An R interface to the Ensembl REST API

Tim Yates

April 18, 2013

Contents

1 Introduction

This package uses the Ensembl REST API[1] (currently in beta) to extract data from Ensembl into R.

As the REST API is in Beta, this package should also be considered to be in flux and functions/-parameters/etc are subject to change as things get finalized.

It could also do with your help. If you find a problem, something you think could be better, or a better way of doing things, please consider visiting the GitHub project at `https://github.com/acbb/EnsemblRest` and posting an issue or a Pull Request. Thanks!

2 Available Methods

To begin with (assuming you have installed this package), you need to load it into your R session:

```
> library( EnsemblRest )
```

The following subsections then list the methods available to you.

[1] `http://beta.rest.ensembl.org/`

2.1 Information

2.1.1 isAlive

Firstly, we can check to see that the REST API is accepting calls:

```
> isAlive()
[1] TRUE
```

2.1.2 infoSpecies

To get a list of available species on the server, you can use the infoSpecies call[2].

```
> infoSpecies()[1:3] # Just the first 3
[[1]]
name    : saccharomyces_cerevisiae
aliases : 4932, saccer, saccharomyces cerevisiae (baker's yeast), baker's yeast, scer, sacchar
groups  : core, otherfeatures, variation, funcgen
release : 71

[[2]]
name    : ciona_savignyi
aliases : ciosav, 51511, ciona savignyi, csavignyi, c.savignyi, csav, sea squirt ciona savigny
groups  : core, otherfeatures
release : 71

[[3]]
name    : myotis_lucifugus
aliases : little brown bat, mlucifugus, myoluc, mluc, 59463, myotis lucifugus, myotis_lucifugu
groups  : core, otherfeatures
release : 71
```

2.1.3 infoAssembly

The infoAssembly call[3] returns information about the currently available assemblies in the given species.

```
> infoAssembly( 'human' )
assembly_name                 : GRCh37.p10
assembly_date                 : 2009-02
coord_system_versions         : , GRCh37, NCBI36, NCBI34, NCBI35
schema_build                  : 71_37
genebuild_start_date          : 2010-07-Ensembl
genebuild_initial_release_date : 2011-04
genebuild_last_geneset_update  : 2013-02
genebuild_method              : full_genebuild
top_level_seq_region_names    : 1, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2, 20, 21, 22, 3,
```

[2] http://beta.rest.ensembl.org/documentation/info/species
[3] http://beta.rest.ensembl.org/documentation/info/assembly_info

2.1.4 assemblyDetails

The assemblyDetails call[4] returns information about one of these assemblies.

```
> assemblyDetails( 'X', 'human' )
is_chromosome           : TRUE
length                  : 155270560
assembly_exception_type : REF
coordinate_system       : chromosome
```

2.1.5 infoComparas

The infoComparas call[5] lists the available comparative genomics databases.

```
> infoComparas()
multi
 "71"
```

2.1.6 infoData

The infoData call[6] shows the data releases available to the REST service

```
> infoData()
[1] 71
```

2.1.7 infoRest

infoRest shows the current version[7] of the REST service

```
> infoRest()
[1] "1.3.2"
```

2.1.8 infoSoftware

And finally in the info section, infoSoftware shows[8] the current version of the Ensembl API.

```
> infoSoftware()
[1] 71
```

[4] http://beta.rest.ensembl.org/documentation/info/assembly_stats
[5] http://beta.rest.ensembl.org/documentation/info/comparas
[6] http://beta.rest.ensembl.org/documentation/info/data
[7] http://beta.rest.ensembl.org/documentation/info/rest
[8] http://beta.rest.ensembl.org/documentation/info/software

2.2 Comparative Genomics

2.2.1 geneTree

This method[9] fetches the gene tree in New Hampshire format for a given Ensembl gene tree identifier.

```
> geneTree( 'ENSGT00390000003602' )
[1] "(((((((ENSXMAP00000006983:0.2808,ENSORLP00000004773:0.6548):0.1077,ENSONIP00000006940:0.3
```

it is also possible to specify the NH format you require (ie: for full format):

```
> geneTree( 'ENSGT00390000003602', nh_format='full' )
[1] "(((((((ENSXMAP00000006983:0.2808,ENSORLP00000004773:0.6548):0.1077,ENSONIP00000006940:0.3
```

