Description of the biomaRt package

Steffen Durinck[‡][*], Wolfgang Huber[¶][†],
Yves Moreau[‡], Bart De Moor[‡]

February 26, 2005

[‡]Department of Electronical Engineering, ESAT-SCD, K.U.Leuven,
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium,
`http://www.esat.kuleuven.ac.be/~dna/BioI`
and [¶]European Bioinformatics Institute, Hinxton, UK

Contents

[*] Steffen.Durinck@esat.kuleuven.ac.be
[†] huber@ebi.ac.uk

1 Introduction

The BioConductor *biomaRt* package enables to directly query databases based on biomaRt such as Ensembl, a software system which produces and maintains automatic annotation on metazoan genomes. This way you can annotate the features on your array with the latest annotations starting from identifiers such as affy id's, locuslink, RefSeq and more. Annotation includes gene names, GO and OMIM annotation (depending on species).

2 objects

2.1 Mart-class

An object of the `Mart` class represents a connection to a BioMart. And has the following slots:

- `connection`: stores the RMySQLConnection to a BioMart
- `mart`: contains the name of the mart

2.2 martTable-class

An object of the `martTable` class represents annotation of a gene. And has the following slots:

- `id`: stores the id used for querying
- `table`: is a list of lists (or vectors) storing annotation information

2.3 Gene-class

An object of the `Gene` class represents annotation of a gene. And has the following slots:

- `id`: stores the id used for querying
- `martID`: stores the mart specific id
- `symbol`: stores the gene symbol
- `chromosome`: stores the chromosome number on which the gene is locatlized
- `band`: stores the band on which the gene is localized
- `start`: stores the start position of the gene on the respective chromosome
- `end`: stores the end position of the gene on the respective chromosome
- `GO`: a slot to store an object of class GO
- `OMIM`: a slot to store an object of class OMIM

2.4 MultiGene-class

When starting from a query id, Ensembl retrieves multiple hits, all these different hits can be storen in an object of the class `MultiGene`. This class is identical to the `Gene` class except that here we store multiple elements in each slot, e.g. multiple symbols matching a query id.

2.5 GO-class

An object of the `GO` class represents GO annotation of a gene. And has the following slots:

- `id`: stores the id used for querying
- `GOID`: stores the GO id's associated with the query id.
- `description`: stores the description of the corresponding GO id's
- `evidence`: stores the evidence code of GO annotation

2.6 OMIM-class

An object of the `OMIM` class represents OMIM annotation of a gene. And has the following slots:

- `id`: stores the id used for querying
- `OMIMID`: stores the OMIM id's associated with the query id.
- `disease`: stores the description of the corresponding OMIM id's

3 Functions

3.1 `martConnect`

A first step in using the biomaRt package is to connect to a BioMart database. The function martConnect establishes a connection with one of the following BioMart databases: snp, ensembl and vega.

Examples:

```
> library(biomaRt)

Loading required package: Biobase
Welcome to Bioconductor
         Vignettes contain introductory material.  To view,
         simply type: openVignette()
         For details on reading vignettes, see
         the openVignette help page.
Loading required package: RMySQL
Loading required package: DBI
```

```
> mart <- martConnect("ensembl")

-  Connected to Ensembl@EBI -
```

3.2 martDisconnect

You can only hold a limited number of connections with different BioMarts. The function martDisconnect can be used to close a mart connection.

Examples:

```
> martDisconnect(mart)

[1] TRUE
```

3.3 getGene

The function `getGene` uses a query id to look up identification and chromosomal information of the corresponding gene. Depending on the selected output, this function returns a `martTable` or an object of class `Gene`. When no information about the identifier is found in Ensembl, an empty `Gene` object will be created. If however multiple genes match a certain identifier, then an object of class `MultiGene` will be return containing information of all matches. Currently the `getGene` function takes identifiers from locuslink, affy, RefSeq and embl. Besides the id argument, this function also has a species, array and type argument.
The id argument is either a vector of identifiers or a single identifier to be annotated.
The species argument should have the species from which the identifier comes as value. For the value of species, we use the full name of the species where separate words are separated by an underscore, e.g. 'gallusgallus'. A list of possible species to choose from can be obtained by executing the function `getSpecies`.
The array argument takes affy array identifiers as values. A list of possible identifiers supported by the package can be obtained by executing the function `getAffyArrays`.
The mart argument is a mart connection, which was obtained using the method `martConnect`
The type takes the values of 'locuslink','refseq' and 'embl' to clarify which type of identifier is specified in the id argument.
The output can be changed using the output argument. One can choose between a martTable (default) and an output of Gene/Multi-Gene objects. Depending on the identifier, different additional arguments will have to be given, summarized below:

- Affy id's: id, array, mart
- Locuslink: id, type, species, mart
- RefSeq: id, type, species, mart
- embl: id, type, species, mart

Examples:

```
> library(biomaRt)
> mart <- martConnect("ensembl")

-  Connected to Ensembl@EBI -

> getGene(id = "1939_at", array = "hgu95av2", mart = mart)

An object of class "martTable"
Slot "id":
[1] "1939_at"

Slot "table":
$symbol
[1] "TP53"

$description
[1] "Cellular tumor antigen p53 (Tumor suppressor p53) (Phosphoprotein p53) (Antigen NY-

$band
[1] "p13.1"

$chromosome
[1] "17"

$start
[1] 7512464

$end
[1] 7531642

$martID
[1] "ENSG00000141510"

> getGene(id = 672, type = "locuslink", species = "homo_sapiens",
+     mart = mart)

An object of class "martTable"
Slot "id":
[1] "672"

Slot "table":
$symbol
```

```
[1] "BRCA1"

$description
[1] "Breast cancer type 1 susceptibility protein. [Source:Uniprot/SWISSPROT;Acc:P38398]"

$band
[1] "q21.31"

$chromosome
[1] "17"

$start
[1] 38449844

$end
[1] 38530934

$martID
[1] "ENSG00000012048"
```

3.4 getGO

The function getGO uses a query id to look up GO annotation of the corresponding gene. It return an object of class GO. When no information about the identifier is found in Ensembl, an empty GO object will be created. Currently the getGO function takes identifiers from locuslink, affy, RefSeq and embl. Besides the id argument, this function also has a species, array and type argument.
The id argument is either a vector of identifiers or a single identifier to be annotated.
The species argument should have the species from which the identifier comes as value. A list of possible species to choose from can be obtained by executing the function getSpecies.
The array argument takes affy array identifiers as values. A list of possible identifiers supported by the package can be obtained by executing the function getAffyArrays.
The mart argument is a mart connection, which was obtained using the method martConnect
A last argument of this function is the type argument which, takes the values of 'locuslink','refseq' and 'embl' to clarify which type of identifier is specified in the id argument. Depending on the identifier, different additional arguments will have to be given, summarized below:

- Affy id's: id, array, mart
- Locuslink: id, type, species, mart
- RefSeq: id, type, species, mart
- embl: id, type, species, mart

