Annotation in Bioconductor
Genomes, Platforms, Ontologies

Educational Materials

Outline

- Data Sets we will use
- Hypergeometric Testing
- Annotation
- Using Annotation
- Accessing on-line resources

Data Sets

- ALL data: a cohort study of patients with Acute Lymphoblastic Leukemia (Ritz Lab); *ALL*.
- Estrogen data: data from a designed factorial experiment on 8 Affymetrix hgu95av2 chips (Miron Lab); *estrogen*.
- Lymphoma data: data from a cohort study of lymphoma (Alizadeh et al, Nature 2000) on cDNA arrays, 9k spots; *lymphoma*.

ALL Data

- Data from 128 patients with ALL, leukocytes collected pre-treatment
- Many different phenotypes, we will use this data for much of our machine learning
- patients with different translocations behave differently. Some translocations, `t(4;11)` and E2APBX have very strong signatures.

```
Format:

     The different covariates are:

   'cod' The patient IDs.

   'diagnosis' The date of diagnosis.

   'sex' The sex of the patient, coded as 'M' and 'F'.
```

Estrogen

- A designed factorial experiment. There were three different versions of this experiment that were carried out (we will use some combination of the data).
- The purpose of the experiment was to find primary targets of estrogen receptor (see the *factDesign* package and references therein for more details.
- Estrogen at two levels, (present/absent), Cyclohexamide at two levels, Rapamycin at two levels and time at two levels.
- Replicate arrays were made at each level.

Lymphoma

- The original cohort study was much larger ($n = 96$), we have selected 8 arrays as a subset for this study
- These 8 had the same spotting order and there are 4 CLL samples and 4 DLCL.
- We use them as a generic example for cDNA arrays, there is relatively little phenotypic data available so they are not really suitable for machine learning examples.

Hypergeometric Distribution

- The Hypergeometric distribution arises quite often in computations that are made in different bioinformatic settings.
- It is related to (and not as general as using methods for two/multi-way tables or generalized linear models).
- There are many ways to describe this distribution, we use the one that aligns with the software in R (`phyper`).
- The distribution arises from sampling a fixed population.
- There is a multivariate version and a non-central version.

Hypergeometric Distribution

- Suppose that we have an urn with N balls in it; of these m are **white** and n are *black*.
- Then k balls are drawn from the urn and of these x are observed to be white.
- We often want to ask whether there are too many white balls in the sample.
- We can also represent this as a two-way table. One set of margins are determined by m and n. The other by k and $N - k$ and x is then the entry in the (m, k) cell of the table.
- In this case the Hypergeometric test is identical to Fisher's exact test.

Hypergeometric Examples

Specific examples of Hypergeometric calculations.

- Given a list of genes selected in some fashion we might want to ask if those on a particular chromosome are overrepresented.
- Given a list of genes selected in some fashion we might want to ask if those in a specific pathway are overrepresented.
- Given a list of genes selected in some fashion we might want to ask if those involved in specific GO categories are overrepresented.

We will explore some of these questions in the tutorials.

Annotation Overview

- Strategic view of annotation requirements for interactive statistical genomics
- Standards overview
- Tools for pathways, GO, homologene, pubmed-HTML
- Detailed looks at co-citation and GO.

Annotation in Bioinformatics

- two types of biomaterial: $tissue$ under study and $probe$ (possibly artificial) used to assay
  - tissue annotation: organism, organ, phenotype
  - probe annotation: sequence, what genomic component the probe is intended to represent
- for *tissue* annotation there are two basic standards that have been proposed MIAME and MAGE-OM
- for *probe* annotation - we are more general and think not just of the probes used, but rather of the broad biological context

MIAME

- schema, a limited vocabulary for fields

```
> getSlots("MIAME")
```

```
          name            lab        contact
   "character"    "character"    "character"
         title       abstract            url
   "character"    "character"    "character"
       samples hybridizations   normControls
        "list"         "list"         "list"
 preprocessing          other
        "list"         "list"
```

- note that there are no constraints on how the fields are ‘filled out’
- MAGE-OM defines more fields and more constraints (www.mged.org)

MAGE-OM/RMAGEML

- use Protege to look at MAGE-OM
- MAGE-ML is an XML serialization of data using tags derived from MAGE-OM
- EBI arrayExpress and other repositories use the MAGE-ML format to store (but may not have strong requirements on use of fields)
- RMAGEML can read many of these documents and import to R marrayRaw objects

Meta-data General Strategies

- Local curation and preprocessing.
  - make use of multiple data sources in a coherent fashion
  - link in local data
  - can be out-of-date
  - versioned
- On-line access
  - up-to date
  - constantly changing
  - not available all of the time to all collaborators

Local Curation

- this is largely the approach taken by the Bioconductor Project, to date
- we will begin this year to support a larger database backend
- version numbers are essential
- data can be stored with a known structure