2.2.2 homologyById

When given an Ensembl Gene ID, returns the homology[10] information for it.

```
> hResponse = homologyById( 'ENSG00000170037' )
> hResponse                 # The response object
$ENSG00000170037
$id: ENSG00000170037 $homologies: 17 homologies
> hResponse[[1]]$homologies[1:2,] # Just the top 2 homologies to save room
  source.perc_pos
1              99
2              99

1 MATSADSPSSPLGAEDLLSDSSEPPGLNQVSSEVTSQLYASLRLSRQAEATARAQLYLPSTSPPHEGLDGFAQELSRSLSVGLEKNLKKKDG
2 MATSADSPSSPLGAEDLLSDSSEPPGLNQVSSEVTSQLYASLRLSRQAEATARAQLYLPSTSPPHEGLDGFAQELSRSLSVGLEKNLKKKDG
  source.protein_id source.perc_id source.cigar_line source.species
1   ENSP00000369614             99              925M   homo_sapiens
2   ENSP00000369614             98              925M   homo_sapiens
       source.id   dn_ds target.perc_pos
1 ENSG00000170037 0.29412             100
2 ENSG00000170037      NA              99

1 MATSADSPSSPLGAEDLLSDSSEPPGLNQVSSEVTSQLYASLRLSRQAEATARAQLYLPSTSPPHEGLDGLAQELSRSLSVGLENNLKKKDG
2 MATSADSPSSPLGAEDLLSDSSEPPGLNQVSSEVTSQLYASLRLSRQAEATARAQLYLPSTSPPHEGLDGLAQELSRSLSVGLENNLKKKDG
   target.protein_id target.perc_id target.cigar_line  target.species
1 ENSPTRP00000014861            100          850MD74M pan_troglodytes
2 ENSGGOP00000006314             98              925M gorilla_gorilla
          target.id   subtype             type
1 ENSPTRG00000008719 Homininae ortholog_one2one
2 ENSGGOG00000006451 Homininae ortholog_one2one
```

2.2.3 homologyBySymbol

You can also retrieve homology information[11] given a symbol and a species;

[9] http://beta.rest.ensembl.org/documentation/info/rest
[10] http://beta.rest.ensembl.org/documentation/info/homology_ensemblgene
[11] http://beta.rest.ensembl.org/documentation/info/homology_symbol

```
> hResponse = homologyBySymbol( 'BRCA2', 'human' )
> hResponse                 # The response object
$BRCA2
$id: ENSG00000139618 $homologies: 17 homologies
> hResponse[[1]]$homologies[1:2,] # Again, just the top 2 homologies to save room
  source.perc_pos
1              99
2              98

1 MPIGSKERPTFFEIFKTRCNKADLGPISLNWFEELSSEAPPYNSEPAEESEHKNNNYEPNLFKTPQRKPSYNQLASTPIIFKEQGLTLPLYQ
2 MPIGSKERPTFFEIFKTRCNKADLGPISLNWFEELSSEAPPYNSEPAEESEHKNNNYEPNLFKTPQRKPSYNQLASTPIIFKEQGLTLPLYQ
  source.protein_id source.perc_id source.cigar_line source.species
1   ENSP00000439902             99             3418M   homo_sapiens
2   ENSP00000439902             96             3418M   homo_sapiens
        source.id   dn_ds target.perc_pos
1 ENSG00000139618 0.29371              99
2 ENSG00000139618 0.61742              98

1 MPIGSKERPTFFEIFKTRCNKADLGPISLNWFEELSSEAPPYNSEPAEESEHKNNNYEPNLFKTPQRKPSYNQLASTPIIFKEQGLTLPLYQ
2 MPVGSKERPTFFEIFKTRCNKADLGPISLHWFEELSSEAPPYNSEPAEESEHKNNNYEPNLFKTPQRKPSYNQLASTPIIFKEQGLTLPLYQ
   target.protein_id target.perc_id    target.cigar_line  target.species
1 ENSPTRP00000009812             99                3418M pan_troglodytes
2 ENSPPYP00000005997             96 975MD302MD442MD1696M    pongo_abelii
          target.id   subtype             type
1 ENSPTRG00000005766 Homininae ortholog_one2one
2 ENSPPYG00000005264 Hominidae ortholog_one2one
```

