many of the more popular Affymetrix chips are prepackaged and available from the Bioconductor website. For cDNA or custom arrays the software tools provided by *AnnBuilder* can be used to create custom data packages. This should only take a few hours and need only be done once or twice a year (depending on changes/improvements to the data resources that we rely on). Note, the Bioconductor project does not generate or validate any biological meta-data. We simply package it into what we believe are useful objects that can be used to enhance data analysis. The reader is cautioned to check all mappings and encouraged to report any failings to us.

Annotation

Now we will look at how to map from Affymetrix identifiers to the different biological meta-data available from the Bioconductor Web site. Other data can be obtained from other sources and used in a similar fashion.

For each chip a separate data package is available. In this vignette we explore the *hu6800* data package. Each specific mapping is contained in an *environment*. For our purposes an environment is simply a *hash* table. It provides a very fast way of looking up values for specific text keys. Depending on which way you would like to map the keys are either Affymetrix identifiers (for example `hu6800CHR` maps from Affymetrix identifiers to the chromosome the gene is located on) or from a specific identifier to the Affymetrix, or probe, identifiers. For example, `hu6800GO2PROBE` maps from Gene Ontology numbers

to the probes that are annotated at that GO term. We will describe GO and its uses more in a subsequent section.

Load the data and obtain the Affymetrix identifiers.

```
> library(annotate)
> library(hu6800)
> ls(pos = 2)

 [1] "hu6800"              "hu6800ACCNUM"        "hu6800CHR"
 [4] "hu6800CHRLENGTHS"    "hu6800CHRLOC"        "hu6800ENZYME"
 [7] "hu6800ENZYME2PROBE" "hu6800GENENAME"      "hu6800GO"
[10] "hu6800GO2ALLPROBES" "hu6800GO2PROBE"      "hu6800GRIF"
[13] "hu6800LOCUSID"       "hu6800MAP"           "hu6800ORGANISM"
[16] "hu6800PATH"          "hu6800PATH2PROBE"    "hu6800PMID"
[19] "hu6800PMID2PROBE"    "hu6800QC"            "hu6800SUMFUNC"
[22] "hu6800SYMBOL"        "hu6800UNIGENE"

> affynames <- ls(env = hu6800CHR)
> length(affynames)

[1] 7129

> hu6800()

Quality control information for  hu6800
Date built:  Thu May 15 09:23:32 2003
Number of probes: 7129
Probe number missmatch: None
Probe missmatch: None
Mappings found for probe based rda files:
         hu6800ACCNUM found 7092 of 7129
         hu6800CHRLOC found 6013 of 7129
         hu6800CHR found 6321 of 7129
         hu6800ENZYME found 1001 of 7129
         hu6800GENENAME found 6340 of 7129
         hu6800GO found 5760 of 7129
         hu6800GRIF found 3475 of 7129
         hu6800LOCUSID found 6343 of 7129
         hu6800MAP found 6315 of 7129
         hu6800PATH found 1351 of 7129
         hu6800PMID found 6294 of 7129
         hu6800SUMFUNC found 373 of 7129
         hu6800SYMBOL found 6340 of 7129
```

```
         hu6800UNIGENE found 6274 of 7129
Mappings found for non-probe based rda files:
          hu6800ENZYME2PROBE found 471
         hu6800GO2ALLPROBES found 3878
         hu6800GO2PROBE found 2831
         hu6800PATH2PROBE found 93
         hu6800PMID2PROBE found 33410
```

After loading the data we see that there are a number of different data sets that have been loaded. These provide the many different mappings from Affymetrix identifiers to other biological data, such as chromosomal location which is found in `hu6800CHR`.

We see that there are 7129 identifiers (which matches the `exprSets` we have.

