How to use bimaps from the ".db" annotation packages

Marc Carlson, Hervé Pagès, Seth Falcon, Nianhua Li

November 3, 2023

NOTE The ‘bimap’ interface to annotation resources is not recommend; instead, use the approach in the vignette Introduction To Bioconductor Annotation Packages.

1 Introduction

1.0.1 Purpose

AnnotationDbi is used primarily to create mapping objects that allow easy access from R to underlying annotation databases. As such, it acts as the R interface for all the standard annotation packages. Underlying each AnnotationDbi supported annotation package is at least one (and often two) annotation databases. AnnotationDbi also provides schemas for theses databases. For each supported model organism, a standard gene centric database is maintained from public sources and is packaged up as an appropriate organism or "org" package.

1.0.2 Database Schemas

For developers, a lot of the benefits of having the information loaded into a real database will require some knowledge about the database schema. For this reason the schemas that were used in the creation of each database type are included in AnnotationDbi. The currently supported schemas are listed in the DBschemas directory of AnnotationDbi. But it is also possible to simply print out the schema that a package is currently using by using its "_dbschema" method.

There is one schema/database in each kind of package. These schemas specify which tables and indices will be present for each package of that type. The schema that a particular package is using is also listed when you type the name of the package as a function to obtain quality control information.

The code to make most kinds of the new database packages is also included in AnnotationDbi. Please see the vignette on SQLForge for more details on how to make additional database packages.
Internal schema Design of org packages

The current design of the organism packages is deliberately simple and gene centric. Each table in the database contains a unique kind of information and also an internal identifier called _id. The internal _id has no meaning outside of the context of a single database. But _id does connect all the data within a single database.

As an example if we wanted to connect the values in the genes table with the values in the kegg table, we could simply join the two tables using the internal _id column. It is very important to note however that _id does not have any absolute significance. That is, it has no meaning outside of the context of the database where it is used. It is tempting to think that an _id could have such significance because within a single database, it looks and behaves similarly to an entrez gene ID. But _id is definitely NOT an entrez gene ID. The entrez gene IDs are in another table entirely, and can be connected to using the internal _id just like all the other meaningful information inside these databases. Each organism package is centered around one type of gene identifier. This identifier is found as the gene_id field in the genes table and is both the central ID for the database as well as the foreign key that chip packages should join to.

The chip packages are 'lightweight', and only contain information about the basic probe to gene mapping. You might wonder how such packages can provide access to all the other information that they do. This is possible because all the other data provided by chip packages comes from joins that are performed by AnnotationDbi behind the scenes at run time. All chip packages have a dependency on at least one organism package. The name of the organism package being depended on can be found by looking at its "ORGPKG" value. To learn about the schema from the appropriate organism package, you will need to look at the "_dbschema" method for that package. In the case of the chip packages, the gene_id that in these packages is mapped to the probe_ids, is used as a foreign key to the appropriate organism package.

Specialized packages like the packages for GO and KEGG, will have their own schemas but will also adhere to the use of an internal _id for joins between their tables. As with the organism packages, this _id is not suitable for use as a foreign key.

For a complete listing of the different schemas used by various packages, users can use the `available.dbschemas` function. This list will also tell you which model organisms are supported.

```r
library(DBI)
library(org.Hs.eg.db)

## Loading required package: AnnotationDbi
## Loading required package: stats4
## Loading required package: BiocGenerics
```
How to use bimaps from the ".db" annotation packages

```
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##   IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##   Filter, Find, Map, Position, Reduce, anyDuplicated, aperm,
##   append, as.data.frame, basename, cbind, colnames, dirname,
##   do.call, duplicated, eval, evalq, get, grep, grepl, intersect,
##   is.unsorted, lapply, mapply, match, mget, order, paste, pmax,
##   pmax.int, pmin, pmin.int, rank, rbind, rownames, sapply,
##   setdiff, table, tapply, union, unique, unsplit, which.max,
##   which.min
## Loading required package: Biobase
## Welcome to Bioconductor
## Vignettes contain introductory material; view with
## 'browseVignettes()'. To cite Bioconductor, see
## 'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package: IRanges
## Loading required package: S4Vectors
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:utils':
##   findMatches
## The following objects are masked from 'package:base':
##   I, expand.grid, uname

library(AnnotationForge)
available.dbschemas()
```
How to use bimaps from the ".db" annotation packages

2 Examples

2.0.1 Basic information

The AnnotationDbi package provides an interface to SQLite-based annotation packages. Each SQLite-based annotation package (identified by a ".db" suffix in the package name) contains a number of AnnDbBimap objects in place of the environment objects found in the old-style environment-based annotation packages. The API provided by AnnotationDbi allows you to treat the AnnDbBimap objects like environment instances. For example, the functions [], get, mget, and ls all behave the same as they did with the older environment based annotation packages. In addition, new methods like [, toTable, subset and others provide some additional flexibility in accessing the annotation data.

```r
library(hgu95av2.db)
#
```

The same basic set of objects is provided with the db packages:

```r
ls("package:hgu95av2.db")
## [1] "hgu95av2" "hgu95av2.db"
## [3] "hgu95av2ACCNUM" "hgu95av2ALIAS2PROBE"
## [5] "hgu95av2CHR" "hgu95av2CHRLENGTHS"
## [7] "hgu95av2CHRL0C" "hgu95av2CHRL0CEND"
## [9] "hgu95av2ENSEMBL" "hgu95av2ENSEMBL2PROBE"
## [11] "hgu95av2ENTREZID" "hgu95av2ENZYME"
## [13] "hgu95av2ENZYME2PROBE" "hgu95av2GENENAME"
## [15] "hgu95av2G0" "hgu95av2G02ALLPROBES"
## [17] "hgu95av2G02PROBE" "hgu95av2MAP"
## [19] "hgu95av2MAPCOUNTS" "hgu95av20MIM"
## [21] "hgu95av20RGANISM" "hgu95av20RGPKG"
## [23] "hgu95av2PATH" "hgu95av2PATH2PROBE"
## [25] "hgu95av2PFAM" "hgu95av2PMID"
## [27] "hgu95av2PMID2PROBE" "hgu95av2PROSITE"
## [29] "hgu95av2REFSEQ" "hgu95av2SYMB0L"
## [31] "hgu95av2UNIPROT" "hgu95av2_dbInfo"
## [33] "hgu95av2_dbconn" "hgu95av2_dbfile"
## [35] "hgu95av2_dbschema"
```

Exercise 1
Start an R session and use the library function to load the hgu95av2.db software package. Use search() to see that an organism package was also loaded and then use the appropriate "_dbschema" methods to the schema for the hgu95av2.db and org.Hs.eg.db packages.
How to use bimaps from the ".db" annotation packages

It is possible to call the package name as a function to get some QC information about it.

```r
qdata = capture.output(hgu95av2())
head(qdata, 20)
```

```
## [1] "Quality control information for hgu95av2:
## [2] "
## [3] "
## [4] "This package has the following mappings:
## [5] "
## [6] "hgu95av2ACCCNUM has 12625 mapped keys (of 12625 keys)"
## [7] "hgu95av2ALIAS2PROBE has 37476 mapped keys (of 258270 keys)"
## [8] "hgu95av2CHR has 11683 mapped keys (of 12625 keys)"
## [9] "hgu95av2CHRLENGTHS has 595 mapped keys (of 711 keys)"
## [10] "hgu95av2CHRLOC has 11637 mapped keys (of 12625 keys)"
## [11] "hgu95av2CHRLOCEND has 11637 mapped keys (of 12625 keys)"
## [12] "hgu95av2ENSEMBL has 11683 mapped keys (of 12625 keys)"
## [13] "hgu95av2ENSEMBL2PROBE has 10016 mapped keys (of 40467 keys)"
## [14] "hgu95av2ENTREZID has 11683 mapped keys (of 12625 keys)"
## [15] "hgu95av2ENZYMNAME has 11683 mapped keys (of 12625 keys)"
## [16] "hgu95av2ENZYMNAME2PROBE has 785 mapped keys (of 975 keys)"
## [17] "hgu95av2Ensembl has 11683 mapped keys (of 12625 keys)"
## [18] "hgu95av2GO has 11683 mapped keys (of 12625 keys)"
## [19] "hgu95av2GO2ALLPROBES has 20621 mapped keys (of 22741 keys)"
## [20] "hgu95av2GO2PROBE has 15988 mapped keys (of 18809 keys)"
```