And using `format='condensed'`, you can get a more condensed result:

```
> hResponse = homologyBySymbol( 'BRCA2', 'human', format='condensed' )
> hResponse                 # The response object
$BRCA2
$id: ENSG00000139618 $homologies: 5 homologies
> hResponse[[1]]$homologies[1:10,]
       subtype         protein_id             species                 id
1    Homininae ENSPTRP00000009812     pan_troglodytes ENSPTRG00000005766
2    Hominidae ENSPPYP00000005997        pongo_abelii ENSPPYG00000005264
3   Hominoidea ENSNLEP00000001277 nomascus_leucogenys ENSNLEG00000001048
4    Homininae ENSGGOP00000015446     gorilla_gorilla ENSGGOG00000015808
5   Catarrhini ENSMMUP00000009432      macaca_mulatta ENSMMUG00000007197
6  Simiiformes ENSCJAP00000034250  callithrix_jacchus ENSCJAG00000018462
7     Primates ENSMICP00000010933  microcebus_murinus ENSMICG00000011994
8  Haplorrhini ENSTSYP00000000441    tarsius_syrichta ENSTSYG00000000478
9    Eutheria ENSECAP00000013146      equus_caballus ENSECAG00000014890
10   Eutheria ENSTTRP00000010004  tursiops_truncatus ENSTTRG00000010541
             type
1  ortholog_one2one
2  ortholog_one2one
3  ortholog_one2one
4  ortholog_one2one
5  ortholog_one2one
6  ortholog_one2one
```

```
7  ortholog_one2one
8  ortholog_one2one
9  ortholog_one2one
10 ortholog_one2one
```

2.3 Cross References

Cross references are links to other data about the object of interest. It should be noted that these other data hold different fields and datatypes, so the results are returned in a `data.frame` containing a superset of column names, with non-applicable columns for a given result filled with `<NA>`.

As a single object may have multiple synonyms, this will cause the object to exist in multiple rows, one for each synonym.

2.3.1 xrefsById

Firstly we can get all external references[12] for a given Ensembl ID:

```
> xrefsById( 'ENSG00000170037' )
                                   display_id         primary_id version
1                        OTTHUMG00000172932 OTTHUMG00000172932       2
2                           ENSG00000170037    ENSG00000170037       0
3                                    CNTROB             116840       0
4                                    CNTROB              29616       0
5                                    CNTROB              29616       0
6   CENTROSOMAL BRCA2-INTERACTING PROT [*611425]           611425       0
7                                 Hs.348012          Hs.348012       0
8                                 Hs.732863          Hs.732863       0
9                                    CNTROB             CNTROB       0
10                                   CNTROB             116840       0
             dbname      info_type          info_text     db_display_name
1              OTTG           NONE                           Havana gene
2       ArrayExpress         DIRECT                          ArrayExpress
3         EntrezGene      DEPENDENT                            EntrezGene
4              HGNC         DIRECT Generated via ccds         HGNC Symbol
5              HGNC         DIRECT Generated via ccds         HGNC Symbol
6          MIM_GENE      DEPENDENT                              MIM gene
7           UniGene SEQUENCE_MATCH                               UniGene
8           UniGene SEQUENCE_MATCH                               UniGene
9  Uniprot_genename      DEPENDENT                 UniProtKB Gene Name
10         WikiGene      DEPENDENT                              WikiGene
                                                                                   description
1                                                                                         <NA>
2
3                                                    centrobin, centrosomal BRCA2 interacting protein
4                                                    centrobin, centrosomal BRCA2 interacting protein
5                                                    centrobin, centrosomal BRCA2 interacting protein
6                                                       CENTROSOMAL BRCA2-INTERACTING PROTEIN; CNTROB
7                                                    Centrobin, centrosomal BRCA2 interacting protein
8  Transcribed locus, moderately similar to NP_444279.2 centrobin isoform alpha [Homo sapiens]
9
10                                                   centrobin, centrosomal BRCA2 interacting protein
   synonyms ensembl_identity ensembl_start xref_start xref_end ensembl_end
1      <NA>               NA            NA         NA       NA          NA
2      <NA>               NA            NA         NA       NA          NA
3      LIP8               NA            NA         NA       NA          NA
4      LIP8               NA            NA         NA       NA          NA
5    PP1221               NA            NA         NA       NA          NA
6      <NA>               NA            NA         NA       NA          NA
```