Finally, quality control data and counts etc. of the mappings found is available by calling the function `hu6800()` and by examining the help page, `?hu6800`. These data provide you with information on when the meta-data package was constructed, what data sources were used and how many different mappings were found.

```
> mygene <- affynames[4001]
> get(mygene, env = hu6800ACCNUM)

[1] "U18237"

> get(mygene, env = hu6800LOCUSID)

[1] 23456

> get(mygene, env = hu6800SYMBOL)

[1] "ABCB10"

> get(mygene, env = hu6800GENENAME)

[1] "ATP-binding cassette, sub-family B (MDR/TAP), member 10"

> get(mygene, env = hu6800SUMFUNC)

[1] NA

> get(mygene, env = hu6800UNIGENE)

[1] "Hs.1710"

> get(mygene, env = hu6800CHR)

[1] "1"
```

```
> get(mygene, env = hu6800CHRLOC)

         1
-226051723

> get(mygene, env = hu6800MAP)

[1] "1q42"

> get(mygene, env = hu6800PMID)

[1] "10922475" "10748049" "7766993"

> get(mygene, env = hu6800GO)

         IEA          NAS          NAS           ND          NAS           ND
"GO:0000166" "GO:0005524" "GO:0004009" "GO:0005554" "GO:0006810" "GO:0000004"
         NAS          IEA           ND          IEA
"GO:0005739" "GO:0016021" "GO:0008372" "GO:0019866"

> multiget(affynames[1:10], env = hu6800PMID)

$"A28102_at"
[1] NA

$"AB000114_at"
[1] "12477932"

$"AB000115_at"
[1] NA

$"AB000220_at"
[1] "12174914" "9405678"  "9168980"

$"AB000381_s_at"
[1] "9169150" "8934543"

$"AB000409_at"
[1] "10859165" "9155018"

$"AB000410_s_at"
 [1] "12578369" "12244119" "12189194" "12164330" "12119232" "12117782"
 [7] "12034821" "11992556" "11927502" "11902834" "11837743" "11827746"
[13] "10449904" "10233168" "9681819"  "9348312"  "9321410"  "9223306"
```

```
[19] "9223305"  "9207108"  "9197244"  "9190902"  "9187114"

$"AB000449_at"
[1] "9344656"

$"AB000450_at"
[1] "9344656"

$"AB000460_at"
[1] "9734812"
```

We have arbitrarily selected the probe set with Affymetrix identifier `"U18237_at"`. And then obtained all the mappings for it. A variety of information about this gene can now easily be obtained.

In some cases we need to obtain data on several genes at once. To do that we wrote a special function called `multiget`. Notice that in some cases there are a number of PubMed abstracts associated with the different genes.

We can do some other interesting things as well.

```
> whChrom <- multiget(affynames, env = hu6800CHR)
> table(unlist(whChrom))

  1  10  11  12  13  14  15  16  17  18  19   2  20  21  22   3   4   5   6   7
637 236 376 380  96 212 168 252 403  93 386 406 139  92 161 337 238 276 370 296
  8   9   X   Y
214 238 300  24

> vv <- sapply(whChrom, length)
> table(vv)

vv
   1    2
7120    9

> whChrom[vv == 2]

$"D49410_at"
[1] "X" "Y"

$"HG2868-HT3012_s_at"
[1] "X" "Y"

$"HG3936-HT4206_at"
[1] "X" "Y"
```

```
$"J03592_at"
[1] "X" "Y"

$"L39064_rna1_at"
[1] "X" "Y"

$"M16279_at"
[1] "X" "Y"

$"U11090_at"
[1] "X" "Y"

$"U82668_rna1_at"
[1] "X" "Y"

$"X17648_at"
[1] "X" "Y"
```

So we see the distribution of probe sets according to chromosome. Also note that there are 10 genes that have two chromosomes assigned. We might want to explore these further by examining resources at the NCBI.

Querying PubMed

For more details please look at the short article in R News, Volume 2/2, pages 28-30, by R. Gentleman and J. Gentry, entitled *Querying PubMed*. The National Library of Medicine (NLM) provides a great deal of information on biological data. We have only just started to explore these data. We need further functionality on MeSH and other data that are there; but for now we concentrate on PubMed.

A mapping has been done between LocusLink and PubMed. This associates specific articles with specific genes. These can be queried interactively using tools available in R and other systems. Full text abstracts are readily available, in some cases other full text articles are available (we just don't have the tools to make use of them). We next demonstrate how you could interact with these.

```
> pmids <- get(mygene, env = hu6800PMID)
> if (interactive()) pubmed(pmids, disp = "browser")
> absts2 <- pm.getabst(mygene, "hu6800")

Loading required package: XML

> pm.titles(absts2)
```

```
[[1]]
[1] "M-ABC2, a new human mitochondrial ATP-binding cassette membrane protein."
[2] "Characterization of ABCB9, an ATP binding cassette protein associated with lysosom
[3] "Characterization and mapping of three new mammalian ATP-binding transporter genes
```

```
> sapply(absts2, function(x) pm.abstGrep("[Pp]rotein", x))
```

```
     U18237_at
[1,]      TRUE
[2,]      TRUE
[3,]     FALSE
```

We obtain the abstracts and then search for the word `protein`.