Examples:

```
> getGO(id = "1939_at", array = "hgu95av2", mart = mart)

An object of class "GO"
Slot "GOID":
 [1] "GO:0000739" "GO:0003700" "GO:0004518" "GO:0005507" "GO:0005524"
 [6] "GO:0008270" "GO:0051074" "GO:0000075" "GO:0006284" "GO:0006289"
[11] "GO:0006310" "GO:0006355" "GO:0006915" "GO:0007050" "GO:0007569"
[16] "GO:0008283" "GO:0008628" "GO:0008630" "GO:0008635" "GO:0030154"
[21] "GO:0030308" "GO:0045786" "GO:0046902" "GO:0051097" "GO:0005730"
[26] "GO:0005739" NA

Slot "description":
 [1] "DNA strand annealing activity"
 [2] "transcription factor activity"
 [3] "nuclease activity"
 [4] "copper ion binding"
 [5] "ATP binding"
 [6] "zinc ion binding"
 [7] "protein tetramerization activity"
 [8] "cell cycle checkpoint"
 [9] "base-excision repair"
[10] "nucleotide-excision repair"
[11] "DNA recombination"
[12] "regulation of transcription, DNA-dependent"
[13] "apoptosis"
[14] "cell cycle arrest"
[15] "cell aging"
[16] "cell proliferation"
[17] "induction of apoptosis by hormones"
[18] "DNA damage response, signal transduction resulting in induction of apoptosis"
[19] "caspase activation via cytochrome c"
[20] "cell differentiation"
[21] "negative regulation of cell growth"
[22] "negative regulation of cell cycle"
[23] "regulation of mitochondrial membrane permeability"
[24] "negative regulation of helicase activity"
[25] "nucleolus"
[26] "mitochondrion"
[27] NA

Slot "evidence":
 [1] "IDA" "IDA" "TAS" "IDA" "IDA" "TAS" "TAS" "TAS" "TAS" "IMP" "TAS" "IDA"
```

```
[13] "IDA" "TAS" "IMP" "TAS" "TAS" "TAS" "IDA" "TAS" "IMP" "IEA" "TAS" "TAS"
[25] "IDA" "IDA" NA
```

```
> getGO(id = 672, type = "locuslink", species = "homo_sapiens",
+     mart = mart)
```

```
An object of class "GO"
Slot "GOID":
 [1] "GO:0003684" "GO:0003713" "GO:0004842" "GO:0005515" "GO:0008270"
 [6] "GO:0015631" "GO:0016563" "GO:0006357" "GO:0006359" "GO:0006978"
[11] "GO:0016567" "GO:0042127" "GO:0042981" "GO:0045739" "GO:0045786"
[16] "GO:0046600" "GO:0000151" "GO:0005615" "GO:0005634" "GO:0005667"
[21] "GO:0008274" "GO:0005622" "GO:0004842" "GO:0008270" "GO:0016567"
[26] "GO:0000151" "GO:0005622" "GO:0004842" "GO:0008270" "GO:0016567"
[31] "GO:0000151" "GO:0005622" "GO:0004842" "GO:0008270" "GO:0016567"
[36] "GO:0000151" "GO:0005622" "GO:0004842" "GO:0008270" "GO:0016567"
[41] "GO:0000151" "GO:0004842" "GO:0008270" "GO:0016567" "GO:0000151"
[46] "GO:0004842" "GO:0008270" "GO:0016567" "GO:0000151" "GO:0004842"
[51] "GO:0008270" "GO:0016567" "GO:0000151" "GO:0003684" "GO:0003713"
[56] "GO:0004842" "GO:0005515" "GO:0008270" "GO:0015631" "GO:0016563"
[61] "GO:0006357" "GO:0006359" "GO:0006978" "GO:0016567" "GO:0042127"
[66] "GO:0042981" "GO:0045739" "GO:0045786" "GO:0046600" "GO:0000151"
[71] "GO:0005615" "GO:0005634" "GO:0005667" "GO:0008274" "GO:0005622"
[76] NA

Slot "description":
 [1] "damaged DNA binding"
 [2] "transcription coactivator activity"
 [3] "ubiquitin-protein ligase activity"
 [4] "protein binding"
 [5] "zinc ion binding"
 [6] "tubulin binding"
 [7] "transcriptional activator activity"
 [8] "regulation of transcription from Pol II promoter"
 [9] "regulation of transcription from Pol III promoter"
[10] "DNA damage response, signal transduction by p53 class mediator resulting in transc
[11] "protein ubiquitination"
[12] "regulation of cell proliferation"
[13] "regulation of apoptosis"
[14] "positive regulation of DNA repair"
[15] "negative regulation of cell cycle"
[16] "negative regulation of centriole replication"
[17] "ubiquitin ligase complex"
[18] "extracellular space"
[19] "nucleus"
```

```
[20] "transcription factor complex"
[21] "gamma-tubulin ring complex"
[22] "intracellular"
[23] "ubiquitin-protein ligase activity"
[24] "zinc ion binding"
[25] "protein ubiquitination"
[26] "ubiquitin ligase complex"
[27] "intracellular"
[28] "ubiquitin-protein ligase activity"
[29] "zinc ion binding"
[30] "protein ubiquitination"
[31] "ubiquitin ligase complex"
[32] "intracellular"
[33] "ubiquitin-protein ligase activity"
[34] "zinc ion binding"
[35] "protein ubiquitination"
[36] "ubiquitin ligase complex"
[37] "intracellular"
[38] "ubiquitin-protein ligase activity"
[39] "zinc ion binding"
[40] "protein ubiquitination"
[41] "ubiquitin ligase complex"
[42] "ubiquitin-protein ligase activity"
[43] "zinc ion binding"
[44] "protein ubiquitination"
[45] "ubiquitin ligase complex"
[46] "ubiquitin-protein ligase activity"
[47] "zinc ion binding"
[48] "protein ubiquitination"
[49] "ubiquitin ligase complex"
[50] "ubiquitin-protein ligase activity"
[51] "zinc ion binding"
[52] "protein ubiquitination"
[53] "ubiquitin ligase complex"
[54] "damaged DNA binding"
[55] "transcription coactivator activity"
[56] "ubiquitin-protein ligase activity"
[57] "protein binding"
[58] "zinc ion binding"
[59] "tubulin binding"
[60] "transcriptional activator activity"
[61] "regulation of transcription from Pol II promoter"
[62] "regulation of transcription from Pol III promoter"
[63] "DNA damage response, signal transduction by p53 class mediator resulting in transc
```

```
[64] "protein ubiquitination"
[65] "regulation of cell proliferation"
[66] "regulation of apoptosis"
[67] "positive regulation of DNA repair"
[68] "negative regulation of cell cycle"
[69] "negative regulation of centriole replication"
[70] "ubiquitin ligase complex"
[71] "extracellular space"
[72] "nucleus"
[73] "transcription factor complex"
[74] "gamma-tubulin ring complex"
[75] "intracellular"
[76] NA

Slot "evidence":
 [1] "NR"  "TAS" "IEA" "IPI" "TAS" "NAS" "TAS" "TAS" "TAS" "TAS" "IEA" "TAS"
[13] "TAS" "NAS" "IEA" "NAS" "IEA" "TAS" "TAS" "TAS" "NAS" "IEA" "IEA" "IEA"
[25] "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA"
[37] "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA" "IEA"
[49] "IEA" "IEA" "IEA" "IEA" "IEA" "NR"  "TAS" "IEA" "IPI" "TAS" "NAS" "TAS"
[61] "TAS" "TAS" "TAS" "IEA" "TAS" "TAS" "NAS" "IEA" "NAS" "IEA" "TAS" "TAS"
[73] "TAS" "NAS" "IEA" NA
```