[12] http://beta.rest.ensembl.org/documentation/info/xref_id

```
7      <NA>              99            1         32     3794         3769
8      <NA>              99            1          6      752          752
9      LIP8              NA           NA         NA       NA           NA
10     <NA>              NA           NA         NA       NA           NA
   score      cigar_line xref_identity
1     NA            <NA>            NA
2     NA            <NA>            NA
3     NA            <NA>            NA
4     NA            <NA>            NA
5     NA            <NA>            NA
6     NA            <NA>            NA
7  18783     376M6D3387M            90
8   3677 289M6D406M1I51M            99
9     NA            <NA>            NA
10    NA            <NA>            NA
```

2.3.2 xrefsByName

Or, we can look for an external reference primary accession[13] (given a species):

```
> xrefsByName( 'NM_004333', 'human' )
   display_id primary_id version description      dbname info_type
1 NM_004333.4  NM_004333       4             RefSeq_mRNA    DIRECT
                   info_text db_display_name
1 Generated via otherfeatures     RefSeq mRNA
```

2.3.3 xrefsBySymbol

And we can finally look up all Ensembl objects referenced by an external symbol for a given species[14]:

```
> xrefsBySymbol( 'BRAF', 'human' )
        type              id
1       gene ENSG00000157764
2 transcript ENST00000288602
```

[13] http://beta.rest.ensembl.org/documentation/info/xref_name
[14] http://beta.rest.ensembl.org/documentation/info/xref_external

2.4 Features

2.4.1 featuresByRegion

We can also look for features along a given range[15] (by default this will just look for genes):

```
> featuresByRegion( '7:140424943-140624564', 'human' )
GRanges with 2 ranges and 6 metadata columns:
      seqnames                ranges strand |              ID
         <Rle>             <IRanges>  <Rle> |        <factor>
  [1]        7 [140424943, 140624564]      - | ENSG00000157764
  [2]        7 [140583872, 140583978]      + | ENSG00000207040
               logic_name feature_type external_name
                 <factor>     <factor>      <factor>
  [1] ensembl_havana_gene         gene          BRAF
  [2]               ncrna         gene            U6
                                                                        description
                                                                          <factor>
  [1] v-raf murine sarcoma viral oncogene homolog B1 [Source:HGNC Symbol;Acc:1097]
  [2]                                U6 spliceosomal RNA [Source:RFAM;Acc:RF00026]
           biotype
          <factor>
  [1] protein_coding
  [2]          snRNA
  ---
  seqlengths:
    7
   NA
```

And by using the `feature` parameter, we can specify what we're looking for

```
> featuresByRegion( '7:140424943-140624564', 'human', feature='transcript' )
GRanges with 6 ranges and 5 metadata columns:
      seqnames                ranges strand |              ID
         <Rle>             <IRanges>  <Rle> |        <factor>
  [1]        7 [140424943, 140482957]      - | ENST00000496384
  [2]        7 [140434279, 140624564]      - | ENST00000288602
  [3]        7 [140434321, 140454011]      - | ENST00000479537
  [4]        7 [140434397, 140624458]      - | ENST00000497784
  [5]        7 [140533861, 140624509]      - | ENST00000469930
  [6]        7 [140583872, 140583978]      + | ENST00000384313
                    logic_name feature_type          Parent
                      <factor>     <factor>        <factor>
  [1]                   havana   transcript ENSG00000157764
  [2] ensembl_havana_transcript   transcript ENSG00000157764
  [3]                   havana   transcript ENSG00000157764
  [4]                   havana   transcript ENSG00000157764
  [5]                   havana   transcript ENSG00000157764
  [6]                    ncrna   transcript ENSG00000207040
                     biotype
                    <factor>
  [1]          protein_coding
  [2]          protein_coding
```

