Package ‘myvariant’

November 20, 2016

Type  Package
Title  Accesses MyVariant.info variant query and annotation services
Version  1.4.0
Date  2015-07-10
Author  Adam Mark
Maintainer  Adam Mark, Chunlei Wu <cwu@scripps.edu>
Description  MyVariant.info is a comprehensive aggregation of variant annotation resources. myvariant is a wrapper for querying MyVariant.info services
License  Artistic-2.0
Depends  R (>= 3.2.1), VariantAnnotation
Imports  httr, jsonlite, S4Vectors, Hmisc, plyr, magrittr, GenomeInfoDb
Suggests  BiocStyle
biocViews  VariantAnnotation, Annotation, GenomicVariation
NeedsCompilation  no

R topics documented:

formatHgvs ................................................................. 2
formatSingleHgvs ....................................................... 2
getVariant ............................................................... 3
getVariants .............................................................. 4
metadata ................................................................. 5
MyVariant ............................................................... 6
myvariant ............................................................... 6
MyVariant-class ........................................................ 7
queryVariant ........................................................... 8
queryVariants .......................................................... 9

Index  10
formatHgvs  

*Get all HGVS IDs from Vcf object.*

**Description**
Read in a Vcf object created by `readVcf` to extract all HGVS IDs for querying MyVariant.info.

**Usage**

```r
formatHgvs(vcf, variant_type = c("snp", "insertion", "deletion"))
```

**Arguments**

- `vcf`  
  Vcf object created by `readVcf`.
- `variant_type`  
  Type of variant HGVS IDs to retrieve from Vcf object. Default c("snp", "insertion", "deletion")

**Value**

vector

**References**


**See Also**

`formatSingleHgvs`

**Examples**

```r
## return HGVS IDs for all snps in a Vcf
file.path <- system.file("extdata", "dbsnp_mini.vcf", package="myvariant")
vcf <- readVcf(file.path, genome="hg19")
hgvs <- formatHgvs(vcf, variant_type="snp")
```

---

formatSingleHgvs  

*Get Hgvs HGVS ID from chromosome, position, reference and alternate alleles.*

**Description**
Create a single HGVS ID for a variant from chromosome, position, reference and alternate alleles.

**Usage**

```r
formatSingleHgvs(chrom, pos, ref, alt, mutant_type=FALSE)
```
getVariant

Arguments

- chrom: Chromosome.
- pos: Position of the variant on the reference genome (hg19).
- ref: Reference allele.
- alt: Alternate allele.
- mutant_type: Logical indicating whether to return the type of mutation along with the HGVS ID.

Value

returns a string

References


See Also

formatHgvs

Examples

```r
## return HGVS ID for a variant
formatSingleHgvs(1, 35367, "G", "A")
```

getVariant

_Return the variant object for the given HGVS id._

Description

This is a wrapper for GET query of "/variant/<hgvsid>" service.

Usage

```
getVariant(hgvsid, fields=NULL, 
..., return.as=c("records", "text"), myvariant)
```

Arguments

- hgvsid: HGVS id
- fields: Fields to return, a list of a comma-sep string. If fields=="all", all available fields are returned.
- ... return.as: "records" (list), "text" (JSON).
- myvariant: A MyVariant object that describes how to connect to data resources. See MyVariant-class. If missing, default object will be used that accesses the main MyVariant.info portal. Default is recommended.
getVariants

Value
returns a variant object containing the queried annotations

References

See Also
getVariants queryVariant queryVariants

Examples
## return the variant object for the given HGVS id
getVariant("chr7:g.55241707G>T")

## customize fields
getVariant("chr7:g.55241707G>T", fields=c("dbnsfp.cadd.phred", "dbnsfp.polyphen2"), return.as="text")

getVariants Return the list of variant objects for the given list of HGVS ids.

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

Usage
getVariants(hgvsids, fields=NULL, verbose=NULL, ..., return.as=c("DataFrame", "records", "text"), myvariant)

Arguments
hgvsids A vector, list, or comm-sep string HGVS ids
fields A vector of fields to return. If fields=="all", all available fields are returned.
verbose A logical turning on or off process status messages. Default = TRUE.
...
return.as "DataFrame" (default), "records" (list), "text" (JSON).
myvariant A MyVariant object that describes how to connect to data resources. See MyVariant-class. If missing, default object will be used that accesses the main MyVariant.info portal. Default is recommended.

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

**Description**

Get metadata for MyVariant.info services.

**Usage**

```r
metadata(x, ...)
```

**Arguments**

- `x`  
  MyVariant object
- `...`  
  MyVariant object slot parameters

**Value**

returns the metadata including available databases and number of documents.