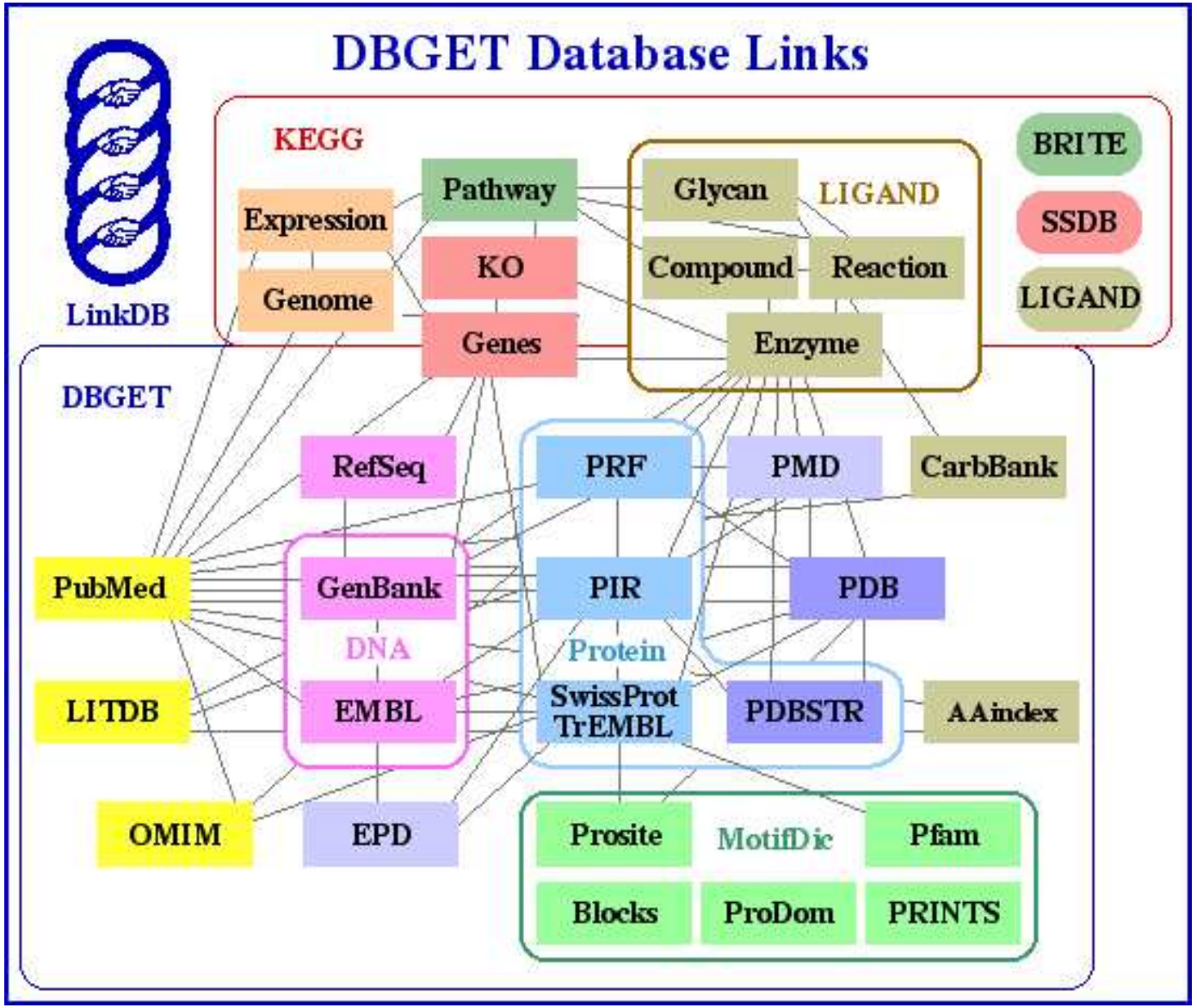
Online Meta-data

- most up to date - but can change from day to day
- must know of the appropriate accessing mechanism
- rely on the provider understanding the difference between a *web service* and a *web site*.