[15] http://beta.rest.ensembl.org/documentation/info/feature_region

```
[3] nonsense_mediated_decay
[4] nonsense_mediated_decay
[5]         retained_intron
[6]                   snRNA
---
seqlengths:
  7
 NA
```

You can specify multiple features (columns which don't exist for a given type of result will be filled with NA)

```
> featuresByRegion( '7:140424943-140624564', 'human', feature=c('gene','transcript') )
GRanges with 8 ranges and 7 metadata columns:
      seqnames                ranges strand |              ID
         <Rle>             <IRanges>  <Rle> |        <factor>
  [1]        7 [140424943, 140624564]      - | ENSG00000157764
  [2]        7 [140583872, 140583978]      + | ENSG00000207040
  [3]        7 [140424943, 140482957]      - | ENST00000496384
  [4]        7 [140434279, 140624564]      - | ENST00000288602
  [5]        7 [140434321, 140454011]      - | ENST00000479537
  [6]        7 [140434397, 140624458]      - | ENST00000497784
  [7]        7 [140533861, 140624509]      - | ENST00000469930
  [8]        7 [140583872, 140583978]      + | ENST00000384313
                   logic_name feature_type external_name
                     <factor>     <factor>      <factor>
  [1]       ensembl_havana_gene         gene          BRAF
  [2]                     ncrna         gene            U6
  [3]                   havana   transcript          <NA>
  [4] ensembl_havana_transcript   transcript          <NA>
  [5]                   havana   transcript          <NA>
  [6]                   havana   transcript          <NA>
  [7]                   havana   transcript          <NA>
  [8]                     ncrna   transcript          <NA>
                                                                  description
                                                                    <factor>
  [1] v-raf murine sarcoma viral oncogene homolog B1 [Source:HGNC Symbol;Acc:1097]
  [2]                              U6 spliceosomal RNA [Source:RFAM;Acc:RF00026]
  [3]                                                                      <NA>
  [4]                                                                      <NA>
  [5]                                                                      <NA>
  [6]                                                                      <NA>
  [7]                                                                      <NA>
  [8]                                                                      <NA>
                    biotype          Parent
                   <factor>        <factor>
  [1]          protein_coding            <NA>
  [2]                   snRNA            <NA>
  [3]          protein_coding ENSG00000157764
  [4]          protein_coding ENSG00000157764
  [5] nonsense_mediated_decay ENSG00000157764
  [6] nonsense_mediated_decay ENSG00000157764
  [7]         retained_intron ENSG00000157764
  [8]                   snRNA ENSG00000207040
```

```
---
seqlengths:
  7
 NA
```

2.5 Lookup

2.5.1 lookupId

To find the database and species containing a known Ensembl id, you can use the lookup function[16] like so:

```
> lookupId( 'ENSG00000170037' )
                id      species object_type db_type
1 ENSG00000170037 homo_sapiens        Gene    core
```

[16] http://beta.rest.ensembl.org/documentation/info/lookup

2.6 Mapping

The mapping functions are used to convert co-ordinates between systems or databases.

There is currently an issue with `mappingCdna`, `mappingCds` and `mappingTranslation` in that the `seq_region_name` is not returned from the REST interface. This has been reported to Ensembl and should be fixed in the next release.

2.6.1 mapping

The `mapping` function[17] converts the co-ordinates in one assembly into another, ie:

```
> mapping( 'NCBI36', 'X:1..10000:1', 'GRCh37', 'human' )
[[1]]
GRanges with 2 ranges and 3 metadata columns:
      seqnames         ranges strand |    assembly coordinate_system
         <Rle>      <IRanges>  <Rle> | <character>       <character>
  [1]        X [    1, 10000]      + |      NCBI36        chromosome
  [2]        X [60001, 70000]      + |      GRCh37        chromosome
             type
      <character>
  [1]    original
  [2]      mapped
  ---
  seqlengths:
    X
   NA
```

As you can see, it returns one `GRanges` object per result, with an `original` row and a `mapped` row.

