Database mining with biomaRt

Steffen Durinck[‡*]

July 26, 2009

Abstract

Comprehensive analysis of data generated from high-throughput biological experiments, involves integration of a variety of information that can be retrieved from public databases. A simple example is to annotate a set of features that are found differentially expressed in a microarray experiment with corresponding gene symbols and genomic locations. Most public databases provide access to their data via web browsers. However, a major remaining bioinformatics challenge is how to efficiently have access to this biological data from within a data analysis environment. BioMart is a generic, query oriented data management system, capable of integrating distributed data resources. It is developed at the European Bioinformatics Institute (EBI) and Cold Spring Harbour Laboratory (CSHL). biomaRt is a software package aimed at integrating data from BioMart systems into R, enabling biological database mining.

1 Public BioMart databases

Loading the library

```
> options(width = 100)
> library(biomaRt)
```

Display the available public BioMart databases:

```
> listMarts()
```

[*]sdurinck@gmail.com

```
                    biomart                                                version
1                   ensembl                           ENSEMBL 55 GENES (SANGER UK)
2                       snp                       ENSEMBL 55 VARIATION  (SANGER UK)
3       functional_genomics               ENSEMBL 55 FUNCTIONAL GENOMICS (SANGER UK)
4                       vega                                    VEGA 35  (SANGER UK)
5                       msd                                 MSD PROTOTYPE (EBI UK)
6                      htgt HIGH THROUGHPUT GENE TARGETING AND TRAPPING (SANGER UK)
7                   QTL_MART                               GRAMENE 29 QTL DB (CSHL US)
8       ENSEMBL_MART_ENSEMBL                                GRAMENE 29 GENES (CSHL US)
9           ENSEMBL_MART_SNP                                  GRAMENE 29 SNPs (CSHL US)
10        GRAMENE_MARKER_29                               GRAMENE 29 MARKERS (CSHL US)
11           GRAMENE_MAP_29                             GRAMENE 29 MAPPINGS (CSHL US)
12                 REACTOME                                        REACTOME (CSHL US)
13         wormbase_current                                    WORMBASE (CSHL US)
14                    dicty                           DICTYBASE (NORTHWESTERN US)
15                rgd__mart                                       RGD GENES (MCW US)
16            ipi_rat__mart                                  RGD IPI MART (MCW US)
17               SSLP__mart                       RGD MICROSATELLITE MARKERS (MCW US)
18                 g4public                                           HGNC (EBI UK)
19                    pride                                         PRIDE (EBI UK)
20              intermart-1                                       INTERPRO (EBI UK)
21             uniprot_mart                                        UNIPROT (EBI UK)
22 ensembl_expressionmart_48                                     EURATMART (EBI UK)
23         bacterial_mart_52                    ENSEMBL BACTERIA 52 GENES (EBI UK)
24           metazoa_mart_52                    ENSEMBL METAZOA  52 GENES (EBI UK)
25           protist_mart_52                    ENSEMBL PROTISTS 52 GENES (EBI UK)
26                 biomartDB                       PARAMECIUM GENOME (CNRS FRANCE)
27        Eurexpress Biomart                         EUREXPRESS (MRC EDINBURGH UK)
28      pepseekerGOLD_mart06                PEPSEEKER (UNIVERSITY OF MANCHESTER UK)
29     Pancreatic_Expression PANCREATIC EXPRESSION DATABASE (INSTITUTE OF CANCER UK)
```

Select a BioMart database to use:

```
> ensembl = useMart("ensembl")
```

2 Datasets

A BioMart database usually contains multiple datasets. The listDatasets function will retrieve all available datasets present in a particular BioMart.

```
> datasets = listDatasets(ensembl)
> datasets[1, ]
```

```
                dataset                              description version
1 oanatinus_gene_ensembl Ornithorhynchus anatinus genes (OANA5)   OANA5
```

Finaly we can select the BioMart and dataset we want to use.

```
> ensembl = useMart("ensembl", dataset = "hsapiens_gene_ensembl")

Checking attributes ... ok
Checking filters ... ok
```

3 Attributes and Filters

The listAttributes function retrieves all available attributes of a dataset.

```
> attributes = listAttributes(ensembl)
```

Similarly we can retrieve all available filters:

```
> filters = listFilters(ensembl)
```