Meta-data resources

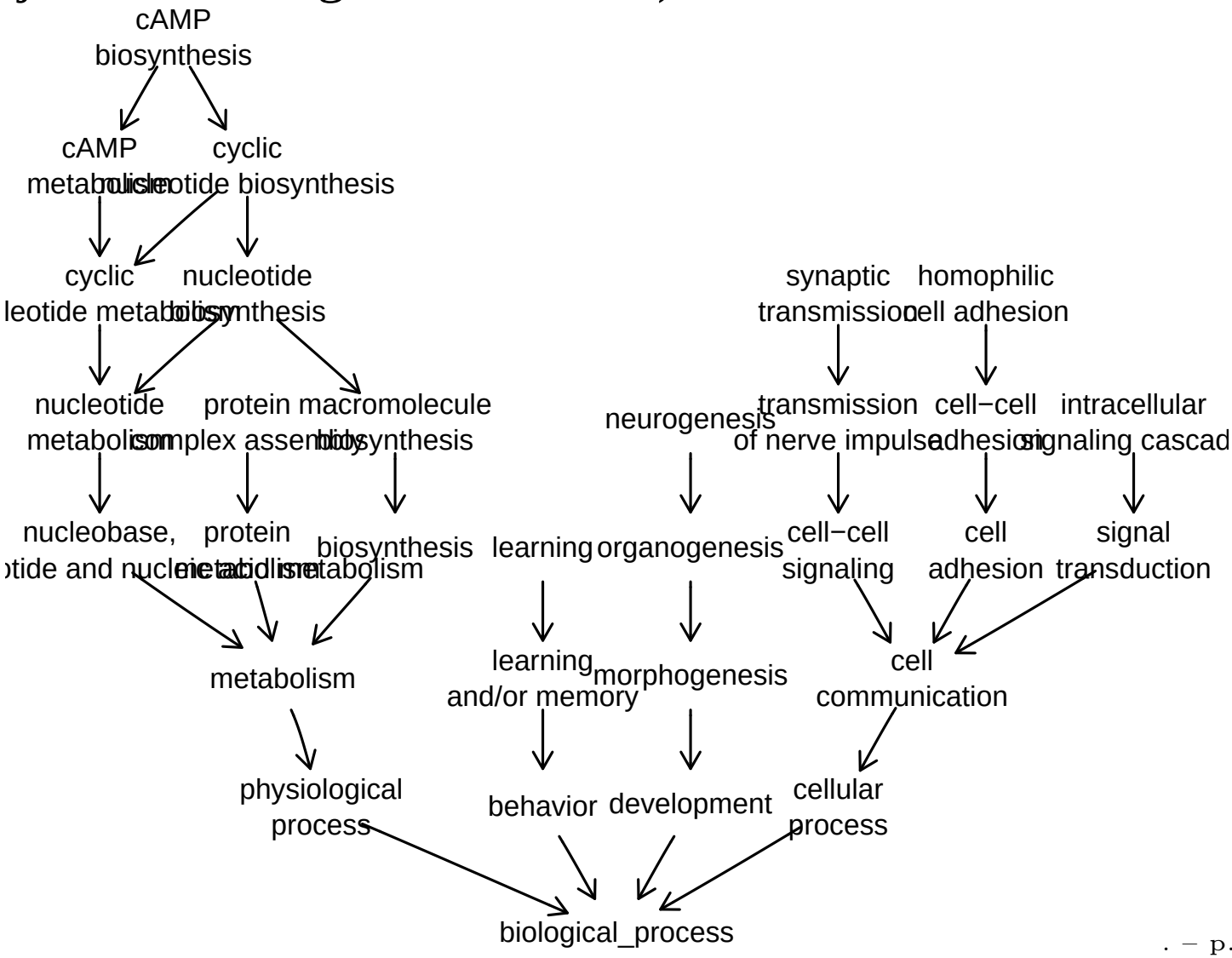
- They are many and varied
- Bioconductor mainly relies on materials from the NCBI as the basis for beginning to create our local curations.
- The main tool for constructing annotation is *AnnBuilder*.
- The main output is a collection of R packages
- Look for a database output in the near future (since the data sets are getting too large and their interactions too complicated).

A slice of annotation resources: KEGG DBGET



Another key resource: Gene Ontology (GO)

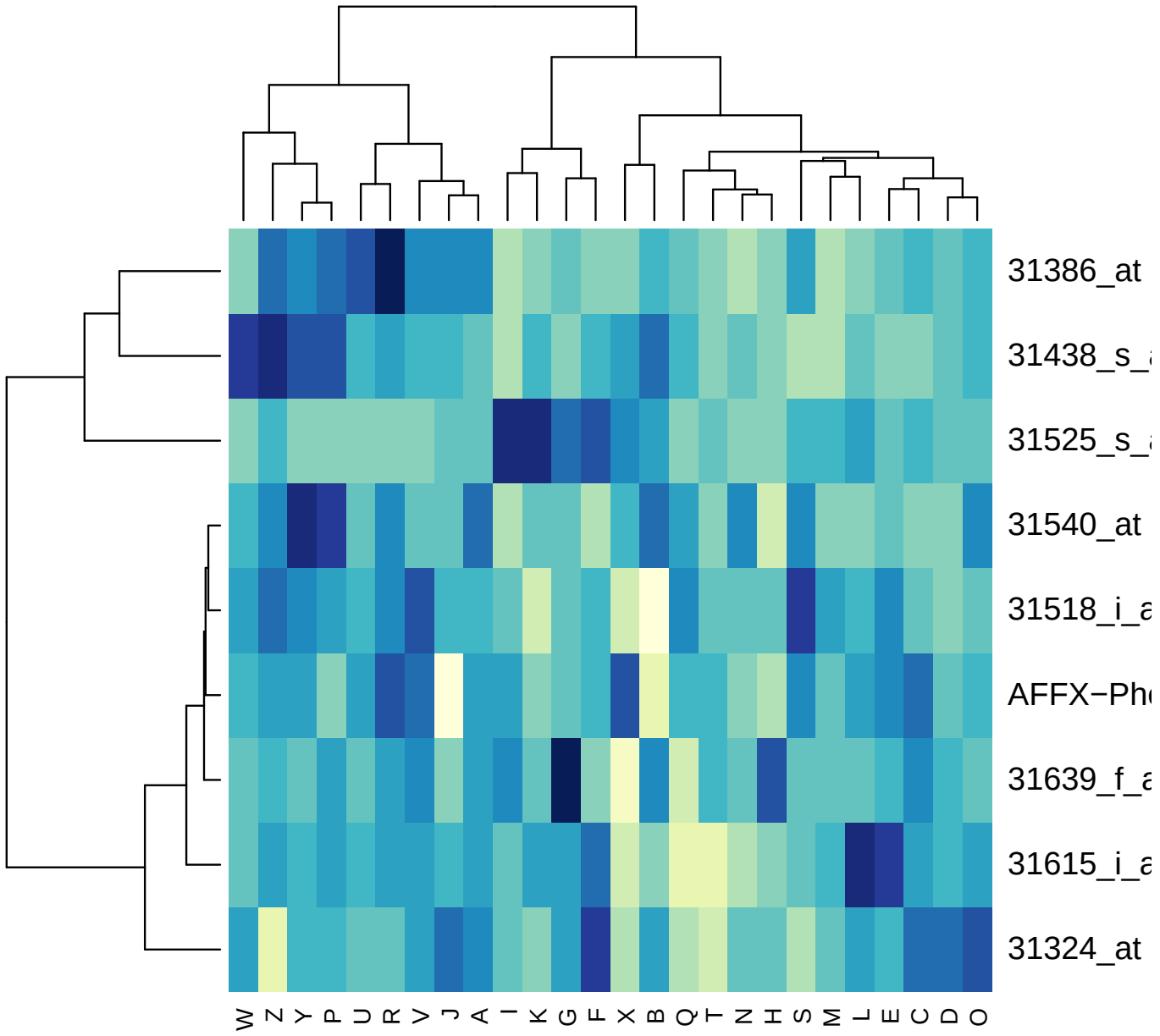
- GO is a very large directed acyclic terminology graph (most arcs join terms to generalizations)