Alternatively, you can get similar information on how many items are in each of the provided maps by looking at the MAPCOUNTs:

```r
hgu95av2MAPCOUNTS
```

To demonstrate the `environment` API, we'll start with a random sample of probe set IDs.

```r
all_probes <- ls(hgu95av2ENTREZID)
length(all_probes)
```

```
## [1] 12625
```

```r
set.seed(0xa1beef)
probes <- sample(all_probes, 5)
probes
```

```
## [1] "39758_f_at" "34055_at" "34045_at" "38348_at" "40584_at"
```

The usual ways of accessing annotation data are also available.
How to use bimaps from the ".db" annotation packages

```r
hgu95av2ENTREZID[[probes[1]]]
## [1] "3916"

hgu95av2ENTREZID$"31882_at"
## [1] "9136"

syms <- unlist(mget(probes, hgu95av2SYMBOL))
syms
## 39758_f_at  34055_at  34045_at  38348_at  40584_at
## "LAMP1"  "ACVR1B"  "CT62"  "ACOX2"  "NUP88"
```

The annotation packages provide a huge variety of information in each package. Some common types of information include gene symbols (SYMBOL), GO terms (GO), KEGG pathway IDs (KEGG), ENSEMBL IDs (ENSEMBL) and chromosome start and stop locations (CHRLOC and CHRLOCEND). Each mapping will have a manual page that you can read to describe the data in the mapping and where it came from.

```r
?hgu95av2CHRLOC
```

**Exercise 2**

*For the probes in `probes` above, use the annotation mappings to find the chromosome start locations.*

### 2.0.2 Manipulating Bimap Objects

Many filtering operations on the annotation Bimap objects require conversion of the AnnDbBimap into a list. In general, converting to lists will not be the most efficient way to filter the annotation data when using a SQLite-based package. Compare the following two examples for how you could get the 1st ten elements of the hgu95av2SYMBOL mapping. In the 1st case we have to get the entire mapping into list form, but in the second case we first subset the mapping object itself and this allows us to only convert the ten elements that we care about.

```r
system.time(as.list(hgu95av2SYMBOL)[1:10])
## vs:

system.time(as.list(hgu95av2SYMBOL[1:10]))
```

There are many different kinds of Bimap objects in AnnotationDbi, but most of them are of class AnnDbBimap. All /RclassBimap objects represent data as a set of left and right keys. The typical usage of these mappings is to search for right keys that match a set of left keys that have been supplied by the user. But sometimes it is also convenient to go in the opposite direction.
How to use bimaps from the ".db" annotation packages

The annotation packages provide many reverse maps as objects in the package namespace for backwards compatibility, but the reverse mappings of almost any map is also available using `revmap`. Since the data are stored as tables, no extra disk space is needed to provide reverse mappings.

```r
unlist(mget(syms, revmap(hgu95av2SYMBOL)))
## LAMP1 ACVR1B1 ACVR1B2 ACVR1B3 ACVR1B4
## "39758_f_at" "34055_at" "34056_g_at" "34415_at" "36451_at"
## ACVR1B5 ACVR1B6 CT62 AC0X2 NUP881
## "39199_at" "921_s_at" "34045_at" "38348_at" "40584_at"
## NUP882
## "40804_at"
```

So now that you know about the `revmap` function you might try something like this:

```r
as.list(revmap(hgu95av2PATH)["00300"])
## $`00300`
## [1] "36132_at" "35870_at"
```

Note that in the case of the PATH map, we don't need to use `revmap(x)` because `hgu95av2.db` already provides the PATH2PROBE map:

```r
x <- hgu95av2PATH
## except for the name, this is exactly revmap(x)
revx <- hgu95av2PATH2PROBE
revx2 <- revmap(x, objName="PATH2PROBE")
revx2
## PATH2PROBE map for chip hgu95av2 (object of class "ProbeAnnDbBimap")
identical(revx, revx2)
## [1] TRUE
as.list(revx["00300"])
## $`00300`
## [1] "36132_at" "35870_at"
```

Note that most maps are reversible with `revmap`, but some (such as the more complex GO mappings), are not. Why is this? Because to reverse a mapping means that there has to be a "value" that will always become the "key" on the newly reversed map. And GO mappings have several distinct possibilities to choose from (GO ID, Evidence code or Ontology). In non-reversible cases like this, AnnotationDbi will usually provide a pre-defined reverse map. That way, you will always know what you are getting when you call `revmap`
How to use bimaps from the ".db" annotation packages

While we are on the subject of GO and GO mappings, there are a series of special methods for GO mappings that can be called to find out details about these IDs. `Term`, `GOID`, `Ontology`, `Definition`, `Synonym`, and `Secondary` are all useful ways of getting additional information about a particular GO ID. For example:

```r
Term("GO:0000018")
```

```r
##
## GO:0000018
## "regulation of DNA recombination"
```

```r
Definition("GO:0000018")
```

```r
##
## GO:0000018
## "Any process that modulates the frequency, rate or extent of DNA recombination, a DNA metabolic process in which a new DNA molecule is formed by reassortment of genes resulting in gene combinations different from those that were present in the parents."
```

Exercise 3

*Given the following set of RefSeq IDs: c("NG_005114","NG_007432","NG_008063"), Find the Entrez Gene IDs that would correspond to those. Then find the GO terms that are associated with those entrez gene IDs.*

org.Hs.eg.db packages.

2.0.3 The Contents and Structure of Bimap Objects

Sometimes you may want to display or subset elements from an individual map. A `Bimap` interface is available to access the data in table (`data.frame`) format using `[]` and `toTable`.

```r
head(toTable(hgu95av2GO[probes]))
```

```r
##
## probe_id go_id Evidence Ontology
## 1 34055_at GO:0000082 IDA BP
## 2 34055_at GO:0001701 IEA BP
## 3 34055_at GO:0001942 IEA BP
## 4 34055_at GO:0006355 IDA BP
## 5 34055_at GO:0006468 IBA BP
## 6 34055_at GO:0006468 IDA BP
```

The `toTable` function will display all of the information in a `Bimap`. This includes both the left and right values along with any other attributes that might be attached to those values. The left and right keys of the `Bimap` can be extracted using `Lkeys` and `Rkeys`. If is is necessary to only display information that is directly associated with the left to right links in a `Bimap`, then the `links` function can be used. The `links` returns a data frame with one row for each link in the bimap that it is applied to. It only reports the left and right keys along with any attributes that are attached to the edge between these two values.
How to use bimaps from the ".db" annotation packages

Note that the order of the cols returned by `toTable` does not depend on the direction of the map. We refer to it as an 'undirected method':

```
# probe_id path_id
# 1 1000_at 04010
# 2 1000_at 04012
# 3 1000_at 04062
# 4 1000_at 04114
# 5 1000_at 04150
# 6 1000_at 04270
```

Notice however that the Lkeys are always on the left (1st col), the Rkeys always in the 2nd col.

For `length()` and `keys()`, the result does depend on the direction, hence we refer to these as 'directed methods':

```
length(x)
## [1] 12625
length(revx)
## [1] 229
```

```
allProbeSetIds <- keys(x)
allKEGGIds <- keys(revx)
```

There are more 'undirected' methods listed below:

```
junk <- Lkeys(x)  # same for all maps in hgu95av2.db (except pseudo-map MAPCOUNTS)
Llength(x)  # nb of Lkeys
## [1] 12625
```

```
junk <- Rkeys(x)  # KEGG ids for PATH/PATH2PROBE maps, GO ids for GO/GO2PROBE/GO2ALLPROBES maps, etc...
Rlength(x)  # nb of Rkeys
```
Notice how they give the same result for x and revmap(x)

You might be tempted to think that Lkeys and Llength will tell you all that you want to know about the left keys. But things are more complex than this, because not all keys are mapped. Often, you will only want to know about the keys that are mapped (ie. the ones that have a corresponding Rkey). To learn this you want to use the mappedkeys or the undirected variants mappedLkeys and mappedRkeys. Similarly, the count.mappedkeys, count.mappedLkeys and count.mappedRkeys methods are very fast ways to determine how many keys are mapped. Accessing keys like this is usually very fast and so it can be a decent strategy to subset the mapping by 1st using the mapped keys that you want to find.

```r
x = hgu95av2ENTREZID[1:10]
## Directed methods
mappedkeys(x) # mapped keys
## [1] "1000_at" "1001_at" "1002_f.at" "1003_s.at" "1004_at"
## [6] "1005_at" "1006_at" "1007_s.at" "1008_f.at" "1009_at"

count.mappedkeys(x) # nb of mapped keys
## [1] 10

## Undirected methods
mappedLkeys(x) # mapped left keys
## [1] "1000_at" "1001_at" "1002_f.at" "1003_s.at" "1004_at"
## [6] "1005_at" "1006_at" "1007_s.at" "1008_f.at" "1009_at"

count.mappedLkeys(x) # nb of mapped Lkeys
## [1] 10
```

