Some ways to get annotations about sequence data

Marc Carlson

November 12, 2008

1 Introduction

There are two good ways to get annotations about sequence data into bioconductor. You can either use an organism level annoatation package, or if you need more information you can also use biomaRt.

2 Using an annotation package

The org packages provide a range of different gene-centric annotations about an organism. For most organisms, the packages are entrez gene centric. One such package is the org.Mm.eg.db package. The org packages should all contain the chromosome a gene maps to, as well as a genes start site, stop site and the orientation of the gene.

```
> ##load the package
> library("org.Mm.eg.db")
> ##look what we just loaded
> ls(2)

 [1] "org.Mm.eg"                "org.Mm.egACCNUM"
 [3] "org.Mm.egACCNUM2EG"       "org.Mm.egALIAS2EG"
 [5] "org.Mm.egCHR"             "org.Mm.egCHRLENGTHS"
 [7] "org.Mm.egCHRLOC"          "org.Mm.egCHRLOCEND"
 [9] "org.Mm.eg_dbconn"         "org.Mm.eg_dbfile"
[11] "org.Mm.eg_dbInfo"         "org.Mm.eg_dbschema"
[13] "org.Mm.egENSEMBL"         "org.Mm.egENSEMBL2EG"
[15] "org.Mm.egENSEMBLPROT"     "org.Mm.egENSEMBLPROT2EG"
[17] "org.Mm.egENSEMBLTRANS"    "org.Mm.egENSEMBLTRANS2EG"
[19] "org.Mm.egENZYME"          "org.Mm.egENZYME2EG"
```

```
[21] "org.Mm.egGENENAME"        "org.Mm.egGO"
[23] "org.Mm.egGO2ALLEGS"       "org.Mm.egGO2EG"
[25] "org.Mm.egMAP"             "org.Mm.egMAP2EG"
[27] "org.Mm.egMAPCOUNTS"       "org.Mm.egMGI"
[29] "org.Mm.egMGI2EG"          "org.Mm.egORGANISM"
[31] "org.Mm.egPATH"            "org.Mm.egPATH2EG"
[33] "org.Mm.egPFAM"            "org.Mm.egPMID"
[35] "org.Mm.egPMID2EG"         "org.Mm.egPROSITE"
[37] "org.Mm.egREFSEQ"          "org.Mm.egREFSEQ2EG"
[39] "org.Mm.egSYMBOL"          "org.Mm.egSYMBOL2EG"
[41] "org.Mm.egUNIGENE"         "org.Mm.egUNIGENE2EG"
[43] "org.Mm.egUNIPROT"
```

```
> ##Data for the org packages comes from the latest UCSC data
> ##which is from NCBI (UCSC calls it mm9, NCBI Build 37.1)
>
> ##Have a peak:
> as.list(org.Mm.egCHRLOC)[1:4]
```

```
$`100008564`
[1] NA

$`100008567`
[1] NA

$`100009600`
        9
-20866836

$`100009609`
        7
-92088678
```

```
> ##Notice For each entrez gene ID, there is a start location for the UCSC genome
> ## negative values are the minus strand
> ## positve values are the positive strand
>
> ## for the stop locations use:
> as.list(org.Mm.egCHRLOCEND)[1:4]
```

```
$`100008564`
[1] NA
```

```
$`100008567`
[1] NA

$`100009600`
        9
-20871537

$`100009609`
        7
-92112519

> ##or can use get, mget etc. with the entrez gene ID
> EGs = c("18392","18414","56513")
> mget(EGs, org.Mm.egCHRLOC, ifnotfound=NA)

$`18392`
        4
108252058

$`18414`
      15
-6763576

$`56513`
        8         8
108225548 108225053

> mget(EGs, org.Mm.egCHRLOCEND, ifnotfound=NA)

$`18392`
        4
108287436

$`18414`
      15
-6824313

$`56513`
        8         8
108227394 108227394
```

```
> ##You can also retrieve ENSEMBL IDs using this package
> mget(EGs, org.Mm.egENSEMBL, ifnotfound=NA)

$`18392`
[1] "ENSMUSG00000028587"

$`18414`
[1] "ENSMUSG00000022146"

$`56513`
[1] "ENSMUSG00000005699"
```

3 Using biomaRt

If you can't find what you are looking for in the annotation packages, you can also consider trying biomaRt. biomaRt is slower, not versioned, and requires a greater level of knowledge to use, but sometimes there is information there that is not included in the annoation packages yet. An example of this are the exon boundaries. One thing to pay attention to is that the biomaRt ensembl database used in this example is a different source of annotations from the annotation packages above (which come from UCSC). So we recommend against mixing and matching these two annotation sets as there might be disagreements.