Annotation in the workflow

- typical oligo situation: modest number of microarrays from a fixed platform (e.g., hgu133a or mgu74av2)
- preprocessing and normalization are typically carried out without regard to annotation (some changes are in store)
- for a *gene* we have a vector of expression values across samples
- for a *sample* (array) we have a vector of expression values across genes
- the *interpretation* of a given expression value is derived from annotation on the corresponding tissue-gene relationship

Annotation example: hgu95av2



Data package structure

```
[1] "hgu95av2"          "hgu95av2ACCNUM"
[3] "hgu95av2CHR"       "..."
```

- for platform P, there are environments named P[suff] for 20 or more suffixes
- simple resources: suff=ORGANISM (string) or QC (function)
- chromosomal data: CHR, CHRLOC, CHRLENGTHS (numeric), MAP (cytoband)

Data package structure (2)

- identification: suff=SYMBOL, GENENAME, ACCNUM, LOCUSID, UNIGENE, SUMFUNC (locuslink-based)
- literature links: suff=PMID, GRIF, PMID2PROBE
- external databases: suff=HGID, ENZYME(2PROBE), PATH(2PROBE)
- gene ontology: GO, GO2PROBE, GO2ALLPROBES

What is 1001_at?

- Bioconductor provides packages that collect annotation for many widely used oligo platforms
- *hgu95av2* is one of them

```
> objects("package:hgu95av2")

 [1] "hgu95av2"              "hgu95av2ACCNUM"
 [3] "hgu95av2CHR"           "hgu95av2CHRLENGTHS"
 [5] "hgu95av2CHRLOC"        "hgu95av2ENZYME"
 [7] "hgu95av2ENZYME2PROBE" "hgu95av2GENENAME"
 [9] "hgu95av2GO"            "hgu95av2GO2ALLPROBES"
[11] "hgu95av2GO2PROBE"      "hgu95av2GRIF"
[13] "hgu95av2HGID"          "hgu95av2LOCUSID"
[15] "hgu95av2MAP"           "hgu95av2NM"
[17] "hgu95av2NP"            "hgu95av2OMIM"
[19] "hgu95av2ORGANISM"      "hgu95av2PATH"
[21] "hgu95av2PATH2PROBE"    "hgu95av2PMID"
[23] "hgu95av2PMID2PROBE"    "hgu95av2QC"
[25] "hgu95av2SUMFUNC"       "hgu95av2SYMBOL"
[27] "hgu95av2UNIGENE"
```

Basic annotation

- use either the $ operator or the [[operator
- you can also use `get` or to get several items at once use `mget`

```
> get("1001_at", env = hgu95av2SYMBOL)

[1] "TIE"

> hgu95av2GENENAME$"1001_at"

[1] " tyrosine kinase with immunoglobulin and epidermal growth factor homo

> hgu95av2UNIGENE[["1001_at"]]

[1] "Hs.78824"

> mget(c("1001_at", "1861_at", "1018_at"),
+     hgu95av2CHR)

$"1001_at"
[1] "1"


$"1861_at"
```

GO and 1001_at

- `get`, `$` and `[[` can return complex objects

```
> dTIEtags <- hgu95av2GO$"1001_at"
> print(dtn <- names(dTIEtags))

[1] "GO:0004714" "GO:0005524" "GO:0004872"
[4] "GO:0007498" "GO:0006468" "GO:0007165"
[7] "GO:0005887" "GO:0016740"

> dTIEtags[[1]]

$GOID
[1] "GO:0004714"


$Evidence
[1] "TAS"


$Ontology
[1] "MF"
```

Gene Ontology tag resolution

```
> unlist(mget(dtn, env = GOTERM))[1:5]

                                                    GO:0004714.MF
"transmembrane receptor protein tyrosine kinase activity"
                                                    GO:0005524.MF
                                                    "ATP binding"
                                                    GO:0004872.MF
                                                "receptor activity"
                                                    GO:0007498.BP
                                             "mesoderm development"
                                                    GO:0006468.BP
                                  "protein amino acid phosphorylation"
```

- EBI GOA (gene ontology annotation) initiative contributes maps from loci to terms
- qualifiers: subontologies (BP, CC, MF), evidence codes

Helper Functions

help on `getGO` in *annotate* gives information on all the helper functions such as `getSYMBOL`,

```
> library(annotate)
> getGO("1001_at", "hgu95av2")[[1]]

$GOID
[1] "GO:0004714"

$Evidence
[1] "TAS"

$Ontology
[1] "MF"
```