If you want to find keys that are not mapped to anything, you might want to use isNA.

```r
y = hgu95av2ENTREZID[isNA(hgu95av2ENTREZID)] # usage like is.na()
Lkeys(y)[1:4]
## [1] "1037_at" "1047_s.at" "1089_i.at" "108_g.at"
```

**Exercise 4**

How many probesets do not have a GO mapping for the hgu95av2.db package? How many have no mapping? Find a probeset that has a GO mapping. Now look at the GO mappings for this probeset in table form.
2.0.4 Some specific examples

Let's use what we have learned to get information about the probes that are not assigned to a chromosome:

```r
x <- hgu95av2CHR
Rkeys(x)
## [1] "19"  "12"  "8"  "14"  "3"  "2"  "17"  "16"  "9"  "X"  "6"  "1"  "7"
## [14] "10"  "11"  "22"  "5"  "18"  "15"  "Y"  "20"  "21"  "4"  "13"  "MT"  "Un"

chroms <- Rkeys(x)[23:24]
chroms
## [1] "4"  "13"

Rkeys(x) <- chroms
toTable(x)
## probe_id chromosome
## 1 1029_s_at 4
## 2 1036_at 4
## 3 1058_at 13
## 4 1065_at 13
## 5 1115_at 4
## 6 1189_at 13
## 7 1198_at 13
## 8 1219_at 4
## 9 1220_g_at 4
## 10 1249_at 4
## 11 1285_at 4
## 12 1303_at 4
## 13 1325_at 4
## 14 1348_s_at 13
## 15 1369_s_at 4
## 16 1377_at 4
## 17 1378_g_at 4
## 18 1451_s_at 13
## 19 1503_at 13
## 20 1507_s_at 4
## 21 1527_s_at 13
## 22 1528_at 13
## 23 1529_at 13
## 24 1530_g_at 13
## 25 1531_at 13
## 26 1532_g_at 13
## 27 1538_s_at 4
## 28 1542_at 4
## 29 1545_g_at 13
```
How to use bimaps from the ".db" annotation packages

```
## 30 1567..at  13
## 31 1570..f.at  13
## 32 1571..f.at  13
## 33 1593..at  4
## 34 1597..at  13
## 35 1598..g.at  13
## 36 159..at  4
## 37 1600..at  4
## 38 1604..at  4
## 39 1605..g.at  4
## 40 1616..at  13
## 41 1624..at  4
## 42 1629..s.at  4
## 43 1653..at  4
## 44 1670..at  13
## 45 1672..f.at  13
## 46 1679..at  4
## 47 1708..at  4
## 48 1709..g.at  4
## 49 170..at  13
## 50 1720..at  4
## 51 1721..g.at  4
## 52 1731..at  4
## 53 1732..at  4
## 54 1819..at  13
## 55 1828..s.at  4
## 56 1836..at  4
## 57 1883..s.at  4
## 58 1888..s.at  4
## 59 1900..at  13
## 60 1905..s.at  13
## 61 1913..at  4
## 62 1914..at  13
## 63 1931..at  13
## 64 1934..s.at  4
## 65 1943..at  4
## 66 1954..at  4
## 67 1963..at  13
## 68 1964..g.at  13
## 69 1968..g.at  4
## 70 1987..at  4
## 71 1988..at  4
## 72 1989..at  13
## 73 1990..g.at  13
## 74 2044..s.at  13
```
How to use bimaps from the ".db" annotation packages

```plaintext
## 75  2062_at  4
## 76  2092_s_at  4
## 77   214_at  4
## 78   215_g_at  4
## 79  252_g_at  13
## 80  253_g_at  13
## 81  260_at  4
## 82  281_s_at  4
## 83  31314_at  4
## 84  31320_at  13
## 85  31333_at  4
## 86  31345_at  4
## 87  31349_at  4
## 88  31356_at  4
## 89  31382_f_at  4
## 90  31404_at  13
## 91  31408_at  4
## 92  31464_at  13
## 93  31465_g_at  13
## 94  31516_f_at  13
## 95  31543_at  4
## 96  31562_at  13
## 97  31584_at  13
## 98  31628_at  13
## 99  31631_f_at  4
##100  31639_f_at  13
##101  31640_r_at  13
##102  31670_s_at  4
##103  31684_at  4
##104  31686_at  4
##105  31706_at  4
##106  31744_at  4
##107  31753_at  13
##108  31790_at  13
##109  31792_at  4
##110  31805_at  4
##111  31811_r_at  4
##112  31847_at  13
##113  31849_at  13
##114  31851_at  13
##115  31876_r_at  4
##116  31894_at  4
##117  31969_i_at  4
##118  31970_r_at  4
##119  32006_r_at  4
```

13
How to use bimaps from the ".db" annotation packages

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>32026_s_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>32080_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>32102_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>32145_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>32146_s_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>32147_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>32148_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>32180_s_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>32200_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>32299_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>32337_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>32349_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>32353_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>32357_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>32368_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>32393_s_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>32439_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>32446_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>32449_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>32465_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>32482_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>32506_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>32570_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>32580_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>32595_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>32602_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>32641_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>32675_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>32703_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>32768_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>32769_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>32770_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>32771_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>32812_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>32822_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>32832_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>32862_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>32906_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>32979_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>32986_s_at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>32998_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>33013_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>33068_f_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>33069_f_at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>33100_at</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
## How to use bimaps from the ".db" annotation packages

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>##</td>
<td>165</td>
<td>33150..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>166</td>
<td>33151..s..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>167</td>
<td>33155..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>168</td>
<td>33156..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>169</td>
<td>33168..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>170</td>
<td>33171..s..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>171</td>
<td>33172..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>172</td>
<td>33173..g..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>173</td>
<td>33199..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>174</td>
<td>33208..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>175</td>
<td>33241..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>176</td>
<td>33249..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>177</td>
<td>33267..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>178</td>
<td>33276..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>179</td>
<td>33299..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>180</td>
<td>33318..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>181</td>
<td>33356..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>182</td>
<td>33359..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>183</td>
<td>33369..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>184</td>
<td>33370..r..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>185</td>
<td>33382..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>186</td>
<td>33483..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>187</td>
<td>33488..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>188</td>
<td>33490..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>189</td>
<td>33494..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>190</td>
<td>33519..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>191</td>
<td>33520..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>192</td>
<td>33525..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>193</td>
<td>33526..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>194</td>
<td>33529..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>195</td>
<td>33536..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>196</td>
<td>33544..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>197</td>
<td>33564..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>198</td>
<td>33576..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>199</td>
<td>33584..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>200</td>
<td>33596..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>201</td>
<td>33657..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>202</td>
<td>33687..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>203</td>
<td>33700..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>204</td>
<td>33733..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>205</td>
<td>33791..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>206</td>
<td>33823..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>207</td>
<td>33827..at</td>
<td>13</td>
</tr>
<tr>
<td>##</td>
<td>208</td>
<td>33837..at</td>
<td>4</td>
</tr>
<tr>
<td>##</td>
<td>209</td>
<td>33859..at</td>
<td>13</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