Remember also when using biomaRt, that it has to talk to an external server most of the time. So you may have to repeat some of the following steps if the internet is not cooperating.

```
> ##Getting the data from biomaRt:
>
> library("biomaRt")
> ##Choose a database
> listMarts()[1:5,]

  biomart                           version
1 ensembl      ENSEMBL 50 GENES (SANGER UK)
2     snp ENSEMBL 50 VARIATION  (SANGER UK)
3    vega              VEGA 32  (SANGER UK)
4     msd            MSD PROTOTYPE (EBI UK)
5 uniprot        UNIPROT PROTOTYPE (EBI UK)
```

```
> ##Get the current ensembl database.
> ensembl = useMart("ensembl")
> ##List the datasets therein
> listDatasets(ensembl)[1:10,]
```

```
                   dataset                            description   version
1   oanatinus_gene_ensembl Ornithorhynchus anatinus genes (OANA5)     OANA5
2  cporcellus_gene_ensembl      Cavia porcellus genes (GUINEAPIG) GUINEAPIG
3  gaculeatus_gene_ensembl Gasterosteus aculeatus genes (BROADS1)   BROADS1
4   lafricana_gene_ensembl     Loxodonta africana genes (BROADE1)   BROADE1
5    agambiae_gene_ensembl       Anopheles gambiae genes (AgamP3)    AgamP3
6  mlucifugus_gene_ensembl     Myotis lucifugus genes (MICROBAT1) MICROBAT1
7    hsapiens_gene_ensembl            Homo sapiens genes (NCBI36)    NCBI36
8    aaegypti_gene_ensembl           Aedes aegypti genes (AaegL1)    AaegL1
9   csavignyi_gene_ensembl         Ciona savignyi genes (CSAV2.0)   CSAV2.0
10     fcatus_gene_ensembl                Felis catus genes (CAT)       CAT
```

```
> ##Then set up so that you use that for this session
> ##(we will choose the mouse one from NCBI build 37.1):
> ensembl = useDataset("mmusculus_gene_ensembl",mart=ensembl)
> ##List attributes
> attributes = listAttributes(ensembl)
> attributes[1:10,]
```

```
                    name    description
1           affy_mg_u74a   Affy mg u74a
2         affy_mg_u74av2 Affy mg u74av2
3           affy_mg_u74b   Affy mg u74b
4         affy_mg_u74bv2 Affy mg u74bv2
5           affy_mg_u74c   Affy mg u74c
6         affy_mg_u74cv2 Affy mg u74cv2
7           affy_moe430a   Affy moe430a
8           affy_moe430b   Affy moe430b
9    affy_moex_1_0_st_v1      AFFY MoEx
10 affy_mogene_1_0_st_v1    AFFY MoGene
```

```
> ##And filters
> filters = listFilters(ensembl)
> filters[1:10,]
```

```
               name            description
1      affy_mg_u74a     Affy mg u74a ID(s)
```

```
2    affy_mg_u74av2   Affy mg u74av2 ID(s)
3      affy_mg_u74b     Affy mg u74b ID(s)
4    affy_mg_u74bv2   Affy mg u74bv2 ID(s)
5      affy_mg_u74c     Affy mg u74c ID(s)
6    affy_mg_u74cv2   Affy mg u74cv2 ID(s)
7      affy_moe430b     Affy moe430b ID(s)
8   affy_mouse430_2  Affy mouse430 2 ID(s)
9  affy_mouse430a_2 Affy mouse430a 2 ID(s)
10   affy_mu11ksuba   Affy mu11ksuba ID(s)
```

```
> ##Some entrez gene IDs
> EGs = c("18392","18414","56513")
> ##1st a Simple example to just get some gene names:
> getBM(attributes = "external_gene_id",
+       filters = "entrezgene",
+       values = EGs,
+       mart=ensembl)
```

```
  external_gene_id
1            Orc1l
2             Osmr
3           Pard6a
```

```
> ##Transcript starts and ends:
> getBM(attributes = c("entrezgene","transcript_start","transcript_end"),
+       filters = "entrezgene",
+       values = EGs,
+       mart=ensembl)
```

```
  entrezgene transcript_start transcript_end
1      18392        108252066      108288633
2      18414          6763590        6824283
3      56513        108225054      108227393
4      56513        108225571      108227393
5      56513        108225571      108227262
```

```
> ##Additionally, you can get exon boundaries.
> ##But 1st you have to find out what the attributes are called...
> attributeSummary(ensembl)
```

```
     category                          group
1    Features                      EXTERNAL:
```

```
2    Features                         GENE:
3    Features                      PROTEIN:
4    Homologs             AEDES ORTHOLOGS:
5    Homologs         ANOPHELES ORTHOLOGS:
6    Homologs         ARMADILLO ORTHOLOGS:
7    Homologs          BUSHBABY ORTHOLOGS:
8    Homologs               CAT ORTHOLOGS:
9    Homologs           CHICKEN ORTHOLOGS:
10   Homologs             CHIMP ORTHOLOGS:
11   Homologs CIONA INTESTINALIS ORTHOLOGS:
12   Homologs     CIONA SAVIGNYI ORTHOLOGS:
13   Homologs       COMMON SHREW ORTHOLOGS:
14   Homologs               COW ORTHOLOGS:
15   Homologs               DOG ORTHOLOGS:
16   Homologs        DROSOPHILA ORTHOLOGS:
17   Homologs         C.ELEGANS ORTHOLOGS:
18   Homologs          ELEPHANT ORTHOLOGS:
19   Homologs              FUGU ORTHOLOGS:
20   Homologs       STICKLEBACK ORTHOLOGS:
21   Homologs        GUINEA PIG ORTHOLOGS:
22   Homologs          HEDGEHOG ORTHOLOGS:
23   Homologs                         GENE:
24   Homologs              HORSE ORTHOLOGS
25   Homologs             HUMAN ORTHOLOGS:
26   Homologs            MEDAKA ORTHOLOGS:
27   Homologs          MICROBAT ORTHOLOGS:
28   Homologs         MOUSELEMUR ORTHOLOGS
29   Homologs                     PARALOGS:
30   Homologs           OPOSSUM ORTHOLOGS:
31   Homologs          ORANGUTAN ORTHOLOGS
32   Homologs               PIKA ORTHOLOGS
33   Homologs          PLATYPUS ORTHOLOGS:
34   Homologs            RABBIT ORTHOLOGS:
35   Homologs               RAT ORTHOLOGS:
36   Homologs            RHESUS ORTHOLOGS:
37   Homologs          SQUIRREL ORTHOLOGS:
38   Homologs            TENREC ORTHOLOGS:
39   Homologs         TETRAODON ORTHOLOGS:
40   Homologs        TREE SHREW ORTHOLOGS:
41   Homologs           XENOPUS ORTHOLOGS:
```

```
42   Homologs              YEAST ORTHOLOGS:
43   Homologs          ZEBRAFISH ORTHOLOGS:
44  Sequences                    SEQUENCES:
45  Sequences            Header Information
46       SNPs         GENE ASSOCIATED SNPS:
47       SNPs                         GENE:
48 Structures                         EXON:
49 Structures                         GENE:

> ##Lets zoom in on these exon/Structure attributes
> listAttributes(ensembl, category = "Structures", group = "EXON:")

              name             description
1  ensembl_exon_id         Ensembl Exon ID
2   exon_chrom_end       Exon Chr End (bp)
3 exon_chrom_start     Exon Chr Start (bp)
4            phase                   phase
5             rank Exon Rank in Transcript

> ##Find the exon starts and stops for "56513"
> getBM(attributes = c("ensembl_exon_id","exon_chrom_start","exon_chrom_end"),
+       filters = "entrezgene",
+       values = "56513",
+       mart=ensembl)

> ##We can also search based on GO terms
> library(GO.db)
> GOTERM[["GO:0016564"]]

GOID: GO:0016564
Term: transcription repressor activity
Ontology: MF
Definition: Any transcription regulator activity that prevents or
    downregulates transcription.
Synonym: negative transcriptional regulator activity
Synonym: transcriptional repressor activity

> ##here is what we have for EGs affiliated with that term
> GOEGs = unique(org.Mm.egGO2EG[["GO:0016564"]])
> GOEGs
```

```
  [1] "11614"     "11770"     "11906"     "11910"     "12029"     "12053"
  [7] "12151"     "12265"     "12395"     "13047"     "13048"     "13163"
 [13] "13345"     "13433"     "15110"     "15184"     "15205"     "15242"
 [19] "15404"     "15412"     "15426"     "16468"     "16600"     "16969"
 [25] "17257"     "17425"     "17701"     "17859"     "17936"     "17937"
 [31] "17978"     "18037"     "18091"     "18171"     "18432"     "18507"
 [37] "19015"     "19016"     "19401"     "19645"     "19712"     "19763"
 [43] "19821"     "20185"     "20218"     "20230"     "20371"     "20465"
 [49] "20473"     "20602"     "20893"     "21385"     "21386"     "21833"
 [55] "21834"     "21849"     "21907"     "22025"     "22778"     "22781"
 [61] "23942"     "23950"     "24136"     "27049"     "29871"     "52679"
 [67] "53975"     "54427"     "56218"     "56233"     "56381"     "56461"
 [73] "57741"     "58805"     "59058"     "66935"     "67824"     "71041"
 [79] "72567"     "74120"     "74123"     "74318"     "79221"     "81703"
 [85] "83925"     "84653"     "93759"     "108655"    "110521"    "110805"
 [91] "114142"    "114712"    "140477"    "208727"    "216161"    "231004"
 [97] "231798"    "234219"    "237412"    "240690"    "245688"    "329416"
[103] "330627"    "382867"    "100009600"
```

```
> ##Then we can retrieve these from biomaRt like this:
> geneLocs <- getBM(c("ensembl_gene_id", "transcript_start",
+          "transcript_end", "chromosome_name"), "entrezgene",
+           GOEGs, mart=ensembl)
```

4 Session Information

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

```
R version 2.9.0 Under development (unstable) (2008-10-20 r46762)
x86_64-unknown-linux-gnu

locale:
LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONET

attached base packages:
[1] tools     stats     graphics  grDevices datasets  utils     methods
[8] base

other attached packages:
```

```
[1] GO.db_2.2.5         biomaRt_1.17.0      org.Mm.eg.db_2.2.6
[4] RSQLite_0.7-1       DBI_0.2-4           AnnotationDbi_1.5.2
[7] Biobase_2.3.1

loaded via a namespace (and not attached):
[1] RCurl_0.91-0 XML_1.96-0
```