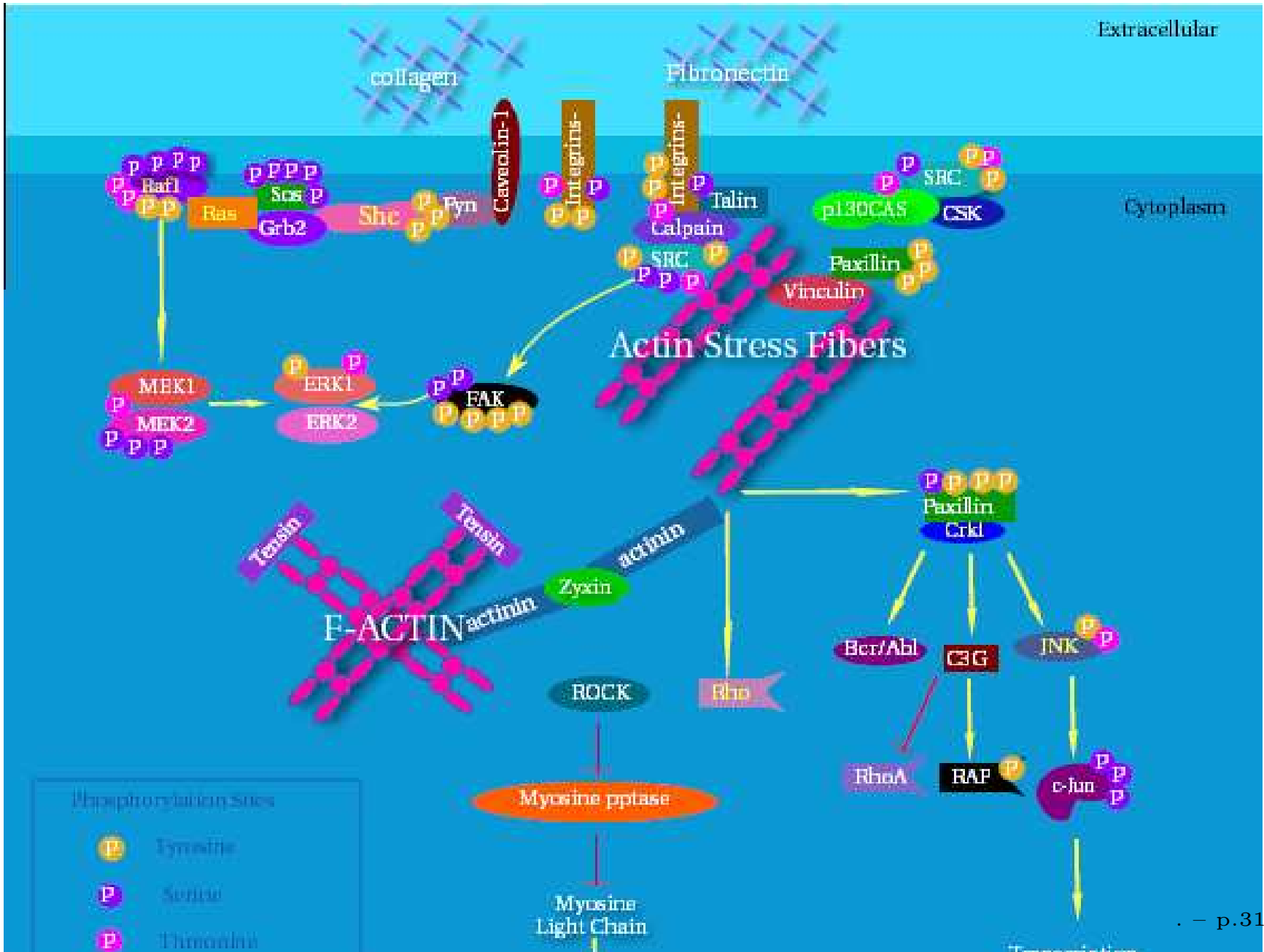
Summary to this point

- various annotation requirements
- solutions in Bioconductor:
  - array data objects (marrayRaw, marrayNorm, affyBatch, exprSet) bind the probe-identifying and tissue-identifying data to expression measures
  - maps from probe ids to standard nomenclature and other metadata are provided as R environments in packages
    - platform-oriented packages (hgu95av2, hgu133a, ...)
    - annotation-oriented packages (GO, KEGG, homology)
    - organism-oriented packages (humanLLMappings, YEAST)

Beyond Lookup

- resolving tags about genes or probes is only one aspect of metadata manipulation of interest
- other tasks
  - isolating groups of probes with similar functions or pathway occupancies
  - building cross-species datasets on the basis of homology
  - testing for functional enrichment; reporting on discovered phenomena

Pathway concepts: integrin



Pathway concepts

- image is from an NCI-based SVG pathway image repository
- nice to look at (perhaps) but little to compute with in this particular resource
- key is remote: colors and shapes of glyphs not locally defined
- multiple object types
- R packages *graph*, *RBGL*, and *Rgraphviz* provide infrastructure for constructing pathway tools.