<table>
<thead>
<tr>
<th>#</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>210</td>
<td>33975.at</td>
<td>4</td>
</tr>
<tr>
<td>211</td>
<td>33990.at</td>
<td>4</td>
</tr>
<tr>
<td>212</td>
<td>33991.g.at</td>
<td>4</td>
</tr>
<tr>
<td>213</td>
<td>33992.at</td>
<td>4</td>
</tr>
<tr>
<td>214</td>
<td>33997.at</td>
<td>4</td>
</tr>
<tr>
<td>215</td>
<td>34021.at</td>
<td>4</td>
</tr>
<tr>
<td>216</td>
<td>34022.at</td>
<td>4</td>
</tr>
<tr>
<td>217</td>
<td>34026.at</td>
<td>13</td>
</tr>
<tr>
<td>218</td>
<td>34029.at</td>
<td>4</td>
</tr>
<tr>
<td>219</td>
<td>34048.at</td>
<td>4</td>
</tr>
<tr>
<td>220</td>
<td>34051.at</td>
<td>13</td>
</tr>
<tr>
<td>221</td>
<td>34058.at</td>
<td>4</td>
</tr>
<tr>
<td>222</td>
<td>34075.at</td>
<td>4</td>
</tr>
<tr>
<td>223</td>
<td>34122.at</td>
<td>4</td>
</tr>
<tr>
<td>224</td>
<td>34131.at</td>
<td>4</td>
</tr>
<tr>
<td>225</td>
<td>34144.at</td>
<td>4</td>
</tr>
<tr>
<td>226</td>
<td>34145.at</td>
<td>4</td>
</tr>
<tr>
<td>227</td>
<td>34170.s.at</td>
<td>4</td>
</tr>
<tr>
<td>228</td>
<td>34181.at</td>
<td>4</td>
</tr>
<tr>
<td>229</td>
<td>34198.at</td>
<td>4</td>
</tr>
<tr>
<td>230</td>
<td>34211.at</td>
<td>13</td>
</tr>
<tr>
<td>231</td>
<td>34225.at</td>
<td>4</td>
</tr>
<tr>
<td>232</td>
<td>34239.at</td>
<td>13</td>
</tr>
<tr>
<td>233</td>
<td>34240.s.at</td>
<td>13</td>
</tr>
<tr>
<td>234</td>
<td>34247.at</td>
<td>4</td>
</tr>
<tr>
<td>235</td>
<td>34248.at</td>
<td>4</td>
</tr>
<tr>
<td>236</td>
<td>34275.s.at</td>
<td>4</td>
</tr>
<tr>
<td>237</td>
<td>34284.at</td>
<td>13</td>
</tr>
<tr>
<td>238</td>
<td>34307.at</td>
<td>13</td>
</tr>
<tr>
<td>239</td>
<td>34319.at</td>
<td>4</td>
</tr>
<tr>
<td>240</td>
<td>34324.at</td>
<td>13</td>
</tr>
<tr>
<td>241</td>
<td>34334.at</td>
<td>13</td>
</tr>
<tr>
<td>242</td>
<td>34335.at</td>
<td>13</td>
</tr>
<tr>
<td>243</td>
<td>34341.at</td>
<td>4</td>
</tr>
<tr>
<td>244</td>
<td>34342.s.at</td>
<td>4</td>
</tr>
<tr>
<td>245</td>
<td>34353.at</td>
<td>4</td>
</tr>
<tr>
<td>246</td>
<td>34398.at</td>
<td>13</td>
</tr>
<tr>
<td>247</td>
<td>34411.at</td>
<td>4</td>
</tr>
<tr>
<td>248</td>
<td>34423.at</td>
<td>4</td>
</tr>
<tr>
<td>249</td>
<td>34459.at</td>
<td>13</td>
</tr>
<tr>
<td>250</td>
<td>34476.r.at</td>
<td>4</td>
</tr>
<tr>
<td>251</td>
<td>34482.at</td>
<td>4</td>
</tr>
<tr>
<td>252</td>
<td>34512.at</td>
<td>4</td>
</tr>
<tr>
<td>253</td>
<td>34551.at</td>
<td>4</td>
</tr>
<tr>
<td>254</td>
<td>34564.at</td>
<td>4</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

<table>
<thead>
<tr>
<th>#</th>
<th>Identifier</th>
<th>Value</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>34565.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>34578.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>257</td>
<td>34583.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>34596.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>34637.f.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>34638.r.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>261</td>
<td>34657.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>262</td>
<td>34672.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>263</td>
<td>34745.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>264</td>
<td>34803.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>265</td>
<td>34898.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>266</td>
<td>34953.i.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>267</td>
<td>34954.r.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>268</td>
<td>34955.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>269</td>
<td>34973.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>34984.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>271</td>
<td>34988.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>272</td>
<td>35020.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>273</td>
<td>35021.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>274</td>
<td>35025.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>275</td>
<td>35028.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>276</td>
<td>35039.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>277</td>
<td>35053.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>278</td>
<td>35061.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>279</td>
<td>35063.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>280</td>
<td>35081.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>281</td>
<td>35105.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>282</td>
<td>35107.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>283</td>
<td>35110.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>284</td>
<td>35131.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>285</td>
<td>35134.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>286</td>
<td>35140.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>287</td>
<td>35147.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>288</td>
<td>35164.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>289</td>
<td>35181.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>290</td>
<td>35182.f.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>291</td>
<td>35193.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>292</td>
<td>35213.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>293</td>
<td>35214.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>294</td>
<td>35215.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>295</td>
<td>35220.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>296</td>
<td>35285.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>297</td>
<td>35306.at</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>298</td>
<td>35344.at</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>299</td>
<td>35356.at</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>35357</td>
<td>.at</td>
</tr>
<tr>
<td>301</td>
<td>35371</td>
<td>.at</td>
</tr>
<tr>
<td>302</td>
<td>35372</td>
<td>.r.at</td>
</tr>
<tr>
<td>303</td>
<td>35400</td>
<td>.at</td>
</tr>
<tr>
<td>304</td>
<td>35410</td>
<td>.at</td>
</tr>
<tr>
<td>305</td>
<td>35435</td>
<td>.s.at</td>
</tr>
<tr>
<td>306</td>
<td>35437</td>
<td>.at</td>
</tr>
<tr>
<td>307</td>
<td>35469</td>
<td>.at</td>
</tr>
<tr>
<td>308</td>
<td>35470</td>
<td>.at</td>
</tr>
<tr>
<td>309</td>
<td>35471</td>
<td>.g.at</td>
</tr>
<tr>
<td>310</td>
<td>35481</td>
<td>.at</td>
</tr>
<tr>
<td>311</td>
<td>35507</td>
<td>.at</td>
</tr>
<tr>
<td>312</td>
<td>35523</td>
<td>.at</td>
</tr>
<tr>
<td>313</td>
<td>35554</td>
<td>.f.at</td>
</tr>
<tr>
<td>314</td>
<td>35555</td>
<td>.r.at</td>
</tr>
<tr>
<td>315</td>
<td>35564</td>
<td>.at</td>
</tr>
<tr>
<td>316</td>
<td>35591</td>
<td>.at</td>
</tr>
<tr>
<td>317</td>
<td>35656</td>
<td>.at</td>
</tr>
<tr>
<td>318</td>
<td>35662</td>
<td>.at</td>
</tr>
<tr>
<td>319</td>
<td>35664</td>
<td>.at</td>
</tr>
<tr>
<td>320</td>
<td>35678</td>
<td>.at</td>
</tr>
<tr>
<td>321</td>
<td>35698</td>
<td>.at</td>
</tr>
<tr>
<td>322</td>
<td>35725</td>
<td>.at</td>
</tr>
<tr>
<td>323</td>
<td>35730</td>
<td>.at</td>
</tr>
<tr>
<td>324</td>
<td>35777</td>
<td>.at</td>
</tr>
<tr>
<td>325</td>
<td>35793</td>
<td>.at</td>
</tr>
<tr>
<td>326</td>
<td>35827</td>
<td>.at</td>
</tr>
<tr>
<td>327</td>
<td>35837</td>
<td>.at</td>
</tr>
<tr>
<td>328</td>
<td>35845</td>
<td>.at</td>
</tr>
<tr>
<td>329</td>
<td>35871</td>
<td>.s.at</td>
</tr>
<tr>
<td>330</td>
<td>35877</td>
<td>.at</td>
</tr>
<tr>
<td>331</td>
<td>35904</td>
<td>.at</td>
</tr>
<tr>
<td>332</td>
<td>35939</td>
<td>.s.at</td>
</tr>
<tr>
<td>333</td>
<td>35940</td>
<td>.at</td>
</tr>
<tr>
<td>334</td>
<td>35949</td>
<td>.at</td>
</tr>
<tr>
<td>335</td>
<td>35972</td>
<td>.at</td>
</tr>
<tr>
<td>336</td>
<td>35989</td>
<td>.at</td>
</tr>
<tr>
<td>337</td>
<td>35991</td>
<td>.at</td>
</tr>
<tr>
<td>338</td>
<td>36012</td>
<td>.at</td>
</tr>
<tr>
<td>339</td>
<td>36013</td>
<td>.at</td>
</tr>
<tr>
<td>340</td>
<td>36017</td>
<td>.at</td>
</tr>
<tr>
<td>341</td>
<td>36021</td>
<td>.at</td>
</tr>
<tr>
<td>342</td>
<td>36031</td>
<td>.at</td>
</tr>
<tr>
<td>343</td>
<td>36046</td>
<td>.at</td>
</tr>
<tr>
<td>344</td>
<td>36047</td>
<td>.at</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