3.5 getOMIM

The function getOMIM uses a query id to look up OMIM annotation of the corresponding gene. It return an object of class OMIM. When no information about the identifier is found in Ensembl, an empty OMIM object will be created. Currently the getOMIM function takes identifiers from locuslink, affy, RefSeq and embl. Besides the id argument, this function also has an array ,type and mart argument.
The id argument is either a vector of identifiers or a single identifier to be annotated.
The array argument takes affy array identifiers as values. A list of possible identifiers supported by the package can be obtained by executing the function getAffyArrays.
The type argument takes the values of 'locuslink','refseq' and 'embl' to clarify which type of identifier is specified in the id argument. If the argumant array is used then biomaRt knows the identifiers given corresponf to affy id's. The mart argument is a mart connection, which was obtained using the method martConnect
Depending on the identifier, different additional arguments will have to be given, summarized below:

- Affy id's: id, array, mart
- Locuslink: id, type, mart
- RefSeq: id, type, mart

- embl: id, type, mart

Examples:

```
> getOMIM(id = "1939_at", array = "hgu95av2", mart = mart)

An object of class "OMIM"
Slot "OMIMID":
[1] 191170 191170

Slot "disease":
[1] "Colorectal cancer, 114500 (3)" "Li-Fraumeni syndrome (3)"

> getOMIM(id = 672, type = "locuslink", mart = mart)

An object of class "OMIM"
Slot "OMIMID":
[1] 113705 113705

Slot "disease":
[1] "Breast cancer-1 (3)" "Ovarian cancer (3)"
```

3.6 getFeature

The function getFeature looks up affy identifiers on a given affy array which correspond to a given symbol. As output this function returns a martTable. Currently the getFeature function takes identifiers from affy only. Besides the symbol argument, this function also has array and mart argument.
The mart argument is a mart connection, which was obtained using the method martConnect
A last argument of this function is the type argument which, takes the values of 'affy', 'locuslink','refseq' and 'embl' to clarify which type of identifier is specified in the id argument.