KEGG environments

```
> KEGG()

Quality control information for  KEGG
Date built:  Thu Mar  4 19:59:33 2004
Mappings found for non-probe based rda files:
          KEGGENZYMEID2GO found 3378
         KEGGEXTID2PATHID found 5176
         KEGGGO2ENZYMEID found 3379
         KEGGPATHID2EXTID found 450
         KEGGPATHID2NAME found 210
         KEGGPATHNAME2ID found 210
```

- KEGGPATHNAME2ID will map from names to numeric tage
- `ls` on the environment will tell us all the names
- or use `DPExplorer` from the *tkWidgets* package to examine the elements.
- or write your own special querying tools using `grep`

KEGG pathway names

```
> sort(ls(KEGGPATHNAME2ID))[1:16]

 [1] "1,1,1-Trichloro-2,2-bis(4-chlorophenyl)ethane (DDT) degradation"
 [2] "1,2-Dichloroethane degradation"
 [3] "1,4-Dichlorobenzene degradation"
 [4] "2,4-Dichlorobenzoate degradation"
 [5] "3-Chloroacrylic acid degradation"
 [6] "ABC transporters, ABC-2 and other types"
 [7] "ABC transporters, eukaryotic"
 [8] "ABC transporters, prokaryotic"
 [9] "ATP synthesis"
[10] "ATPases"
[11] "Alanine and aspartate metabolism"
[12] "Alkaloid biosynthesis I"
[13] "Alkaloid biosynthesis II"
[14] "Alzheimer's disease"
[15] "Amino Acid Metabolism"
[16] "Aminoacyl-tRNA biosynthesis"
```

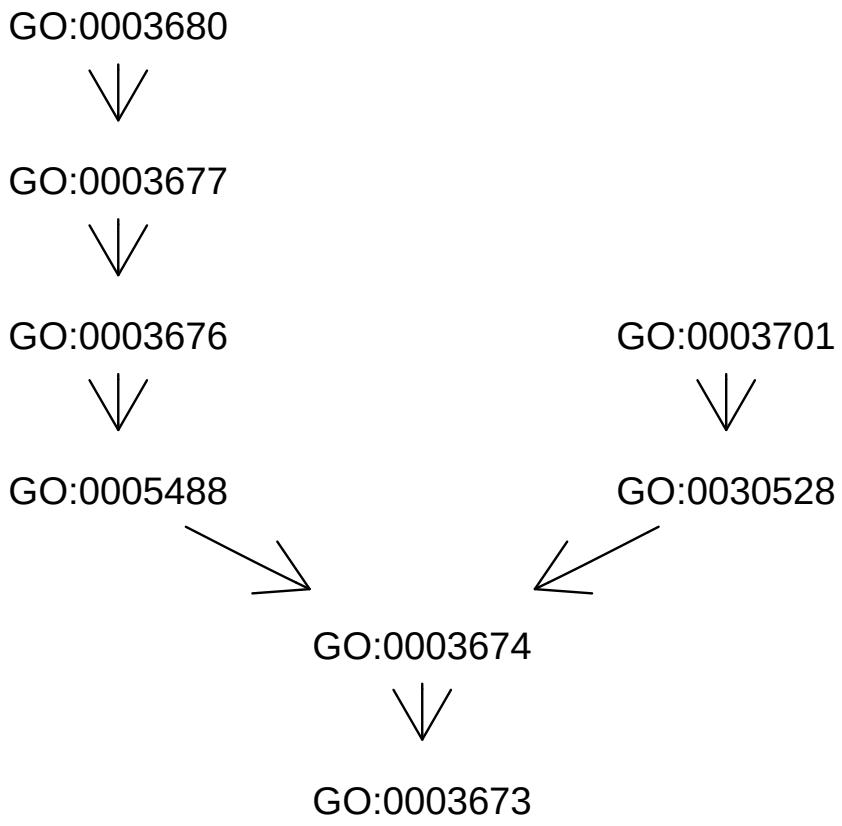
KEGG pathway search

```
> knames <- ls(KEGGPATHNAME2ID)
> ii <- grep("ntegrin", knames)
> knames[ii]

[1] "Integrin-mediated cell adhesion"

> get(knames[ii], KEGGPATHNAME2ID)

[1] "04510"
```

Quiz: given this code, how do we determine what probe sets on hgu133a have been associated

with this pathway?