<table>
<thead>
<tr>
<th>#</th>
<th>Value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>345</td>
<td>36065.at</td>
<td>4</td>
</tr>
<tr>
<td>346</td>
<td>36080.at</td>
<td>4</td>
</tr>
<tr>
<td>347</td>
<td>36143.at</td>
<td>4</td>
</tr>
<tr>
<td>348</td>
<td>36157.at</td>
<td>4</td>
</tr>
<tr>
<td>349</td>
<td>36188.at</td>
<td>13</td>
</tr>
<tr>
<td>350</td>
<td>36194.at</td>
<td>4</td>
</tr>
<tr>
<td>351</td>
<td>36212.at</td>
<td>13</td>
</tr>
<tr>
<td>352</td>
<td>36243.at</td>
<td>4</td>
</tr>
<tr>
<td>353</td>
<td>36247.f.at</td>
<td>4</td>
</tr>
<tr>
<td>354</td>
<td>36269.at</td>
<td>4</td>
</tr>
<tr>
<td>355</td>
<td>36274.at</td>
<td>13</td>
</tr>
<tr>
<td>356</td>
<td>36358.at</td>
<td>4</td>
</tr>
<tr>
<td>357</td>
<td>36363.at</td>
<td>4</td>
</tr>
<tr>
<td>358</td>
<td>36433.at</td>
<td>4</td>
</tr>
<tr>
<td>359</td>
<td>36434.r.at</td>
<td>4</td>
</tr>
<tr>
<td>360</td>
<td>36510.at</td>
<td>13</td>
</tr>
<tr>
<td>361</td>
<td>36521.at</td>
<td>13</td>
</tr>
<tr>
<td>362</td>
<td>36606.at</td>
<td>4</td>
</tr>
<tr>
<td>363</td>
<td>36622.at</td>
<td>4</td>
</tr>
<tr>
<td>364</td>
<td>36627.at</td>
<td>4</td>
</tr>
<tr>
<td>365</td>
<td>36659.at</td>
<td>13</td>
</tr>
<tr>
<td>366</td>
<td>36717.at</td>
<td>4</td>
</tr>
<tr>
<td>367</td>
<td>36788.at</td>
<td>13</td>
</tr>
<tr>
<td>368</td>
<td>367.at</td>
<td>13</td>
</tr>
<tr>
<td>369</td>
<td>36814.at</td>
<td>4</td>
</tr>
<tr>
<td>370</td>
<td>36830.at</td>
<td>13</td>
</tr>
<tr>
<td>371</td>
<td>36913.at</td>
<td>4</td>
</tr>
<tr>
<td>372</td>
<td>36914.at</td>
<td>4</td>
</tr>
<tr>
<td>373</td>
<td>36915.at</td>
<td>4</td>
</tr>
<tr>
<td>374</td>
<td>36918.at</td>
<td>4</td>
</tr>
<tr>
<td>375</td>
<td>36939.at</td>
<td>4</td>
</tr>
<tr>
<td>376</td>
<td>36968.s.at</td>
<td>13</td>
</tr>
<tr>
<td>377</td>
<td>36990.at</td>
<td>4</td>
</tr>
<tr>
<td>378</td>
<td>37006.at</td>
<td>4</td>
</tr>
<tr>
<td>379</td>
<td>37019.at</td>
<td>4</td>
</tr>
<tr>
<td>380</td>
<td>37023.at</td>
<td>13</td>
</tr>
<tr>
<td>381</td>
<td>37056.at</td>
<td>4</td>
</tr>
<tr>
<td>382</td>
<td>37058.at</td>
<td>4</td>
</tr>
<tr>
<td>383</td>
<td>37062.at</td>
<td>4</td>
</tr>
<tr>
<td>384</td>
<td>37067.at</td>
<td>13</td>
</tr>
<tr>
<td>385</td>
<td>37079.at</td>
<td>13</td>
</tr>
<tr>
<td>386</td>
<td>37099.at</td>
<td>13</td>
</tr>
<tr>
<td>387</td>
<td>37109.at</td>
<td>13</td>
</tr>
<tr>
<td>388</td>
<td>37154.at</td>
<td>13</td>
</tr>
<tr>
<td>389</td>
<td>37170.at</td>
<td>4</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>390</td>
<td>37172</td>
<td>at 13</td>
</tr>
<tr>
<td>391</td>
<td>37173</td>
<td>at 4</td>
</tr>
<tr>
<td>392</td>
<td>37187</td>
<td>at 4</td>
</tr>
<tr>
<td>393</td>
<td>37206</td>
<td>at 4</td>
</tr>
<tr>
<td>394</td>
<td>37219</td>
<td>at 4</td>
</tr>
<tr>
<td>395</td>
<td>37223</td>
<td>at 4</td>
</tr>
<tr>
<td>396</td>
<td>37243</td>
<td>at 4</td>
</tr>
<tr>
<td>397</td>
<td>37244</td>
<td>at 13</td>
</tr>
<tr>
<td>398</td>
<td>37280</td>
<td>at 4</td>
</tr>
<tr>
<td>399</td>
<td>37282</td>
<td>at 4</td>
</tr>
<tr>
<td>400</td>
<td>37291</td>
<td>r.at 4</td>
</tr>
<tr>
<td>401</td>
<td>37303</td>
<td>at 13</td>
</tr>
<tr>
<td>402</td>
<td>37322</td>
<td>s.at 4</td>
</tr>
<tr>
<td>403</td>
<td>37323</td>
<td>r.at 4</td>
</tr>
<tr>
<td>404</td>
<td>37356</td>
<td>r.at 4</td>
</tr>
<tr>
<td>405</td>
<td>37366</td>
<td>at 4</td>
</tr>
<tr>
<td>406</td>
<td>37404</td>
<td>at 4</td>
</tr>
<tr>
<td>407</td>
<td>37416</td>
<td>at 4</td>
</tr>
<tr>
<td>408</td>
<td>37472</td>
<td>at 4</td>
</tr>
<tr>
<td>409</td>
<td>37518</td>
<td>at 13</td>
</tr>
<tr>
<td>410</td>
<td>37520</td>
<td>at 4</td>
</tr>
<tr>
<td>411</td>
<td>37521</td>
<td>s.at 4</td>
</tr>
<tr>
<td>412</td>
<td>37522</td>
<td>r.at 4</td>
</tr>
<tr>
<td>413</td>
<td>37571</td>
<td>at 13</td>
</tr>
<tr>
<td>414</td>
<td>37578</td>
<td>at 4</td>
</tr>
<tr>
<td>415</td>
<td>37593</td>
<td>at 13</td>
</tr>
<tr>
<td>416</td>
<td>37619</td>
<td>at 4</td>
</tr>
<tr>
<td>417</td>
<td>37658</td>
<td>at 13</td>
</tr>
<tr>
<td>418</td>
<td>37707</td>
<td>i.at 4</td>
</tr>
<tr>
<td>419</td>
<td>37708</td>
<td>r.at 4</td>
</tr>
<tr>
<td>420</td>
<td>37723</td>
<td>at 4</td>
</tr>
<tr>
<td>421</td>
<td>37747</td>
<td>at 4</td>
</tr>
<tr>
<td>422</td>
<td>37748</td>
<td>at 4</td>
</tr>
<tr>
<td>423</td>
<td>37752</td>
<td>at 4</td>
</tr>
<tr>
<td>424</td>
<td>37757</td>
<td>at 13</td>
</tr>
<tr>
<td>425</td>
<td>37840</td>
<td>at 4</td>
</tr>
<tr>
<td>426</td>
<td>37852</td>
<td>at 4</td>
</tr>
<tr>
<td>427</td>
<td>37926</td>
<td>at 13</td>
</tr>
<tr>
<td>428</td>
<td>37930</td>
<td>at 13</td>
</tr>
<tr>
<td>429</td>
<td>37964</td>
<td>at 4</td>
</tr>
<tr>
<td>430</td>
<td>38008</td>
<td>at 4</td>
</tr>
<tr>
<td>431</td>
<td>38016</td>
<td>at 4</td>
</tr>
<tr>
<td>432</td>
<td>38024</td>
<td>at 4</td>
</tr>
<tr>
<td>433</td>
<td>38025</td>
<td>r.at 4</td>
</tr>
<tr>
<td>434</td>
<td>38035</td>
<td>at 13</td>
</tr>
</tbody>
</table>
# How to use bimaps from the ".db" annotation packages