Examples:

```
> getFeature(symbol = "P53", array = "hgu95av2", mart = mart)

An object of class "martTable"
Slot "id":
[1] "36079_at"  "1939_at"   "1974_s_at" "31618_at"  "1711_at"   "36136_at"
[7] "33749_at"  "1860_at"   "34822_at"

Slot "table":
$symbol
```

```
[1] "TP53I3"  "TP53"    "TP53"    "TP53"    "TP53BP1" "TP53I11" "TP53AP1"
[8] "TP53BP2" "TP53BP2"

$description
[1] "tumor protein p53 inducible protein 3; quinone oxidoreductase homolog; p53-induced
[2] "Cellular tumor antigen p53 (Tumor suppressor p53) (Phosphoprotein p53) (Antigen NY-
[3] "Cellular tumor antigen p53 (Tumor suppressor p53) (Phosphoprotein p53) (Antigen NY-
[4] "Cellular tumor antigen p53 (Tumor suppressor p53) (Phosphoprotein p53) (Antigen NY-
[5] "Tumor suppressor p53-binding protein 1 (p53-binding protein 1) (53BP1). [Source:Uni
[6] "p53-induced protein [Homo sapiens]. [Source:RefSeq;Acc:NM_006034]"
[7] "TP53 activated protein 1; TP53 target gene 1; H_RG012D21.9 [Homo sapiens]. [Source:
[8] "Apoptosis stimulating of p53 protein 2 (Tumor suppressor p53-binding protein 2) (p5
[9] "Apoptosis stimulating of p53 protein 2 (Tumor suppressor p53-binding protein 2) (p5

$martID
[1] "ENSG00000115129" "ENSG00000141510" "ENSG00000141510" "ENSG00000141510"
[5] "ENSG00000067369" "ENSG00000175274" "ENSG00000182165" "ENSG00000143514"
[9] "ENSG00000143514"
```

3.7 getSequence

The function `getSequence` retrieves the sequence given it's chromosome, start and end position. As output this function returns a `martTable`. The mart argument is a mart connection, which was obtained using the method `martConnect` and should in this case be the sequence mart.

Examples:

```
> seqMart <- martConnect(mart = "sequence")

-  Connected to sequence mart @ EBI -

> getSequence(species = "gallus_gallus", chromosome = 1, start = 400,
+     end = 500, mart = seqMart)

An object of class "martTable"
Slot "id":
[1] "1_400_500"

Slot "table":
$chromosome
[1] 1

$start
```

```
[1] 400

$end
[1] 500

$sequence
[1] "GTGACATTTCCAGCATTCAGTGTGTCAAAGCCTAGCTTCATTTTTGAATGTATTGAGGGGCAGATGTCCATCTCATGAATCAT

> martDisconnect(seqMart)

[1] TRUE
```

3.8 getSNP

The function getSNP retrieves all SNP's between a given a start and end position on a gives chromosome.. As output this function returns a martTable. The mart argument is a mart connection, which was obtained using the method martConnect and should in this case be the snp mart.