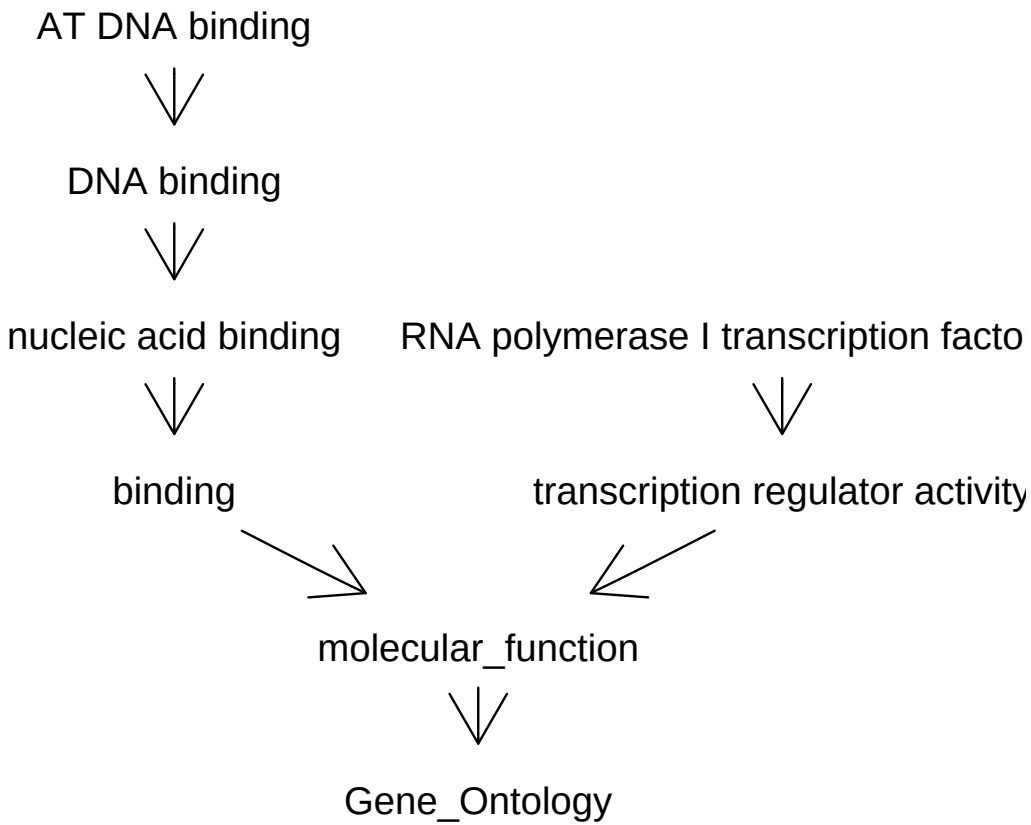
GO: Gene Ontology



GO: Gene Ontology

```
> nnn <- nodes(cg3)
> nnn

[1] "GO:0003680" "GO:0003677" "GO:0003676"
[4] "GO:0005488" "GO:0003674" "GO:0003673"
[7] "GO:0003701" "GO:0030528"

> nn <- unlist(mget(nnn, GOTERM))
> nn

                                  GO:0003680.MF
                               "AT DNA binding"
                                  GO:0003677.MF
                                  "DNA binding"
                                  GO:0003676.MF
                            "nucleic acid binding"
                                  GO:0005488.MF
                                      "binding"
                                  GO:0003674.MF
                             "molecular_function"
                               GO:0003673.GO root
                                  "Gene_Ontology"
                                  GO:0003701.MF
```

Better plot


AT DNA binding
DNA binding
nucleic acid binding
RNA polymerase I transcription facto
binding
transcription regulator activity
molecular_function
Gene_Ontology

Inferential use

- gene selection is based on expression distribution
- interpretation may be difficult if there are many genes selected and functions are not explicitly known
- prevalent technique: use 2x2 tables formed by GO mappings, example(GOHyperG)

	mapped to T	not mapped to T
sel	a	b
-sel	c	d

NCBI HomoloGene

- The HomoloGene project provides interspecies mapping information.
- The *homologene* is undergoing substantial renovation and enhancement for the next release.
- Mappings based on amino acid similarity tend to be better, but our microarrays are based on RNA. We need some way to make sure that the similarity holds for the RNA probes that were actually used.
- Interspecies issues are becoming important.

Some Species Codes

TaxID	Organism		
3702	Arabidopsis thaliana	9913	Bos taurus
6239	Caenorhabditis elegans	7955	Danio rerio
7227	Drosophila melanogaster	9606	Homo sapiens
4513	Hordeum vulgare	4081	Lycopersicon esculentum
3880	Medicago truncatula	10090	Mus musculus
4530	Oryza sativa	10116	Rattus norvegicus
9823	Sus scrofa	4565	Triticum aestivum
8355	Xenopus laevis	4577	Zea mays

HomoloGene Example

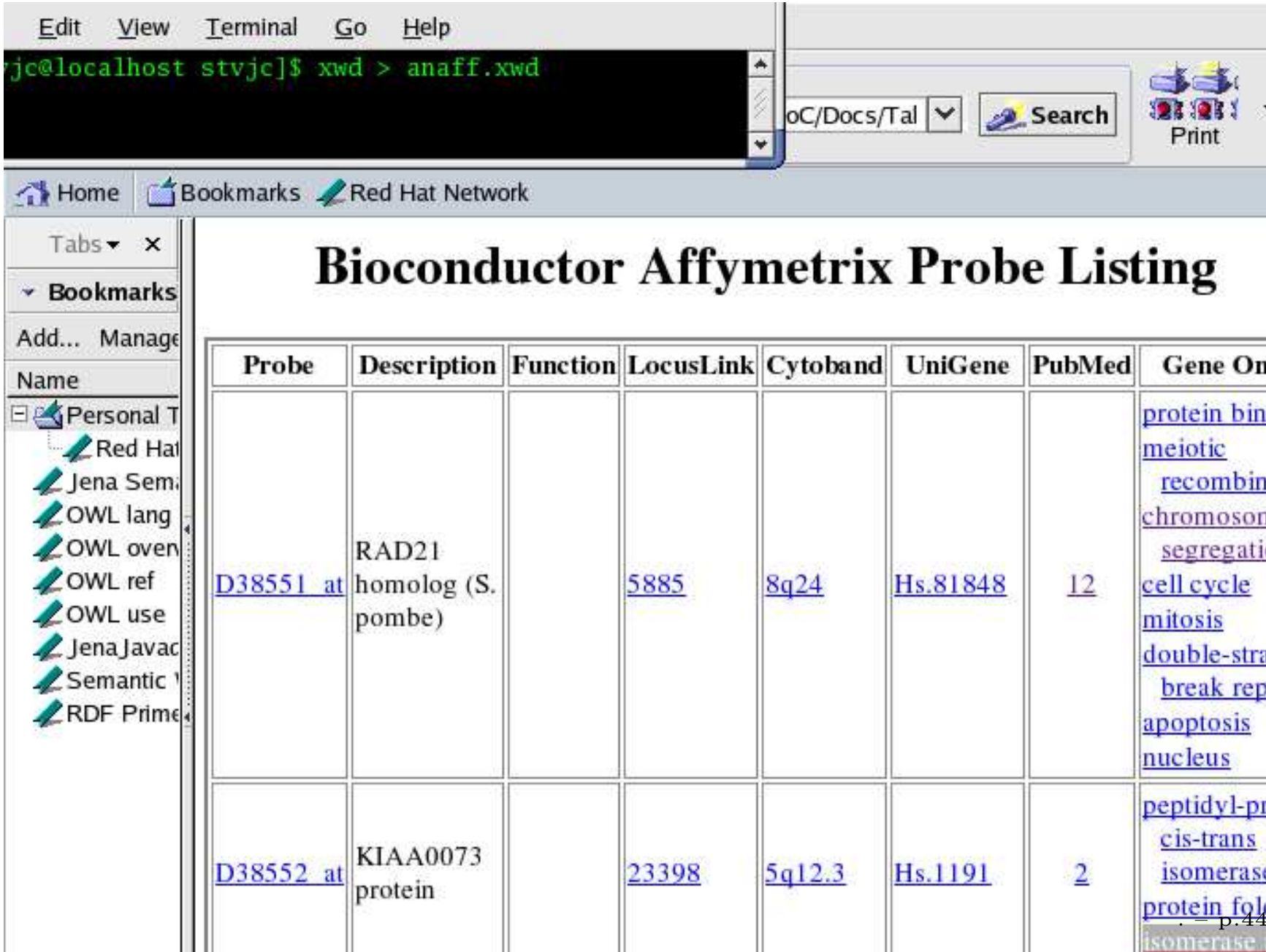
We can obtain information about homologys between different organisms by querying these different hash tables.

```
> homology9606HGID2HGID$"539083"

     Mus musculus Rattus norvegicus
         "229747"           "428809"
             7165               7719
         "603551"           "358626"
   Xenopus laevis         Sus scrofa
          "86003"           "332223"
       Bos taurus
         "585536"

> hg1 = homology9606HGID2PS$"539083"
> hg1[2:5]

10090;192196;B 10116;312251;B      7165;NA;B
       "85.48"        "93.74"        "75.18"
     7719;NA;B
       "74.82"
```