2.6.2 mappingCdna

This function[18] converts CDNA co-ordinates for a given Ensembl Transcript to genomic co-ordinates.

```
> mappingCdna( 'ENST00000288602', '100..300' )
GRanges with 2 ranges and 2 metadata columns:
      seqnames                ranges strand |       gap      rank
         <Rle>             <IRanges>  <Rle> | <numeric> <numeric>
  [1]        7 [140624366, 140624465]      - |         0         0
  [2]        7 [140549912, 140550012]      - |         0         0
  ---
  seqlengths:
    7
   NA
```

2.6.3 mappingCds

Or you can convert CDS co-ordinates[19] instead of CDNA ones:

```
> mappingCds( 'ENST00000288602', '100..300' )
```

[17] http://beta.rest.ensembl.org/documentation/info/assembly_map
[18] http://beta.rest.ensembl.org/documentation/info/assembly_cdna
[19] http://beta.rest.ensembl.org/documentation/info/assembly_cds

```
GRanges with 3 ranges and 2 metadata columns:
      seqnames                ranges strand |       gap      rank
         <Rle>             <IRanges>  <Rle> | <numeric> <numeric>
  [1]        7 [140624366, 140624404]      - |         0         0
  [2]        7 [140549911, 140550012]      - |         0         0
  [3]        7 [140534613, 140534672]      - |         0         0
  ---
  seqlengths:
    7
   NA
```

2.6.4 mappingTranslation

And finally, it is possible to convert from protein co-ordinates to genomic ones using the `mappingTranslation` method[20]:

```
> mappingTranslation( 'ENSP00000288602', '100..300' )
GRanges with 5 ranges and 2 metadata columns:
      seqnames                ranges strand |       gap      rank
         <Rle>             <IRanges>  <Rle> | <numeric> <numeric>
  [1]        7 [140534409, 140534615]      - |         0         0
  [2]        7 [140508692, 140508795]      - |         0         0
  [3]        7 [140507760, 140507862]      - |         0         0
  [4]        7 [140501212, 140501360]      - |         0         0
  [5]        7 [140500242, 140500281]      - |         0         0
  ---
  seqlengths:
    7
   NA
```

[20] `http://beta.rest.ensembl.org/documentation/info/assembly_translation`

2.7 Sequences

2.7.1 sequenceById

Fetch a sequence based on the stable id of an Ensembl feature[21] (I'm using `str` here to avoid overflowing the pdf too much):

```
> str( sequenceById( 'ENSG00000157764' ), give.head=F, strict.width='cut' )
List of 4
 $ desc    :"chromosome:GRCh37:7:140424943:140624564:-1"
 $ id      :"ENSG00000157764"
 $ seq     :"CGCCTCCCTTCCCCCTCCCCGCCCGACAGCGGCCGCTCGGGCCCCGGCTCTCGGTTATAAGATGG..
 $ molecule:"dna"
```

You can also get different types of sequence, here is an example for the spliced CDNA sequence of a transcript:

```
> str( sequenceById( 'ENST00000408384', type='cdna' ), give.head=F, strict.width='cut' )
List of 4
 $ desc    : NULL
 $ id      :"ENST00000408384"
 $ seq     :"GGATGCCCAGCTAGTTTGAATTTTAGATAAACAACGAATAATTTCGTAGCATAAATATGTCCCAA..
 $ molecule:"dna"
```

And again, for the protein coding

```
> str( sequenceById( 'ENSP00000334393', type='protein' ), give.head=F, strict.width='cut' )
List of 4
 $ desc    : NULL
 $ id      :"ENSP00000334393"
 $ seq     :"MVTEFIFLGLSDSQELQTFLFMLFFVFYGGIVFGNLLIVITVVSDSHLHSPMYFLLANLSLIDLS..
 $ molecule:"protein"
```

2.7.2 sequenceByRegion

You can also just query for a region of a given species[22]:

```
> str( sequenceByRegion( 'X:1000000..1000100:1', 'human' ), give.head=F, strict.width='cut' )
List of 3
 $ id      :"chromosome:GRCh37:X:1000000:1000100:1"
 $ seq     :"GAAACAGCTACTTGGAAGGCTGAAGCAGGAGGATTGTTTGAGTCTAGGAGTTTGAGGCTGCAGTG..
 $ molecule:"dna"
```

If you pass `format='fasta'` to the above method, it will just return you a character vector containing a FastA formatted sequence.