Examples:

```
> snpMart <- martConnect(mart = "snp")

-  Connected to SNP mart @ EBI -

> getSNP(mart = snpMart, chromosome = 8, start = 148350, end = 159000,
+     species = "homo_sapiens")

An object of class "martTable"
Slot "id":
  [1] "26087865"                   "26087866"
  [3] "26087867"                   NA
  [5] "26087868"                   NA
  [7] "26087869"                   "26021974"
  [9] NA                           NA
 [11] NA                           "24784114"
 [13] "26087849"                   "26087850"
 [15] "26104118"                   "26087851"
 [17] "26104119"                   NA
 [19] "24784113,24784112"          "26104120"
 [21] "26087852"                   "25971294"
 [23] "25891113,25891112,25891111" NA
 [25] NA                           "26104122"
 [27] NA                           "25631888,25631887"
 [29] "26104123"                   "26087854"
 [31] "25631889"                   "25631890"
 [33] "25631891"                   "26087855"
 [35] "26087856"                   "26087857"
 [37] "26104124"                   "26087858"
 [39] "26104125"                   "25420792"
 [41] "25928633,25928632"          "26112982"
 [43] "25928635,25928634"          "25928637,25928636"
 [45] "26112983"                   "25928639,25928638"
 [47] "25917424,25917423"          "25928641,25928640"
 [49] "26104126"                   NA
 [51] NA                           "25928643,25928642"
 [53] "25928645,25928644"          "25928647,25928646"
 [55] "25928649,25928648"          "25928651,25928650"
 [57] "26104115"                   "25928653,25928652"
 [59] "25928655,25928654"          "25928657,25928656"
 [61] "25928659,25928658"          "26021975"
 [63] "26104116"                   "26104117"
 [65] "25928661,25928660"          NA
 [67] "25928663,25928662"          "25928665,25928664"
 [69] "24784110,24784109"          "25928667,25928666"
 [71] "25928669,25928668"          "26087839"
 [73] "25928671,25928670"          "25928673,25928672"
 [75] "25928675,25928674"          "25928677,25928676"
 [77] "25928679,25928678"          "26104104"
 [79] "26104105"                   "26104106"
 [81] "26104107"                   "26104108"
 [83] "26104109"                   "26104110"
 [85] "26104111"                   "26104112"
 [87] "26104113"                   "26104114"
 [89] NA                           "26022513,26022512"
 [91] "26022511,26022510"          "26022509"
 [93] "26087842"                   "26021976"
 [95] "26021977"                   "26127094"
 [97] NA                           "26025531"
 [99] NA                           "26087843"
[101] "26087844"                   NA
[103] NA                           NA
[105] NA                           "26021978"
[107] "26021979"                   "25928681,25928680"
[109] NA                           NA
[111] "25814411"                   "26021980"
[113] "26087838"                   NA
[115] NA                           NA
```

```
[117] NA                            NA
[119] NA                            NA
[121] NA                            NA
[123] NA                            NA
[125] "25632163"                    "26021896"
[127] "26021897"                    NA
[129] "25820705"                    "26087590"
[131] "26087591"                    "26087592"
[133] "25820704"                    NA
[135] "26087593"                    "26087594"
[137] NA                            NA
[139] NA                            NA
[141] NA                            NA
[143] NA                            NA
[145] NA                            NA
[147] NA                            NA
[149] NA                            NA
[151] NA                            NA
[153] NA                            NA
[155] NA                            NA
[157] NA                            NA
[159] NA                            NA

Slot "table":
$snpStart
  [1] 148394 148411 148462 148471 148499 148525 148533 148535 148539 148601
 [11] 148632 148652 148652 148707 148707 148710 148710 148712 148733 148733
 [21] 148745 148774 148803 148803 148828 148839 148855 148863 148865 148867
 [31] 148868 148870 148871 148873 148874 148876 148876 148897 148897 148967
 [41] 148967 148972 149002 149012 149046 149052 149057 149094 149099 149108
 [51] 149130 149151 149155 149173 149179 149187 149188 149197 149220 149233
 [61] 149235 149242 149250 149251 149348 149375 149375 149380 149399 149409
 [71] 149442 149474 149547 149549 149565 149576 149590 149665 149676 149693
 [81] 149694 149706 149731 149753 149811 149825 149830 149949 149951 149951
 [91] 149952 149990 150376 150392 150396 150537 150679 150682 150707 150753
[101] 150760 150771 150797 150798 150826 150862 150923 151010 151010 151062
[111] 151222 151224 151231 151466 151936 151950 151966 152286 152578 152837
[121] 152923 152926 153226 153249 153265 153314 153330 153370 153387 153419
[131] 153420 153421 153432 153438 153461 153486 153550 153654 153784 154050
[141] 154086 154984 155405 155490 155672 155697 155804 155982 156322 156771
[151] 156851 156997 157014 157016 157212 157420 157429 158012 158484 158695

$allele
  [1] "C/A"   "A/G"   "C/T"   "T/G"   "G/A"   "G/A"   "G/A"   "C/T"   "C/T"
 [10] "G/A"   "A/G"   "T/C"   "A/G"   "G/A"   "G/A"   "G/A"   "G/A"   "A/G"  
 [19] "T/G"   "A/C"   "A/G"   "C/A"   "G/A/C" "G/A"   "A/G"   "T/C"   "T/G"  
 [28] "C/T"   "-/TA"  "-/TAG" "A/T"   "G/A"   "T/A"   "A/G"   "C/T"   "-/AC" 
 [37] "-/AC"  "T/C"   "T/C"   "A/G"   "A/G"   "G/C"   "G/C"   "T/C"   "T/C"  
 [46] "C/T"   "A/C"   "C/T"   "T/C"   "G/T"   "G/A"   "A/G"   "C/A"   "G/A"  
 [55] "G/A"   "C/A"   "A/T"   "G/C"   "G/C"   "A/G"   "C/T"   "A/G"   "C/T"  
 [64] "G/A"   "A/G"   "A/G"   "T/C"   "A/C"   "T/C"   "T/G"   "C/A"   "A/G"  
 [73] "A/G"   "A/T"   "C/G"   "G/A"   "G/C"   "C/T"   "G/C"   "C/T"   "A/G"  
 [82] "A/C"   "A/C"   "A/C"   "C/G"   "A/T"   "A/C"   "A/G"   "G/A"   "G/A"  
 [91] "T/C"   "C/T"   "A/G"   "C/G"   "T/C"   "T/C"   "G/A"   "G/C"   "C/T"  
[100] "C/A"   "C/T"   "A/G"   "G/T"   "T/G"   "C/T"   "C/T"   "C/A"   "G/T"  
[109] "G/T"   "C/T"   "G/C"   "A/T"   "C/G"   "C/T"   "A/G"   "T/C"   "A/G"  
[118] "C/G"   "C/T"   "A/C"   "A/C"   "A/C"   "A/G"   "T/C"   "G/A"   "G/C"  
[127] "A/C"   "C/G"   "G/A"   "G/A"   "A/G"   "A/T"   "T/C"   "C/T"   "C/G"  
[136] "-/GAT" "A/G"   "C/T"   "C/G"   "T/A"   "G/C"   "A/G"   "G/T"   "G/A"  
[145] "G/A"   "T/C"   "G/T"   "C/T"   "T/C"   "T/A"   "T/C"   "A/C"   "T/C"  
[154] "T/A"   "C/G"   "A/G"   "A/G"   "A/C"   "T/C"   "G/A"  

$coding
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA NA  1  1  1 NA NA NA NA NA NA NA NA NA NA NA NA
[151] NA NA NA NA NA NA NA NA NA NA

$intronic
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA NA NA NA NA  1  1  1  1  1  1  1  1  1  1  1  1
[151]  1  1  1  1  1  1  1  1  1  1

$syn
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA NA  0  0  1 NA NA NA NA NA NA NA NA NA NA NA NA
[151] NA NA NA NA NA NA NA NA NA NA
```

```
$utr5
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[151] NA NA NA NA NA NA NA NA NA NA

$utr3
  [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [26]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [51]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [76]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[101]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[126]  1  1  1  1  1  1  1  1  1  1 NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[151] NA NA NA NA NA NA NA NA NA NA
```

```
> martDisconnect(snpMart)
```

```
[1] TRUE
```

