Package ‘mygene’

April 2, 2024

Type Package
Title Access MyGene.Info_ services
Version 1.38.0
Date 2014-04-18
Author Adam Mark, Ryan Thompson, Cyrus Afrasiabi, Chunlei Wu
Maintainer Adam Mark, Cyrus Afrasiabi, Chunlei Wu <cwu@scripps.edu>
Description MyGene.Info_ provides simple-to-use REST web services
to query/retrieve gene annotation data. It's designed
with simplicity and performance emphasized. *mygene*,
is an easy-to-use R wrapper to access MyGene.Info_ services.
License Artistic-2.0
Depends R (>= 3.2.1), GenomicFeatures,
Imports httr (>= 0.3), jsonlite (>= 0.9.7), S4Vectors, Hmisc, sqldf,
plyr
Suggests BiocStyle
biocViews Annotation
git_url https://git.bioconductor.org/packages/mygene
git_branch RELEASE_3_18
git_last_commit f4ab791
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-04-01

R topics documented:

getGene ......................................................... 2
getGenes ...................................................... 3
makeTxDbFromMyGene ......................................... 4
metadata ....................................................... 5
MyGene ......................................................... 5
getGene

Return the gene object for the given geneid.

Description

This is a wrapper for GET query of "/gene/<geneid>" service.

Usage

getGene(geneid, fields = c("symbol", "name", "taxid", "entrezgene"),
..., return.as=c("records", "text"), mygene)

Arguments

- **geneid**: Entrez/ensembl gene id
- **fields**: Fields to return, a list of a comma-sep string. If fields="all", all available fields are returned.
- **...**: Includes species as well as several other fields. View available fields by calling ?metadata. Also, see http://docs.mygene.info/en/latest/doc/annotation_service.html for complete argument details and syntax.
- **return.as**: "records" (list), "text" (JSON).
- **mygene**: A MyGene object that describes how to connect to data resources. See **MyGene-class**. If missing, default object will be used that accesses the main MyGene.info portal. Default is recommended.

Value

returns a gene object containing the queried annotations

References

Ref: http://docs.mygene.info/en/latest/doc/annotation_service.html

See Also

- getGenes
- query
- queryMany
getGenes

Examples

```r
## return the gene object for the given gene id
getGene(1017)

## customize fields
getGene(1017, fields=c("name", "symbol", "refseq"), return.as="text")

## all fields
getGene(1017, fields="all")
```

getGenes

Return the list of gene objects for the given list of geneids.

Description

This is a wrapper for POST query of "/gene" service.

Usage

```r
getGenes(geneids, fields = c("symbol", "name", "taxid", "entrezgene"), ..., 
         return.as=c("DataFrame", "records", "text"), mygene)
```

Arguments

- `geneids` A vector, list, or comm-sep string entrez/ensembl gene ids
- `fields` A vector of fields to return. If fields=="all", all available fields are returned.
- `...` Includes species as well as several other fields. View available fields by calling ?metadata. Also, see http://docs.mygene.info/en/latest/doc/annotation_service.html for complete argument details and syntax.
- `return.as` "DataFrame" (default), "records" (list), "text" (JSON).
- `mygene` A MyGene object that describes how to connect to data resources. See MyGene-class. If missing, default object will be used that accesses the main MyGene.info portal. Default is recommended.

Value

returns a gene object (DataFrame, list, or JSON text) containing the queried annotations

References

Ref: http://docs.mygene.info/en/latest/doc/annotation_service.html

See Also

gene id query queryMany
Examples

```r
## Return the list of gene object for the given list of gene ids.
getGenes(c(1017,1018))

## mix types of gene ids
getGenes(c(1017,1018,"ENSG00000148795"))
```

Description

`makeTxDbFromMyGene` allows the user to make a `TxDb` object in the Genomic Features package from a mygene "exons" query using a default MyGene object.

Usage

```r
makeTxDbFromMyGene(gene.list, scopes, species, returnall=FALSE)
```

Arguments

- `gene.list`: A list, vector, or comma-separated string of query terms.
- `scopes`: Type of types of identifiers, either a list, vector, or a comma-separated fields to specify type of input qterms, e.g. c("reporter", "ensembl.gene", "symbol") refer to "http://docs.mygene.info/en/latest/doc/data.html#available-fields" for full list of fields.
- `species`: Names or taxonomy ids
- `returnall`: Logical, if TRUE, return list of genes without exons annotations. False by Default.

Details

`makeTxDbFromMyGene` invokes either the query or queryMany method and passes the response to construct a `TxDb` object. See `?TxDb` for utilizing transcript annotations.

Value

returns `TxDb` object

See Also

`makeTxDb`

Examples

```r
xli <- c("DDX26B", "CCDC83", "MAST3", "RPL11")
txdb <- makeTxDbFromMyGene(xli, scopes="symbol", species="human")
```
Description

Get metadata for MyGene.info services.

Usage

metadata(x, ...)

Arguments

x   MyGene object
...

MyGene object slot parameters

Value

returns the metadata including available fields, genome assemblies, sources, statistics, taxonomy, and timestamp

References

http://mygene.info/v2/metadata

Examples

## Get metadata
mygene<-MyGene()
metadata(mygene)

## get available fields to return
metadata(mygene)$available_fields

Description

Construct a MyGene object.

Usage

MyGene(...)
Arguments

See help page for MyGene-class

Value

MyGene object

Examples

mygene

mygene()  

Access MyGene.info annotation services

Description

MyGene.Info_ provides simple-to-use REST web services to query/retrieve gene annotation data. It’s designed with simplicity and performance emphasized. *mygene* is an easy-to-use R wrapper to access MyGene.info services.