Another source of data is GenBank. We can explore interactions with it in much the same way as with PubMed.

```
> g10 <- affynames[1:10]
> gbacc <- multiget(g10, hu6800ACCNUM)
> if (interactive()) genbank(gbacc, disp = "browser")
> gb <- genbank(gbacc[1], disp = "data")
```

Finally we consider interactions with LocusLink. Here, we use local data (LocusLink does not provide a good interface for automatic querying). We generate HTML output. This is often an easy way to communicate your results with other researchers working on the same analysis.

```
> llid <- multiget(affynames[1:10], hu6800LOCUSID)
> map <- multiget(affynames[1:10], hu6800MAP)
> symb <- multiget(affynames[1:10], hu6800SYMBOL)
> res <- data.frame(affynames[1:10], cbind(unlist(symb), unlist(map)))
> names(res) <- c("Affy ID", "Gene symbol", "Chromosomal location")
> ll.htmlpage(llid, filename = "LocusLink.html", title = "HTML report",
+     othernames = res, table.head = c("LocusID", names(res)),
+     table.center = TRUE, repository = "ll")
```

The last command creates a page markedup using HTML with embedded links. You can use this page to explore the data yourself or you can supply it to your collaborators.

Other Meta-Data

The discussion above considered the use of different NLM and NCBI resources for annotating genomic data. This is largely done by providing data on a per gene basis.

There are many other sources of biologic meta-data. These include KEGG (the Kyoto Encyclopedia of Genes and Genomes) located at `www.kegg.org` and GO (the Gene Ontology Consortium) located at `www.go.org`. Bioconductor produces and releases meta-data packages specific to each of these data resources. The packages are call *KEGG* and *GO* respectively. You are encouraged to explore these (and any other primary sources) for more details and if you find something new we would be happy to work with you to incorporate it into Bioconductor.

As part of our project to develop interactive widgets to help navigate the different data packages we have produced `DPExplorer` which is a tcl/tk widget for exploring the different data packages.

It allows the user to explore the contents of the different data packages and to select items from those packages for us in other settings.

In the next exercise we will locate and select the set of Affymetrix identifiers that are associated with apoptosis. We will use both GO and KEGG to do this. The KEGG mapping provides us with a list of genes that are engaged in the apoptosis pathway while the GO mappings will provide us with the list of genes which have been annotated at the apoptosis term. That basically means that they are believed to play a role in this biological process (but it need not be as direct as being part of the pathway).

```
> library(GO)
> bpcGO <- contents(GOBPID2TERM)
> go.bpc <- bpcGO[grep("^[Aa]poptosis", unlist(bpcGO))]
> go.bpc

$"GO:0006915"

"apoptosis"
```

So we see that the GO label that has been associated with the term *apoptosis* is `GO:0006915`. We can now find all the probes that are annotated at that term using the chip specific data package. Here we use the *hu6800* data package.

```
> affyApop <- get(names(go.bpc), env = hu6800GO2PROBE)
> length(affyApop)

[1] 84

> affyAllApop <- get(names(go.bpc), env = hu6800GO2ALLPROBES)
> length(affyAllApop)

[1] 209
```

These two lists of probe identifiers are quite different. The first is the set of probes (acutally LocusLink identifiers) that have been mapped to specifically to the GO term `GO:0006915`. The second, `affyAllApop` is the set of probe identifiers that have been mapped either to the term `GO:0006915` or to a more specific term. The idea behind GO is that genes associated with a biological process such as apoptosis can either be annotated at that node or at a more specific node.

We now repeat the procedure for KEGG.

```
> library(KEGG)
> apKEGG <- contents(KEGGPATHID2NAME)
> kegg.ap <- apKEGG[grep("^[Aa]poptosis", unlist(apKEGG))]
> kegg.ap

$"04210"
      04210
"Apoptosis"

> affykegg.ap <- get(names(kegg.ap), env = hu6800PATH2PROBE)
> sum(affykegg.ap %in% affyApop)

[1] 10

> sum(affykegg.ap %in% affyAllApop)

[1] 38
```

So we see that all of the probes in the pathway (according to KEGG) are annotated at the GO term `apoptosis`. Which is pretty nice.

What we really do not know how to do well is to incorporate these data into an analysis. We are hoping that by making available the infrastructure that you are using today that you will develop good analytic methods for using these meta-data and will make them available to others.