Annotation

Mikhail Dozmorov

2016-10-03

Probe annotation

Any given experiment typically involves a set of known identifers (probes in the case of a microarray experiment). These identifers are typically unique (for any manufacturer). But having Affymetrix IDs ("1001_at") as probe identifiers is not very helpful when we want to know which gene a probe is actually assesses. There are numerous annotation packages available on Bioconductor at http://www.bioconductor.org/packages/release/data/annotation/.

We will consider the Affymetrix human gene chip, hgu133a, for our example. We first load this chip's package and annotate.

```
library("annotate")
# biocLite('hgu133a.db') # Install, if not available
library("hgu133a.db")
ls("package:hgu133a.db")  # Objects in this package
```

```
##  [1] "hgu133a"              "hgu133a_dbconn"       "hgu133a_dbfile"       "hgu133a_dbInfo"       "hgu
## [10] "hgu133aCHRLENGTHS"    "hgu133aCHRLOC"        "hgu133aCHRLOCEND"     "hgu133aENSEMBL"       "hgu
## [19] "hgu133aGO"            "hgu133aGO2ALLPROBES"  "hgu133aGO2PROBE"      "hgu133aMAP"           "hgu
## [28] "hgu133aPATH2PROBE"    "hgu133aPFAM"          "hgu133aPMID"          "hgu133aPMID2PROBE"    "hgu
```

We see the listing of 36 different R objects in this package. Most of them represent mappings from the identifiers on the Affymetrix chip to the different biological resources. You can find out more about them by using the R help system, since each has a manual page that describes the data together with other information such as where, when and what files were used to construct the mappings. Also, each meta-data package has one object that has the same name as the package basename, in this case it is hgu133a. This is function and it can be invoked to find out some of the different statistics regarding the mappings that were done. Its manual page lists all data resources that were used to create the meta-data package.

```
hgu133a()  # Description of an object
```

```
## Quality control information for hgu133a:
##
##
## This package has the following mappings:
##
## hgu133aACCNUM has 22283 mapped keys (of 22283 keys)
## hgu133aALIAS2PROBE has 46188 mapped keys (of 119263 keys)
## hgu133aCHR has 19857 mapped keys (of 22283 keys)
## hgu133aCHRLENGTHS has 93 mapped keys (of 93 keys)
## hgu133aCHRLOC has 19711 mapped keys (of 22283 keys)
## hgu133aCHRLOCEND has 19711 mapped keys (of 22283 keys)
## hgu133aENSEMBL has 19565 mapped keys (of 22283 keys)
## hgu133aENSEMBL2PROBE has 13633 mapped keys (of 27937 keys)
## hgu133aENTREZID has 19860 mapped keys (of 22283 keys)
## hgu133aENZYME has 3056 mapped keys (of 22283 keys)
## hgu133aENZYME2PROBE has 892 mapped keys (of 975 keys)
```

```
## hgu133aGENENAME has 19860 mapped keys (of 22283 keys)
## hgu133aGO has 19230 mapped keys (of 22283 keys)
## hgu133aGO2ALLPROBES has 19454 mapped keys (of 20856 keys)
## hgu133aGO2PROBE has 14985 mapped keys (of 16339 keys)
## hgu133aMAP has 19798 mapped keys (of 22283 keys)
## hgu133aOMIM has 17279 mapped keys (of 22283 keys)
## hgu133aPATH has 7877 mapped keys (of 22283 keys)
## hgu133aPATH2PROBE has 229 mapped keys (of 229 keys)
## hgu133aPMID has 19794 mapped keys (of 22283 keys)
## hgu133aPMID2PROBE has 435275 mapped keys (of 492261 keys)
## hgu133aREFSEQ has 19769 mapped keys (of 22283 keys)
## hgu133aSYMBOL has 19860 mapped keys (of 22283 keys)
## hgu133aUNIGENE has 19822 mapped keys (of 22283 keys)
## hgu133aUNIPROT has 19326 mapped keys (of 22283 keys)
##
##
## Additional Information about this package:
##
## DB schema: HUMANCHIP_DB
## DB schema version: 2.1
## Organism: Homo sapiens
## Date for NCBI data: 2015-Sep27
## Date for GO data: 20150919
## Date for KEGG data: 2011-Mar15
## Date for Golden Path data: 2010-Mar22
## Date for Ensembl data: 2015-Jul16
```

Now let's consider a specific object, say the hgu133aSYMBOL object.

```
hgu133aSYMBOL  # Check specific object
```

```
## SYMBOL map for chip hgu133a (object of class "ProbeAnnDbBimap")
```

If we type its name we see that it is an R environment; all this means is that it is a special data type that is efficient at storing and retrieving mappings between symbols (the Affymetrix identifiers) and associated values (the gene symbols). We can retrieve values from this environment in many different ways. Suppose that we are interested in finding the gene symbol for the Affymetrix probe, 39900_at. Then we can do it in any one of the following ways:

```
get("1316_at", env = hgu133aSYMBOL)  # Accessing elements
```

```
## [1] "THRA"
```

```
hgu133aSYMBOL[["1316_at"]]
```

```
## [1] "THRA"
```

```
hgu133aSYMBOL$"1316_at"
```

```
## [1] "THRA"
```