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>435</td>
<td>38065_at</td>
<td>4</td>
</tr>
<tr>
<td>436</td>
<td>38102_at</td>
<td>13</td>
</tr>
<tr>
<td>437</td>
<td>38120_at</td>
<td>4</td>
</tr>
<tr>
<td>438</td>
<td>38168_at</td>
<td>4</td>
</tr>
<tr>
<td>439</td>
<td>38254_at</td>
<td>4</td>
</tr>
<tr>
<td>440</td>
<td>38304_r_at</td>
<td>13</td>
</tr>
<tr>
<td>441</td>
<td>38350_f_at</td>
<td>13</td>
</tr>
<tr>
<td>442</td>
<td>38353_at</td>
<td>13</td>
</tr>
<tr>
<td>443</td>
<td>38375_at</td>
<td>13</td>
</tr>
<tr>
<td>444</td>
<td>38438_at</td>
<td>4</td>
</tr>
<tr>
<td>445</td>
<td>38485_at</td>
<td>4</td>
</tr>
<tr>
<td>446</td>
<td>38488_s_at</td>
<td>4</td>
</tr>
<tr>
<td>447</td>
<td>38489_at</td>
<td>4</td>
</tr>
<tr>
<td>448</td>
<td>38587_at</td>
<td>4</td>
</tr>
<tr>
<td>449</td>
<td>38606_at</td>
<td>4</td>
</tr>
<tr>
<td>450</td>
<td>38615_at</td>
<td>13</td>
</tr>
<tr>
<td>451</td>
<td>38639_at</td>
<td>4</td>
</tr>
<tr>
<td>452</td>
<td>38643_at</td>
<td>4</td>
</tr>
<tr>
<td>453</td>
<td>38649_at</td>
<td>13</td>
</tr>
<tr>
<td>454</td>
<td>38714_at</td>
<td>4</td>
</tr>
<tr>
<td>455</td>
<td>38715_at</td>
<td>4</td>
</tr>
<tr>
<td>456</td>
<td>38736_at</td>
<td>4</td>
</tr>
<tr>
<td>457</td>
<td>38751_i_at</td>
<td>4</td>
</tr>
<tr>
<td>458</td>
<td>38752_r_at</td>
<td>4</td>
</tr>
<tr>
<td>459</td>
<td>38767_at</td>
<td>4</td>
</tr>
<tr>
<td>460</td>
<td>38768_at</td>
<td>4</td>
</tr>
<tr>
<td>461</td>
<td>38778_at</td>
<td>4</td>
</tr>
<tr>
<td>462</td>
<td>38821_at</td>
<td>4</td>
</tr>
<tr>
<td>463</td>
<td>38825_at</td>
<td>4</td>
</tr>
<tr>
<td>464</td>
<td>38838_at</td>
<td>4</td>
</tr>
<tr>
<td>465</td>
<td>38854_at</td>
<td>4</td>
</tr>
<tr>
<td>466</td>
<td>38891_at</td>
<td>4</td>
</tr>
<tr>
<td>467</td>
<td>38923_at</td>
<td>4</td>
</tr>
<tr>
<td>468</td>
<td>38957_at</td>
<td>13</td>
</tr>
<tr>
<td>469</td>
<td>38972_at</td>
<td>13</td>
</tr>
<tr>
<td>470</td>
<td>38988_at</td>
<td>4</td>
</tr>
<tr>
<td>471</td>
<td>39028_at</td>
<td>13</td>
</tr>
<tr>
<td>472</td>
<td>39032_at</td>
<td>13</td>
</tr>
<tr>
<td>473</td>
<td>39037_at</td>
<td>4</td>
</tr>
<tr>
<td>474</td>
<td>39056_at</td>
<td>4</td>
</tr>
<tr>
<td>475</td>
<td>39083_at</td>
<td>4</td>
</tr>
<tr>
<td>476</td>
<td>39131_at</td>
<td>13</td>
</tr>
<tr>
<td>477</td>
<td>39132_at</td>
<td>4</td>
</tr>
<tr>
<td>478</td>
<td>39208_i_at</td>
<td>4</td>
</tr>
<tr>
<td>479</td>
<td>39209_r_at</td>
<td>4</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

```plaintext
## 480 39224_at 4
## 481 39256_at 13
## 482 39257_at 13
## 483 39269_at 13
## 484 39295_s_at 4
## 485 39333_at 13
## 486 39337_at 4
## 487 39355_at 4
## 488 39369_at 4
## 489 39380_at 4
## 490 39382_at 4
## 491 39405_at 13
## 492 39469_s_at 13
## 493 39475_at 4
## 494 39481_at 4
## 495 39488_at 13
## 496 39489_g_at 13
## 497 39535_at 4
## 498 39554_at 4
## 499 39555_at 4
## 500 39576_at 4
## 501 39579_at 13
## 502 39600_at 4
## 503 39634_at 4
## 504 39662_s_at 4
## 505 39665_at 4
## 506 39680_at 4
## 507 39690_at 4
## 508 39698_at 4
## 509 39734_at 4
## 510 39746_at 4
## 511 39748_at 13
## 512 39758_f_at 13
## 513 39777_at 13
## 514 39786_at 4
## 515 39847_at 4
## 516 39850_at 4
## 517 39851_at 4
## 518 39852_at 13
## 519 39878_at 13
## 520 39897_at 4
## 521 39924_at 13
## 522 39929_at 4
## 523 39955_at 13
## 524 39960_at 4
```
### How to use bimaps from the ".db" annotation packages

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>525</td>
<td>39979</td>
<td>.at 13</td>
</tr>
<tr>
<td>526</td>
<td>40018</td>
<td>.at 13</td>
</tr>
<tr>
<td>527</td>
<td>40058</td>
<td>.s.at 4</td>
</tr>
<tr>
<td>528</td>
<td>40059</td>
<td>_r.at 4</td>
</tr>
<tr>
<td>529</td>
<td>40060</td>
<td>_r.at 4</td>
</tr>
<tr>
<td>530</td>
<td>40067</td>
<td>_at 13</td>
</tr>
<tr>
<td>531</td>
<td>40072</td>
<td>_at 13</td>
</tr>
<tr>
<td>532</td>
<td>40082</td>
<td>_at 4</td>
</tr>
<tr>
<td>533</td>
<td>400</td>
<td>_at 13</td>
</tr>
<tr>
<td>534</td>
<td>40114</td>
<td>_at 4</td>
</tr>
<tr>
<td>535</td>
<td>40121</td>
<td>_at 4</td>
</tr>
<tr>
<td>536</td>
<td>40148</td>
<td>_at 4</td>
</tr>
<tr>
<td>537</td>
<td>40180</td>
<td>_at 13</td>
</tr>
<tr>
<td>538</td>
<td>40181</td>
<td>_f.at 13</td>
</tr>
<tr>
<td>539</td>
<td>40199</td>
<td>_at 4</td>
</tr>
<tr>
<td>540</td>
<td>40217</td>
<td>_s.at 4</td>
</tr>
<tr>
<td>541</td>
<td>40218</td>
<td>_at 4</td>
</tr>
<tr>
<td>542</td>
<td>40225</td>
<td>_at 4</td>
</tr>
<tr>
<td>543</td>
<td>40226</td>
<td>_at 4</td>
</tr>
<tr>
<td>544</td>
<td>40272</td>
<td>_at 4</td>
</tr>
<tr>
<td>545</td>
<td>40310</td>
<td>_at 4</td>
</tr>
<tr>
<td>546</td>
<td>40312</td>
<td>_at 13</td>
</tr>
<tr>
<td>547</td>
<td>40323</td>
<td>_at 4</td>
</tr>
<tr>
<td>548</td>
<td>40349</td>
<td>_at 4</td>
</tr>
<tr>
<td>549</td>
<td>40354</td>
<td>_at 4</td>
</tr>
<tr>
<td>550</td>
<td>40392</td>
<td>_at 13</td>
</tr>
<tr>
<td>551</td>
<td>40404</td>
<td>_s.at 13</td>
</tr>
<tr>
<td>552</td>
<td>40449</td>
<td>_at 4</td>
</tr>
<tr>
<td>553</td>
<td>40454</td>
<td>_at 4</td>
</tr>
<tr>
<td>554</td>
<td>40456</td>
<td>_at 4</td>
</tr>
<tr>
<td>555</td>
<td>40473</td>
<td>_at 13</td>
</tr>
<tr>
<td>556</td>
<td>40492</td>
<td>_at 4</td>
</tr>
<tr>
<td>557</td>
<td>40530</td>
<td>_at 4</td>
</tr>
<tr>
<td>558</td>
<td>40570</td>
<td>_at 13</td>
</tr>
<tr>
<td>559</td>
<td>40576</td>
<td>_f.at 4</td>
</tr>
<tr>
<td>560</td>
<td>40633</td>
<td>_at 13</td>
</tr>
<tr>
<td>561</td>
<td>40681</td>
<td>_at 13</td>
</tr>
<tr>
<td>562</td>
<td>40697</td>
<td>_at 4</td>
</tr>
<tr>
<td>563</td>
<td>40710</td>
<td>_at 4</td>
</tr>
<tr>
<td>564</td>
<td>40711</td>
<td>_at 4</td>
</tr>
<tr>
<td>565</td>
<td>40727</td>
<td>_at 4</td>
</tr>
<tr>
<td>566</td>
<td>40746</td>
<td>_at 4</td>
</tr>
<tr>
<td>567</td>
<td>40770</td>
<td>_f.at 4</td>
</tr>
<tr>
<td>568</td>
<td>40772</td>
<td>_at 4</td>
</tr>
<tr>
<td>569</td>
<td>40773</td>
<td>_at 4</td>
</tr>
</tbody>
</table>
How to use bimaps from the ".db" annotation packages

```bash
## 570 40818_.at 4
## 571 40828_.at 13
## 572 40839_.at 13
## 573 40853_.at 4
## 574 40880_r.at 4
## 575 40893_.at 13
## 576 408_.at 4
## 577 40908_r.at 13
## 578 40943_.at 4
## 579 40970_.at 13
## 580 40990_.at 4
## 581 40991_r.at 4
## 582 40992_s.at 4
## 583 40993_r.at 4
## 584 41014_s.at 4
## 585 41024_f.at 4
## 586 41025_r.at 4
## 587 41026_f.at 4
## 588 41069_.at 13
## 589 41071_.at 4
## 590 41104_.at 4
## 591 41118_.at 13
## 592 41119_f.at 13
## 593 41145_.at 4
## 594 41148_.at 4
## 595 41182_.at 13
## 596 41191_.at 4
## 597 41276_.at 13
## 598 41277_.at 13
## 599 41300_s.at 13
## 600 41301_.at 13
## 601 41308_.at 4
## 602 41309_g.at 4
## 603 41317_.at 13
## 604 41318_g.at 13
## 605 41319_.at 13
## 606 41376_i.at 4
## 607 41377_f.at 4
## 608 41391_.at 4
## 609 41392_.at 4
## 610 41402_.at 4
## 611 41434_.at 4
## 612 41436_.at 13
## 613 41456_.at 4
## 614 41459_.at 13
```
How to use bimaps from the ".db" annotation packages