Hyperlinked Output

```
> library(annaffy)
> pnames <- c("34152_at", "32886_at", "40613_at",
+     "31379_at", "35573_r_at")
> atable <- aafTableAnn(pnames, "hgu95av2",
+     aaf.handler())
> saveHTML(atable, file = "gtree.html")
```

Interactive report example



Bioconductor Affymetrix Probe Listing

Probe	Description	Function	LocusLink	Cytoband	UniGene	PubMed	Gene On
D38551_at	RAD21 homolog (S. pombe)		5885	8q24	Hs.81848	12	protein bin meiotic recombin chromoson segregati cell cycle mitosis double-stra break rep apoptosis nucleus
D38552_at	KIAA0073 protein		23398	5q12.3	Hs.1191	2	peptidyl-pr cis-trans isomerase protein fol isomerase

Graphics

- The *geneplotter* package has a number of different graphical methods.
- Particularly `alongChrom`, `cPlot`.
- `heatmap` in R can be modified substantially
- Consider using *RColorBrewer* to select color schemes that appropriately reflect what you are trying to do.

Using On-line resources

- As we noted above one method for dealing with meta-data (or annotation data) is to rely on on-line services.
- In R you can use *connections* to query different resources. Download the response to your query and process it.
- Most processing is done via XML and the *XML* package.
- SOAP is a related protocol that is sometimes supported (KEGG for example) and there is a package *SSOAP* available from the Omegahat Project.

Querying PubMed

- one of our first tools was based on the web services provided by NCBI and it remains a good example.
- much of this material is taken verbatim from the “Querying online Data”, HowTo in the *annotate* package
- the HowTo’s and Vignettes are two of the best sources of how to do things; they should have complete working examples

Some Querying Examples

- `locuslinkQuery` takes a character string used for querying PubMed
- `locuslinkQuery("leukemia", "Hs")` finds all human genes that are textually associated with *leukemia*.
- given a list of interesting genes, we first find their *PMIDs* using the annotation packages; then query PubMed for the available data.

Querying PubMed

We just pick some arbitrary genes here (but you would use a gene list of some interest to you).

```
> affys <- geneNames(eset)[490:500]
> ids <- getPMID(affys, "hgu95av2")
> ids <- unlist(ids, use.names = FALSE)
> ids <- unique(ids[!is.na(as.numeric(ids))])
> ids[1:20]

 [1] "9695952"  "9054383"  "8849451"  "8764062"
 [5] "8764009"  "8680883"  "8121496"  "7933101"
 [9] "7836461"  "7835343"  "7729427"  "12408966"
[13] "8325638"  "8175896"  "1889752"  "12679040"
[17] "12477932" "9315667"  "12590922" "12198562"
```

And then we would execute code like:

```
x <- pubmed(ids)
a <- xmlRoot(x)
numAbst <- length(xmlChildren(a))
numAbst
```

Providing Meaning

We now turn our attention to the last topic in this lecture. That is, using meta-data to provide meaning to the list of genes that we have selected.

- almost all meta-data packages can be used to help answer different questions
- we will consider the questions of GO term over abundance and PubMed affiliations
- the problems are simpler to understand if we use graph theory to describe the concepts involved
- in later lectures we will cover graph theory in more depth

Using GO

- we saw previously that for any GO term we can ask whether that term is over represented in our data.