3.9 getSpecies

The function getSpecies looks up which species are present in the BioMart. This function currently works only for ensembl.

Examples:

```
> getSpecies(mart)
```

```
 [1] "anopheles_gambiae"       "fugu_rubripes"
 [3] "gallus_gallus"           "drosophila_melanogaster"
 [5] "xenopus_tropicalis"      "caenorhabditis_elegans"
 [7] "danio_rerio"             "homo_sapiens"
 [9] "pan_troglodytes"         "mus_musculus"
[11] "apis_mellifera"          "canis_familiaris"
[13] "tetraodon_nigroviridis"  "rattus_norvegicus"
```

3.10 getAffyArrays

The function getAffyArrays retrieves the Affymetrix array identifiers which are present in ensembl and which can be queried using the biomaRt package.

Examples:

```
> getAffyArrays()

            V1                V2
1  hgu133plus2      homo_sapiens
2     hgu133a2      homo_sapiens
3      hgu133a      homo_sapiens
4      hgu133b      homo_sapiens
5     hgu95av2      homo_sapiens
6       hgu95b      homo_sapiens
7       hgu95c      homo_sapiens
8       hgu95d      homo_sapiens
9       hgu95e      homo_sapiens
10    mgu74av2      mus_musculus
11    mgu74bv2      mus_musculus
12    mgu74cv2      mus_musculus
13  mouse430_2      mus_musculus
14  mouse430a2      mus_musculus
15   mu11ksuba      mus_musculus
16   mu11ksubb      mus_musculus
17    rat230_2 rattus_norvegicus
18      rgu34a rattus_norvegicus
19      rgu34b rattus_norvegicus
20     rg_u34c rattus_norvegicus
```