### Example

```
## 615 41470.at 4
## 616 41491.s.at 13
## 617 41492.r.at 13
## 618 41493.at 13
## 619 41534.at 4
## 620 41555.at 4
## 621 41556.s.at 4
## 622 41585.at 4
## 623 41667.s.at 13
## 624 41668.r.at 13
## 625 41697.at 4
## 626 41801.at 4
## 627 41806.at 4
## 628 41860.at 13
## 629 431.at 4
## 630 504.at 4
## 631 507.s.at 4
## 632 579.at 4
## 633 618.at 4
## 634 630.at 4
## 635 631.g.at 4
## 636 655.at 4
## 637 690.s.at 4
## 638 692.s.at 4
## 639 764.s.at 4
## 640 820.at 4
## 641 886.at 4
## 642 931.at 13
## 643 936.s.at 4
## 644 948.s.at 4
## 645 963.at 13
## 646 975.at 4
## 647 990.at 13
## 648 991.g.at 13

To get this in the classic named-list format:
```
```r
z <- as.list(revmap(x)[chroms])
names(z)
```
```
## [1] "4"  "13"
```
```
z[['Y']]
```
```
## NULL
```
Many of the common methods for accessing Bimap objects return things in list format. This can be convenient. But you have to be careful about this if you want to use unlist(). For example the following will return multiple probes for each chromosome:

```r
chrs = c("12","6")
mget(chrs, revmap(hgu95av2CHR[1:30]), ifnotfound=NA)
## $`12`
## [1] "1018_at" "1019_g_at" "101_at" "1021_at"
## $`6`
## [1] "1007_s_at" "1026_s_at" "1027_at"
```

But look what happens here if we try to unlist that:

```r
unlist(mget(chrs, revmap(hgu95av2CHR[1:30]), ifnotfound=NA))
## 12 12 12 12 6 6
## "1018_at" "1019_g_at" "101_at" "1021_at" "1007_s_at" "1026_s_at"
## 6
## "1027_at"
```

Yuck! One trick that will sometimes help is to use Rfunctionunlist2. But be careful here too. Depending on what step comes next, Rfunctionunlist2 may not really help you...

```r
unlist2(mget(chrs, revmap(hgu95av2CHR[1:30]), ifnotfound=NA))
## 12 12 12 12 6 6
## "1018_at" "1019_g_at" "101_at" "1021_at" "1007_s_at" "1026_s_at"
## 6
## "1027_at"
```

Lets ask if the probes in 'pbids' mapped to cytogenetic location "18q11.2"?

```r
x <- hgu95av2MAP
pbids <- c("38912_at", "41654_at", "907_at", "2053_at", "2054_g_at", "40781_at")
x <- subset(x, Lkeys=pbids, Rkeys="18q11.2")
toTable(x)
```

To coerce this map to a named vector:

```r
pb2cyto <- as.character(x)
pb2cyto[pbids]
```
## How to use bimaps from the ".db" annotation packages

```r
#<NA> <NA> <NA> <NA> <NA> <NA>
# NA NA NA NA NA NA
```

The coercion of the reverse map works too but issues a warning because of the duplicated names for the reasons stated above:

```r
cyto2pb <- as.character(revmap(x))
```

### 2.0.5 Accessing probes that map to multiple targets

In many probe packages, some probes are known to map to multiple genes. The reasons for this can be biological as happens in the arabidopsis packages, but usually it is due to the fact that the genome builds that chip platforms were based on were less stable than desired. Thus what may have originally been a probe designed to measure one thing can end up measuring many things. Usually you don’t want to use probes like this, because if they manufacturer doesn’t know what they map to then their usefulness is definitely suspect. For this reason, by default all chip packages will normally hide such probes in the standard mappings. But sometimes you may want access to the answers that the manufacturer says such a probe will map to. In such cases, you will want to use the `toggleProbes` method. To use this method, just call it on a standard mapping and copy the result into a new mapping (you cannot alter the original mapping). Then treat the new mapping as you would any other mapping.

```r
## How many probes?
dim(hgu95av2ENTREZID)
## [1] 11683 2
## Make a mapping with multiple probes exposed
multi <- toggleProbes(hgu95av2ENTREZID, "all")
## How many probes?
dim(multi)
## [1] 12973 2
```

If you then decide that you want to make a mapping that has only multiple mappings or you wish to revert one of your maps back to the default state of only showing the single mappings then you can use `toggleProbes` to switch back and forth.

```r
## Make a mapping with ONLY multiple probes exposed
multiOnly <- toggleProbes(multi, "multiple")
## How many probes?
dim(multiOnly)
## [1] 1290 2
## Then make a mapping with ONLY single mapping probes
```
How to use bimaps from the ".db" annotation packages

```r
singleOnly <- toggleProbes(multiOnly, "single")
## How many probes?
dim(singleOnly)
## [1] 11683  2
```

Finally, there are also a pair of test methods hasMultiProbes and hasSingleProbes that can be used to see what methods a mapping presently has exposed.

```r
## Test the multiOnly mapping
hasMultiProbes(multiOnly)
## [1] TRUE

hasSingleProbes(multiOnly)
## [1] FALSE

## Test the singleOnly mapping
hasMultiProbes(singleOnly)
## [1] FALSE

hasSingleProbes(singleOnly)
## [1] TRUE
```

2.0.6 Using SQL to access things directly

While the mapping objects provide a lot of convenience, sometimes there are definite benefits to writing a simple SQL query. But in order to do this, it is necessary to know a few things. The 1st thing you will need to know is some SQL. Fortunately, it is quite easy to learn enough basic SQL to get stuff out of a database. Here are 4 basic SQL things that you may find handy:

First, you need to know about SELECT statements. A simple example would look something like this:

```
SELECT * FROM genes;
```
Which would select everything from the genes table.

```
SELECT gene_id FROM genes;
```
Will select only the gene_id field from the genes table.