4 Ensembl

4.1 Task 1a: Annotate set of affy ids with gene symbol and chromosomal location

In this first task we will annotate the following probes (”211550_at”,”202431_s_at”,”206044_s_at”) of the Affymetrix U133plus2 platform with gene symbol and chromosomal location.

```
> affyids = c("211550_at", "202431_s_at", "206044_s_at")
> annotation = getBM(c("affy_hg_u133_plus_2", "ensembl_gene_id", "hgnc_symbol",
+     "chromosome_name", "start_position", "end_position", "band", "strand"),
+     filters = "affy_hg_u133_plus_2", values = affyids, mart = ensembl)
> print(annotation)

  affy_hg_u133_plus_2 ensembl_gene_id hgnc_symbol chromosome_name start_position end_position
1         202431_s_at ENSG00000136997         MYC               8      128748316    128753671
2         206044_s_at ENSG00000157764        BRAF               7      140433817    140624564
3           211550_at ENSG00000146648        EGFR               7       55086714     55324313
    band strand
1 q24.21      1
2    q34     -1
3  p11.2      1
```

4.2 Task 1b: annotate set of Illumina humanwg_6_v2 bead chips with GO biological process identifiers

```
> illuminaIDs = c("ILMN_1728071", "ILMN_1662668")
> goAnnot = getBM(c("illumina_humanwg_6_v2", "go_biological_process_id", "go_biologic
+     filters = "illumina_humanwg_6_v2", values = illuminaIDs, mart = ensembl)
> print(goAnnot[1:5, ])
```

```
  illumina_humanwg_6_v2 go_biological_process_id go_biological_process_linkage_type
1          ILMN_1662668               GO:0000281                                 IMP
2          ILMN_1662668               GO:0006461                                 IDA
3          ILMN_1662668               GO:0006974                                 IDA
4          ILMN_1662668               GO:0007026                                 IDA
5          ILMN_1662668               GO:0007050                                 IDA
```

4.3 Task 2: Retrieve all genes that are involved in Diabetes Mellitus type I or II and have transcription factor activity

```
> diab = getBM(c("ensembl_gene_id", "hgnc_symbol"), filters = c("mim_morbid_accession
+     "go"), values = list(c("125853", "222100"), "GO:0003700"), mart = ensembl)
> print(diab)
```

```
  ensembl_gene_id hgnc_symbol
1 ENSG00000139515        PDX1
2 ENSG00000108753       HNF1B
3 ENSG00000148737      TCF7L2
4 ENSG00000106331        PAX4
5 ENSG00000162992     NEUROD1
6 ENSG00000135100       HNF1A
```

4.4 Task 3: Retrieve the chromosomal locations of all miRNAs on chromosome 13

```
> miRNA = getBM(c("mirbase", "ensembl_gene_id", "start_position", "chromosome_name"),
+     filters = c("chromosome_name", "with_mirbase"), values = list(13, TRUE),
+     mart = ensembl)
> miRNA[1:5, ]
```

```
    mirbase ensembl_gene_id start_position chromosome_name
1 MI0008190 ENSG00000211491       41301964              13
2 MI0003635 ENSG00000207652       41384902              13
3 MI0000070 ENSG00000208006       50623109              13
4 MI0000069 ENSG00000207718       50623255              13
5 MI0003636 ENSG00000207858       90883436              13
```

4.5 Task 4: Retrieve all entrezgene ids on chromosome 22 that have a non synonymous coding SNP

Note that you'll have to use the snptype_filters filter and that possible values for filters can be displayed using the filterOptions function.

```
> filterOptions("snptype_filters", ensembl)

[1] "[STOP_GAINED,STOP_LOST,COMPLEX_INDEL,FRAMESHIFT_CODING,NON_SYNONYMOUS_CODING,STOP_GAINED,SPLICE_SITE,STOP_LOST

> entrez = getBM("entrezgene", filters = c("chromosome_name", "snptype_filters"),
+     values = list(22, "NON_SYNONYMOUS_CODING"), mart = ensembl)
> entrez[1:5, ]

[1]  23784  81061 150160 150165 128954
```

4.6 Task 5: Retrieve all exonic sequences from the CDH1 gene

```
> seq = getSequence(id = "CDH1", type = "hgnc_symbol", seqType = "gene_exon",
+     mart = ensembl)
> seq[1, ]


1 ATATCGGATTTGGAGAGACACTGCCAACTGGCTGGAGATTAATCCGGACACTGGTGCCATTTCCACTCGGGCTGAGCTGGACAGGGAGGATTTTGAGCACGTGAAGAACAGCA
  hgnc_symbol
1        CDH1
```

4.7 Task 6: Retrieve 2000bp sequence upstream of APC and CUL1 translation start site

```
> promoter = getSequence(id = c("APC", "CUL1"), type = "hgnc_symbol", seqType = "coding_gene_flank",
+     upstream = 2000, mart = ensembl)
```