[21] http://beta.rest.ensembl.org/documentation/info/sequence_id
[22] http://beta.rest.ensembl.org/documentation/info/sequence_region

2.8 Variation

2.8.1 variationAllele

```
> var = variationAllele( 'C', '9:22125503-22125502:1', 'human' )
> var                       # The response object
[[1]]
hgvs        :
  C = 9:g.22125502_22125503insC
transcripts : 10 in total
> var[[1]]$transcripts[1:2] # Just the top 2 transcripts of the first response to save room
[[1]]
data             :
        name         gene_id   transcript_id   biotype cdna_allele_string
1 CDKN2B-AS1 ENSG00000240498 ENST00000585267 antisense                -/C
  is_canonical
1        FALSE

alleles          :
        consequence_terms
1 downstream_gene_variant

[[2]]
data             :
        name         gene_id   transcript_id   biotype cdna_allele_string
1 CDKN2B-AS1 ENSG00000240498 ENST00000580576 antisense                -/C
  is_canonical
1        FALSE

alleles          :
        consequence_terms
1 downstream_gene_variant
```

2.8.2 variationId

```
> var = variationId( 'COSM476', 'human' )
> var                       # The response object
[[1]]
name        : COSM476
is_somatic  : TRUE
hgvs        :
  T = 7:g.140453136A>T
transcripts : 4 in total
> var[[1]]$transcripts[1:2] # Again, just the top 2 transcripts to save room
[[1]]
data             :
  name         gene_id   transcript_id        biotype       ccds
1 BRAF ENSG00000157764 ENST00000288602 protein_coding CCDS5863.1
  cdna_allele_string codon_position translation_stable_id translation_start
1                T/A              2       ENSP00000288602               600
  translation_end exon_number cdna_start cdna_end cds_start cds_end
```

```
1             600       15/18       1860     1860      1799    1799
  is_canonical
1         TRUE

alleles          :
  display_codon_allele_string pep_allele_string codon_allele_string
1                     gTg/gAg               V/E             GTG/GAG
             hgvs_transcript                  hgvs_protein polyphen_score
1 ENST00000288602.6:c.1799T>A ENSP00000288602.6:p.Val600Glu          0.999
  polyphen_prediction sift_score sift_prediction consequence_terms
1   probably damaging          0     deleterious  missense_variant

protein_features :
      name                  db
1  PF07714         Pfam domain
2  PF00069         Pfam domain
3 SSF56112 Superfamily domains
4  SM00220       SMART domains
5  SM00219       SMART domains
6  PS50011    PROSITE profiles

[[2]]
data             :
  name         gene_id   transcript_id                biotype
1 BRAF ENSG00000157764 ENST00000479537 nonsense_mediated_decay
  cdna_allele_string codon_position translation_stable_id translation_start
1                T/A              2       ENSP00000418033                28
  translation_end exon_number cdna_start cdna_end cds_start cds_end
1              28         2/6         83       83        83      83
  is_canonical
1        FALSE

alleles          :
  display_codon_allele_string pep_allele_string codon_allele_string
1                     gTg/gAg               V/E             GTG/GAG
            hgvs_transcript                 hgvs_protein polyphen_score
1 ENST00000479537.1:c.83T>A ENSP00000418033.1:p.Val28Glu          0.946
  polyphen_prediction sift_score sift_prediction             V10
1   probably damaging       0.12       tolerated missense_variant
       consequence_terms
1 NMD_transcript_variant

protein_features :
      name                  db
1  PF00069         Pfam domain
2  PF07714         Pfam domain
3 SSF56112 Superfamily domains
4  PS50011    PROSITE profiles
```