Second you need to know about WHERE clauses:

```
SELECT gene_id, _id FROM genes WHERE gene_id=1;
```
Will only get records from the genes table where the gene_id is = 1.

Thirdly, you will want to know about an inner join:
How to use bimaps from the ".db" annotation packages

SELECT * FROM genes,chromosomes WHERE genes._id=chromosomes._id;

This is only slightly more complicated to understand. Here we want to get all the records that are in both the 'genes' and 'chromosomes' tables, but we only want ones where the '_id' field is identical. This is known as an inner join because we only want the elements that are in both of these tables with respect to '_id'. There are other kinds of joins that are worth learning about, but most of the time, this is all you will need to do.

Finally, it is worthwhile to learn about the AS keyword which is useful for making long queries easier to read. For the previous example, we could have written it this way to save space:

SELECT * FROM genes AS g,chromosomes AS c WHERE g._id=c._id;

In a simple example like this you might not see a lot of savings from using AS, so lets consider what happens when we want to also specify which fields we want:

SELECT g.gene_id,c.chromosome FROM genes AS g,chromosomes AS c WHERE g._id=c._id;

Now you are most of the way there to being able to query the databases directly. The only other thing you need to know is a little bit about how to access these databases from R. With each package, you will also get a method that will print the schema for its database, you can view this to see what sorts of tables are present etc.

org.Hs.eg_dbschema()

To access the data in a database, you will need to connect to it. Fortunately, each package will automatically give you a connection object to that database when it loads.

org.Hs.eg_dbconn()

You can use this connection object like this:

query <- "SELECT gene_id FROM genes LIMIT 10;"
result = dbGetQuery(org.Hs.eg_dbconn(), query)
result

Exercise 5
Retrieve the entrez gene ID and chromosome by using a database query. Show how you could do the same thing by using toTable

2.0.7 Combining data from multiple annotation packages at the SQL level

For a more complex example, consider the task of obtaining all gene symbols which are probed on a chip that have at least one GO BP ID annotation with evidence code IMP, IGI, IPI, or IDA. Here is one way to extract this using the environment-based packages:
How to use bimaps from the ".db" annotation packages

```r
## Obtain SYMBOLS with at least one GO BP annotation with evidence IMP, IGI, IPI, or IDA.

system.time({
  bpids <- eapply(hgu95av2GO, function(x) {
    if (length(x) == 1 && is.na(x))
      NA
    else {
      sapply(x, function(z) {
        if (z$Ontology == "BP")
          z$GOID
        else
          NA
      })
    }
  })
  bpids <- unlist(bpids)
  bpids <- unique(bpids[!is.na(bpids)])
  g2p <- mget(bpids, hgu95av2GO2PROBE)
  wantedp <- lapply(g2p, function(x) {
    x[names(x) %in% c("IMP", "IGI", "IPI", "IDA")]
  })
  wantedp <- wantedp[sapply(wantedp, length) > 0]
  wantedp <- unique(unlist(wantedp))
  ans <- unlist(mget(wantedp, hgu95av2SYMBOL))
})
length(ans)
an[1:10]
```

All of the above code could have been reduced to a single SQL query with the SQLite-based packages. But to put together this query, you would need to look 1st at the schema to know what tables are present:

```r
hgu95av2_dbschema()
```

This function will give you an output of all the create table statements that were used to generate the hgu95av2 database. In this case, this is a chip package, so you will also need to see the schema for the organism package that it depends on. To learn what package it depends on, look at the ORGPKG value:

```r
hgu95av2ORGPKG
```

Then you can see that schema by looking at its schema method:
How to use bimaps from the ".db" annotation packages

So now we can see that we want to connect the data in the go_bp, and symbol tables from the org.Hs.eg.sqlite database along with the probes data in the hgu95av2.sqlite database. How can we do that?

It turns out that one of the great conveniences of SQLite is that it allows other databases to be ‘ATTACHed’. Thus, we can keep our data in many different databases, and then ‘ATTACH’ them to each other in a modular fashion. The databases for a given build have been built together and frozen into a single version specifically to allow this sort of behavior. To use this feature, the SQLite ATTACH command requires the filename for the database file on your filesystem. Fortunately, R provides a nice system independent way of getting that information. Note that the name of the database is always the same as the name of the package, with the suffix ’.sqlite’:

```r
orgDBLoc = system.file("extdata", "org.Hs.eg.sqlite", package="org.Hs.eg.db")
attachSQL = paste("ATTACH '", orgDBLoc, "' AS orgDB;", sep = "")
dbGetQuery(hgu95av2_dbconn(), attachSQL)
```

Finally, you can assemble a cross-db sql query and use the helper function as follows. Note that when we want to refer to tables in the attached database, we have to use the ‘orgDB’ prefix that we specified in the ‘ATTACH’ query above:

```r
system.time(
  SQL <- "SELECT DISTINCT probe_id,symbol FROM probes, orgDB.gene_info AS gi, orgDB.genes AS g, orgDB.go_bp AS bp WHERE bp._id=g._id AND gi._id=g._id AND probes.gene_id=g.gene_id AND bp.evidence IN ('IPI', 'IDA', 'IMP', 'IGI")
  zz <- dbGetQuery(hgu95av2_dbconn(), SQL)
)
```

Exercise 6

Retrieve the entrez gene ID, chromosome location information and cytoband information by using a single database query.
How to use bimaps from the ".db" annotation packages

Exercise 7

Expand on the example in the text above to combine data from the hgu95av2.db and org.Hs.eg.db with the GO.db package so as to include the GO ID, and term definition in the output.

## Warning in result_fetch(res@ptr, n = n): SQL statements must be issued with dbExecute() or dbSendStatement() instead of dbGetQuery() or dbSendQuery().

## Warning in result_fetch(res@ptr, n = n): SQL statements must be issued with dbExecute() or dbSendStatement() instead of dbGetQuery() or dbSendQuery().

## Warning in result_fetch(res@ptr, n = n): SQL statements must be issued with dbExecute() or dbSendStatement() instead of dbGetQuery() or dbSendQuery().

## Warning in result_fetch(res@ptr, n = n): SQL statements must be issued with dbExecute() or dbSendStatement() instead of dbGetQuery() or dbSendQuery().

The version number of R and packages loaded for generating the vignette were:

```r
## R Under development (unstable) (2023-10-22 r85388)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.3 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.19-bioc/R/lib/libRblas.so
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
## [3] LC_TIME=en_GB LC_COLLATE=C
## [5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8 LC_NAME=C
## [9] LC_ADDRESS=C LC_TELEPHONE=C
##
## time zone: America/New_York
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4  stats   graphics grDevices utils    datasets
## [7] methods  base
##
## other attached packages:
## [1] hgu95av2.db_3.13.0 AnnotationForge_1.45.0
## [3] org.Hs.eg.db_3.18.0 AnnotationDbi_1.65.2
## [5] IRanges_2.37.0     S4Vectors_0.41.1
## [7] Biocbase_2.63.0   BiocGenerics_0.49.1
```
## How to use bimaps from the "db" annotation packages

```r
## [9] DBI_1.1.3        knitr_1.45
##  loaded via a namespace (and not attached):
## [1] bit_4.0.5     BiocStyle_2.31.0
## [3] compiler_4.4.0 BiocManager_1.30.22
## [5] highr_0.10    crayon_1.5.2
## [7] blob_1.2.4    bitops_1.0-7
## [9] Biostrings_2.71.1 png_0.1-8
## [11] yaml_2.3.7    fastmap_1.1.1
## [13] R6_2.5.1      XVector_0.43.0
## [15] GenomeInfoDb_1.39.0 XML_3.99-0.15
## [17] GenomeInfoDbData_1.2.11 rlang_1.1.1
## [19] KEGGREST_1.43.0 cachem_1.0.8
## [21] xfun_0.41      bit64_4.0.5
## [23] RSQLite_2.3.2   memoise_2.0.1
## [25] cli_3.6.1      zlibbioc_1.49.0
## [27] digest_0.6.33  GO.db_3.18.0
## [29] vctrs_0.6.4    evaluate_0.23
## [31] RCurl_1.98-1.13 rmarkdown_2.25
## [33] htmltools_0.5.7
## [35] pkgconfig_2.0.3 htmltools_0.5.7
```