Details

Package: mygene
Type: Package
Version: 0.99.0
Date: 2014-04-18
License: BSD
Depends: httr jsonlite Hmisc

Author(s)

Adam Mark, Chunlei Wu
Maintainer: Chunlei Wu <help@mygene.info>

References

http://mygene.info/doc/query_service.html
**MyGene-class**

**Description**

R Client to access MyGene.Info annotation services

**Objects from the Class**

Objects can be created by calls of the form `MyGene(base.url="http://mygene.info/v2", delay=1, step=1000, version=version, verbose=TRUE, debug=FALSE)`.

**Slots**

- `base.url`: "http://mygene.info/v2". Object of class "character"
- `delay`: Sleep time between batch retrieval. Object of class "numeric"
- `step`: Batch limit. Object of class "numeric"
- `version`: httr package version. Object of class "character"
- `verbose`: Object of class "logical"
- `debug`: Object of class "logical"

**Methods**

- `getGene(geneid, fields = c("symbol", "name", "taxid", "entrezgene"), ..., return.as=c("records", "text"))`: Return the gene object for the given geneid
- `getGenes(geneids, fields = c("symbol", "name", "taxid", "entrezgene"), ..., return.as=c("DataFrame", "records", "text"))`: Return the list of gene object for the given list of geneids.
- `query(q, fields=c("name", "symbol", "taxid", "entrezgene"), ..., return.as=c("DataFrame", "records", "text"))`: Return the query result.
- `queryMany(qterms, scopes=NULL, fields=c("name", "symbol", "taxid", "entrezgene"), ..., return.as=c("DataFrame", "records", "text"), returnall=FALSE)`: Return the batch query result.
- `metadata(x, ...)`: Get metadata for MyGene.info services.
- `makeTxDbFromMyGene(gene.list, scopes, species, returnall=FALSE)`: Make a TxDb object from transcript annotations

**Author(s)**

Adam Mark, Chunlei Wu, Ryan Thompson

**References**


**Examples**

`showClass("MyGene")`
query

Return the query result.

Description
This is a wrapper for GET query of "/query?q=<query>" service.

Usage
query(q, ..., return.as=c("DataFrame", "records", "text"), mygene)

Arguments
q query term(s), see query syntax at http://mygene.info/doc/query_service.html#query-syntax
... Commonly queried fields include species, fields, size as well as several other fields. View available fields by calling ?metadata. Also, see http://docs.mygene.info/en/latest/doc/query_service.html for complete argument details and syntax.
return.as "DataFrame" (default), "records" (list), or "text" (JSON).
mygene A MyGene object that describes how to connect to data resources. See MyGene-class. If missing, default object will be used that accesses the main MyGene.info portal. Default is recommended.

Value
returns a gene object (DataFrame, list, or JSON text) containing the queried annotations

References

See Also
queryMany getGene getGenes

Examples
## return the query result
query("cdk2", size=5)
query("reporter:1000_at")
query("symbol:cdk2", return.as="text")
query(q="cyclin-dependent kinase", fields="uniprot")
queryMany  

Return the batch query result.

Description

This is a wrapper for POST query of "/query" service.

Usage

queryMany(qterms, scopes=NULL, ..., return.as=c("DataFrame", "records", "text"), mygene)

Arguments

qterms  A vector or list, or string of comma-separated query terms
scopes  Type of types of identifiers, either a list or a comma-separated fields to specify type of input qterms, e.g. c("reporter", "ensembl.gene", "symbol") refer to "http://mygene.info/doc/query_service.html#available_fields" for full list of fields.
...  Commonly queried fields include species, fields, size as well as several other fields. returnall returns a list of all related data including duplicated and missing qterms. False by default. View available fields by calling ?metadata. Also, see http://docs.mygene.info/en/latest/doc/query_service.html for complete argument details and syntax.
return.as  "DataFrame" (default), "records" (list), "text" (JSON).
mygene  A MyGene object that describes how to connect to data resources. See MyGene-class. If missing, default object will be used that accesses the main MyGene.info portal. Default is recommended.

Value

returns a gene object (DataFrame, list, or JSON text) containing the queried annotations

References


See Also

query getGene getGenes

Examples

## return the batch query result
queryMany(c('1053_at', '117_at', '121_at'), scopes="reporter", fields="ensembl.gene", species="human", return.as="records")

queryMany(c('1053_at', '117_at', '121_at'), scopes="reporter", species=9606)
queryMany(c('DDX26B', 'CCDC83', 'MAST3', 'FLOT1'), scopes="symbol", fields="entrezgene", species="human")
Index

* classes
  MyGene-class, 7

* package
  mygene, 6
  getGene, 2, 3, 8, 9
  getGene, missing-method (getGene), 2
  getGene, MyGene-method (getGene), 2
  getGenes, 2, 3, 8, 9
  getGenes, missing-method (getGenes), 3
  getGenes, MyGene-method (getGenes), 3

  makeTxDb, 4
  makeTxDbFromMyGene, 4
  metadata, 5
  metadata, MyGene-method (metadata), 5
  MyGene, 5
  mygene, 6
  MyGene-class, 7

  query, 2, 3, 8, 9
  query, missing-method (query), 8
  query, MyGene-method (query), 8
  queryMany, 2, 3, 8, 9
  queryMany, missing-method (queryMany), 9
  queryMany, MyGene-method (queryMany), 9