If you want to get more than one object from an environment you also have a number of choices. You can convert the object info to a data frame, and subset if by your set of IDs. You can extract them one at a time using either a for loop or apply or eapply. It will be more efficient to use `mget` since it does the retrieval using internal code and is optimized. You may also want to turn the environment into a named list, so that you can perform different list operations on it, this can be done using the function contents or as.list.

```
symbols <- data.frame(ACCNUM = sapply(Biobase::contents(hgu133aACCNUM), paste, collapse = ", "), SYMBOL
    paste, collapse = ", "))  # Get all elements as data frame
```

```
symbols[c("1053_at", "121_at", "1294_at", "1316_at"), ]  # Get annotations for several elements
```

```
##         ACCNUM SYMBOL                              DESC
## 1053_at M87338   RFC2  replication factor C subunit 2
## 121_at  X69699   PAX8                    paired box 8
## 1294_at L13852     NA                              NA
## 1316_at X55005   THRA thyroid hormone receptor, alpha
```

```
symbols <- as.list(hgu133aSYMBOL)  # Get all elements as list
length(symbols)  # How many elements total
```

```
## [1] 22283
```

```
names(symbols)[1:5]
```

```
## [1] "1007_s_at" "1053_at"   "117_at"    "121_at"    "1255_g_at"
```

You can convert a particular annotation mapping to a list with as.list:

```
probe2entrez <- as.list(hgu133aENTREZID)
probe2entrez[1:5]
```

```
## $`1007_s_at`
## [1] NA
##
## $`1053_at`
## [1] "5982"
##
## $`117_at`
## [1] "3310"
##
## $`121_at`
## [1] "7849"
##
## $`1255_g_at`
## [1] "2978"
```

Or the reverse mapping:

```
entrez2probe <- as.list(revmap(hgu133aENTREZID))
entrez2probe[1:5]
```

```
## $`1`
## [1] NA
##
## $`2`
## [1] "217757_at"
##
## $`3`
## [1] NA
##
## $`9`
## [1] "214440_at"
##
## $`10`
## [1] "206797_at"
```

If you have a set of probes, you can use these mappings:

```
probes <- c("215134_at", "221987_s_at", "214740_at", "205903_s_at")
entrez <- unlist(probe2entrez[probes])
```

The `unlist` method above only works when you have a "many-to-one" or "one-to-one" mapping of keys to values. If you have "one-to-many" or "many-to-many", you will get back several mapped values for each key:

```
entrez2probe[entrez]
```

```
## $`55361`
## [1] "209345_s_at" "209346_s_at" "215134_at"
##
## $`55720`
## [1] "218155_x_at" "218156_s_at" "221987_s_at"
##
## $<NA>
## NULL
##
## $`3782`
## [1] "205902_at"   "205903_s_at"
```

However, it is recommended to do as the last step, using manufacturer's IDs throughout the analysis.

Accessing probe sequence data.

```
# biocLite('hgu133aprobe')
library(hgu133aprobe)  # Opens library with probe sequence data.
print.data.frame(hgu133aprobe[1:10, ])  # Prints probe sequences and their positions for first two Affy
```

```
##                    sequence   x   y Probe.Set.Name Probe.Interrogation.Position Target.Strandedness
## 1  CACCCAGCTGGTCCTGTGGATGGGA 467 181      1007_s_at                         3330           Antisense
## 2  GCCCCACTGGACAACACTGATTCCT 531 299      1007_s_at                         3443           Antisense
## 3  TGGACCCCACTGGCTGAGAATCTGG  86 557      1007_s_at                         3512           Antisense
## 4  AAATGTTTCCTTGTGCCTGCTCCTG 365 115      1007_s_at                         3563           Antisense
## 5  TCCTTGTGCCTGCTCCTGTACTTGT 207 605      1007_s_at                         3570           Antisense
## 6  TGCCTGCTCCTGTACTTGTCCTCAG 593 599      1007_s_at                         3576           Antisense
## 7  TCCTGTACTTGTCCTCAGCTTGGGC 425 607      1007_s_at                         3583           Antisense
## 8  ACTTGTCCTCAGCTTGGGCTTCTTC 552 101      1007_s_at                         3589           Antisense
## 9  TCCTCCATCACCTGAAACACTGGAC 680 607      1007_s_at                         3615           Antisense
## 10 AAGCCTATACGTTTCTGTGGAGTAA 532 139      1007_s_at                         3713           Antisense
```

```
# biocLite('Biostrings')
library(Biostrings)
pm <- DNAStringSet(hgu133aprobe$sequence)
names(pm) <- hgu133aprobe$Probe.Set.Name  # Stores probe sequences as DNAStringSet object. See HT-Seq m
head(pm)  # Look what's there
```

```
##   A DNAStringSet instance of length 6
##     width seq
## [1]    25 CACCCAGCTGGTCCTGTGGATGGGA
## [2]    25 GCCCCACTGGACAACACTGATTCCT
## [3]    25 TGGACCCCACTGGCTGAGAATCTGG
## [4]    25 AAATGTTTCCTTGTGCCTGCTCCTG
## [5]    25 TCCTTGTGCCTGCTCCTGTACTTGT
## [6]    25 TGCCTGCTCCTGTACTTGTCCTCAG
```