**References**

http://myvariant.info/v1/metadata

---

## Examples

```r
## given a list of HGVS ids
vars <- c('chr1:g.866422C>T',
          'chr1:g.876664G>A',
          'chr1:g.69635G>C',
          'chr1:g.69869T>A',
          'chr1:g.881918G>A',
          'chr1:g.865625G>A',
          'chr1:g.879368C>A',
          'chr1:g.889226C>T',
          'chr1:g.879429C>G',
          'chr1:g.879423T>G',
          'chr1:g.881602C>T',
          'chr1:g.879115C>G',
          'chr1:g.69892T>C',
          'chr1:g.879381C>T',
          'chr1:g.878330C>G')
```

```r
## Return the list of variant object for the given list of HGVS ids.

df <- getVariants(vars, fields="dbsnp, wellderly")
```
Examples

```r
## Get metadata
myvariant<-MyVariant()
metadata(myvariant)
```

### Description

Construct a MyVariant object.

### Usage

```r
MyVariant(...) 
```

### Arguments

...  

See help page for MyVariant-class

### Value

MyVariant object

### Examples

```r
MyVariant()
```

---

### Description

Access MyVariant.info variant annotation services

### Details

- **Package:** myvariant
- **Type:** Package
- **Version:** 0.99.0
- **Date:** 2014-12-18
- **License:** Artistic-2.0
- **Depends:** httr jsonlite Hmisc
**MyVariant-class**

**Author(s)**

Adam Mark  
Maintainer: Adam Mark <adammark@scripps.edu>

**References**

https://github.com/Network-of-BioThings/myvariant.info/wiki

---

**Description**

R Client to access MyVariant.Info annotation services

**Objects from the Class**

Objects can be created by calls of the form `MyVariant(base.url="http://myvariant.info/v1", delay=1, step=1000, version=version, verbose=TRUE, debug=FALSE)`.

**Slots**

- `base.url`: "http://myvariant.info/v1". 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**

- `getVariant(hgvsid, fields=NULL, ..., return.as=c("records", "text"))`: Return the variant object for the given hgvsid  
- `getVariants(hgvsids, fields=NULL, ..., return.as=c("DataFrame", "records", "text"))`: Return the list of variant object for the given list of hgvsids.  
- `queryVariant(q, fields=NULL, ..., return.as=c("DataFrame", "records", "text"))`: Return the query result.  
- `queryVariants(qterms, scopes=NULL, fields=NULL, ..., return.as=c("DataFrame", "records", "text"))`: Return the batch query result.

**Author(s)**

Adam Mark

**References**

https://github.com/Network-of-BioThings/myvariant.info/wiki

**Examples**

`showClass("MyVariant")`
queryVariant

Return the query result.

Description

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

Usage

queryVariant(q, ..., return.as=c("DataFrame", "records", "text"), myvariant)

Arguments

q query term(s).

... Commonly queried fields include fields, size as well as several other fields. View available fields by calling ?metadata.

return.as "DataFrame" (default), "records" (list), or "text" (JSON).

myvariant A MyVariant object that describes how to connect to data resources. See MyVariant-class. If missing, default object will be used that accesses the main MyVariant.info portal. Default is recommended.

Value

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

References


See Also

queryVariants getVariant getVariants

Examples

## return the query result
queryVariant("dbnsfp.variantname:BRCA2")

queryVariant("chr1:1-1000000")
queryVariants

Return the batch query result.

Description

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

Usage

queryVariants(qterms, scopes=NULL, ..., return.as=c("DataFrame", "records", "text"), myvariant)

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.
... Commonly queried fields include 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.
return.as "DataFrame" (default), "records" (list), "text" (JSON).
myvariant A MyVariant object that describes how to connect to data resources. See MyVariant-class. If missing, default object will be used that accesses the main MyVariant.info portal. Default is recommended.

Value

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

References


See Also

queryVariant getVariant getVariants

Examples

## return the batch query result
Index

*Topic classes
  MyVariant-class, 7
*Topic package
  myvariant, 6

formatHgvs, 2, 3
formatSingleHgvs, 2, 2

getVariant, 3, 8, 9
getVariant, missing-method (getVariant), 3
getVariant, MyVariant-method (getVariant), 3
getVariants, 4, 5, 8, 9
getVariants, missing-method (getVariants), 4
getVariants, MyVariant-method (getVariants), 4

metadata, 5
metadata, MyVariant-method (metadata), 5
MyVariant, 6
myvariant, 6
MyVariant-class, 7

queryVariant, 4, 5, 8, 9
queryVariant, missing-method (queryVariant), 8
queryVariant, MyVariant-method (queryVariant), 8
queryVariants, 4, 5, 8, 9
queryVariants, missing-method (queryVariants), 9
queryVariants, MyVariant-method (queryVariants), 9

readVcf, 2