Package 'myvariant'

November 12, 2026

Type Package

Title Accesses MyVariant.info variant query and annotation services

Version 1.42.0

Date 2025-07-22

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, Seqinfo, GenomeInfoDb

Suggests BiocStyle

biocViews VariantAnnotation, Annotation, GenomicVariation

NeedsCompilation no

git_url https://git.bioconductor.org/packages/myvariant

git_branch RELEASE_3_23

git_last_commit 0c33246

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

formatHgvs 2
formatSingleHgvs 3
getVariant 4
getVariants 5
metadata 6

MyVariant 7
myvariant 7
MyVariant-class 8
queryVariant 9
queryVariants 10

Index **11**

`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

```
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

https://myvariant.info http://www.hgvs.org/mutnomen/recs-DNA.html

See Also

`formatSingleHgvs`

Examples

```
## 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

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

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

https://myvariant.info http://www.hgvs.org/mutnomen/recs-DNA.html

See Also

`formatHgvs`

Examples

```
## 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.

Value

returns a variant object containing the queried annotations

References

http://docs.myvariant.info/en/latest/doc/variant_annotation_service.html#get-request http://docs.myvariant.info/en/latest/doc parameters

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

References

http://docs.myvariant.info/en/latest/doc/variant_annotation_service.html#batch-queries-via-post http://docs.myvariant.info/e

See Also

getVariants queryVariant queryVariants

Examples

```
## 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.879492C>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')

## Return the list of variant object for the given list of HGVS ids.
df <- getVariants(vars, fields="dbsnp, wellderly")
```

`metadata` *metadata*

Description

Get metadata for MyVariant.info services.

Usage

```
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

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

`MyVariant` *MyVariant*

Description

Construct a MyVariant object.

Usage

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

Arguments

`...`	See help page for MyVariant-class

Value

MyVariant object

Examples

```
MyVariant()
```

`myvariant` *Access MyVariant.info variant annotation services*

Description

MyVariant.Info provides REST web services to query/retrieve variant annotations. myvariant is an easy-to-use R wrapper to access MyVariant.info services.

Details

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

Author(s)

Adam Mark

Maintainer: Adam Mark <adammark@scripps.edu>

References

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

`MyVariant-class` *Class* `"MyVariant"`

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"), retur`
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

http://docs.myvariant.info/en/latest/doc/variant_query_service.html#get-request http://docs.myvariant.info/en/latest/doc/vari
syntax

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

http://docs.myvariant.info/en/latest/doc/variant_query_service.html#batch-queries-via-post http://docs.myvariant.info/en/lat

See Also

`queryVariant` `getVariant` `getVariants`

Examples

```
## return the batch query result
```

Index

∗ **classes**
    MyVariant-class, 8
∗ **package**
    myvariant, 7

formatHgvs, 2, *3*
formatSingleHgvs, *2*, 3

getVariant, 4, *9, 10*
getVariant,missing-method *(*getVariant*)*, 4
getVariant,MyVariant-method *(*getVariant*)*, 4
getVariants, *4, 5*, 5, *9, 10*
getVariants,missing-method *(*getVariants*)*, 5
getVariants,MyVariant-method *(*getVariants*)*, 5

metadata, 6
metadata,MyVariant-method *(*metadata*)*, 6
MyVariant, 7
myvariant, 7
MyVariant-class, 8

queryVariant, *4, 5*, 9, *10*
queryVariant,missing-method *(*queryVariant*)*, 9
queryVariant,MyVariant-method *(*queryVariant*)*, 9
queryVariants, *4, 5*, *9*, 10
queryVariants,missing-method *(*queryVariants*)*, 10
queryVariants,MyVariant-method *(*queryVariants*)*, 10

readVcf, *2*