4.8 Task 7: Retrieve human gene symbol and affy identifiers of their homologs in chicken for the following two identifiers from the human affy_hg_u95av2 platform: 1434_at, 1888_s_at

```
> human = useMart("ensembl", dataset = "hsapiens_gene_ensembl")

Checking attributes ... ok
Checking filters ... ok

> chicken = useMart("ensembl", dataset = "ggallus_gene_ensembl")

Checking attributes ... ok
Checking filters ... ok
```

```
> out = getLDS(attributes = c("affy_hg_u95av2", "hgnc_symbol"), filters = "affy_hg_u95av2",
+     values = c("1888_s_at", "1434_at"), mart = human, attributesL = "affy_chicken",
+     martL = chicken)
> out

        V1   V2                  V3
1   1434_at PTEN GgaAffx.25913.1.S1_a
2 1888_s_at  KIT      Gga.606.1.S1_at
```

5 Variation BioMart

5.1 Task 8: Retrieve all refsnp id's and their alleles and position that are located on chromosome 8 and between bp 148350 and 158612

```
> snp = useMart("snp", dataset = "hsapiens_snp")

Checking attributes ... ok
Checking filters ... ok

> out = getBM(attributes = c("refsnp_id", "allele", "chrom_start"), filters = c("chr_name",
+     "chrom_start", "chrom_end"), values = list(8, 148350, 158612), mart = snp)
> out[1:5, ]

     refsnp_id allele chrom_start
1 ENSSNP4490669    C/G      148729
2 ENSSNP5558526    T/C      148909
3 ENSSNP4089737    T/A      149060
4 ENSSNP9060169    C/T      149245
5 ENSSNP4351891    C/G      149250
```

6 Ensembl archives

```
> listMarts(archive = TRUE)
```

	biomart	version
1	ensembl_mart_51	Ensembl 51
2	snp_mart_51	SNP 51
3	vega_mart_51	Vega 32
4	ensembl_mart_50	Ensembl 50
5	snp_mart_50	SNP 50
6	vega_mart_50	Vega 32
7	ensembl_mart_49	ENSEMBL GENES 49 (SANGER)
8	genomic_features_mart_49	Genomic Features
9	snp_mart_49	SNP
10	vega_mart_49	Vega

```
11             ensembl_mart_48   ENSEMBL GENES 48 (SANGER)
12    genomic_features_mart_48            Genomic Features
13                snp_mart_48                         SNP
14               vega_mart_48                        Vega
15             ensembl_mart_47   ENSEMBL GENES 47 (SANGER)
16    genomic_features_mart_47            Genomic Features
17                snp_mart_47                         SNP
18               vega_mart_47                        Vega
19    compara_mart_homology_47            Compara homology
20 compara_mart_multiple_ga_47 Compara multiple alignments
21 compara_mart_pairwise_ga_47 Compara pairwise alignments
22             ensembl_mart_46   ENSEMBL GENES 46 (SANGER)
23    genomic_features_mart_46            Genomic Features
24                snp_mart_46                         SNP
25               vega_mart_46                        Vega
26    compara_mart_homology_46            Compara homology
27 compara_mart_multiple_ga_46 Compara multiple alignments
28 compara_mart_pairwise_ga_46 Compara pairwise alignments
29             ensembl_mart_45   ENSEMBL GENES 45 (SANGER)
30                snp_mart_45                         SNP
31               vega_mart_45                        Vega
32    compara_mart_homology_45            Compara homology
33 compara_mart_multiple_ga_45 Compara multiple alignments
34 compara_mart_pairwise_ga_45 Compara pairwise alignments
35             ensembl_mart_44   ENSEMBL GENES 44 (SANGER)
36                snp_mart_44                         SNP
37               vega_mart_44                        Vega
38    compara_mart_homology_44            Compara homology
39 compara_mart_pairwise_ga_44 Compara pairwise alignments
40             ensembl_mart_43   ENSEMBL GENES 43 (SANGER)
41                snp_mart_43                         SNP
42               vega_mart_43                        Vega
43    compara_mart_homology_43            Compara homology
44 compara_mart_pairwise_ga_43 Compara pairwise alignments
```

```
> listMarts(host = "may2009.archive.ensembl.org/biomart/martservice/")
```

```
               biomart             version
1 ENSEMBL_MART_ENSEMBL           Ensembl 54
2     ENSEMBL_MART_SNP Ensembl Variation 54
```

```
3    ENSEMBL_MART_VEGA              Vega 35
4              REACTOME   Reactome(CSHL US)
5     wormbase_current   WormBase (CSHL US)
6                 pride       PRIDE (EBI UK)
```

7 SessionInfo

```
> sessionInfo()

R version 2.9.0 (2009-04-17)
powerpc-apple-darwin8.11.1

locale:
C

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

other attached packages:
[1] biomaRt_2.0.0

loaded via a namespace (and not attached):
[1] RCurl_0.94-1 XML_2.3-0
```