

# Package ‘rapimave’

July 28, 2021

**Title** R API for MaveDB

**Version** 0.2.0.9000

**Description** Allows programatic access to the MaveDB database of Deep Mutational Scans

**Depends** R (>= 3.1.2)

**License** GPL-3 + file COPYING

**Encoding** UTF-8

**LazyData** true

**Imports** hgvsParseR, httr, RJSONIO

**Suggests** testthat

**RoxygenNote** 7.0.2

**Remotes** github::VariantEffect/hgvsParseR

## R topics documented:

new.experiment . . . . .	2
new.experimentSet . . . . .	3
new.genome . . . . .	4
new.mave.filter . . . . .	5
new.rapimave . . . . .	6
new.refmap . . . . .	8
new.scoreSet . . . . .	8
new.target . . . . .	10
new.user . . . . .	11
new.xref . . . . .	12
print.rapimave . . . . .	12
print.rapimaveExperiment . . . . .	13
print.rapimaveExpSet . . . . .	13
print.rapimaveGenome . . . . .	14
print.rapimaveRefmap . . . . .	14
print.rapimaveScoreSet . . . . .	15
print.rapimaveTarget . . . . .	15
print.rapimaveUser . . . . .	16
print.rapimaveXref . . . . .	16

---

new.experiment	<i>New R-API MaveDB Experiment</i>
----------------	------------------------------------

---

## Description

Internal constructor that creates an immutable new Experiment object. This function is not exported and should not be called outside of the rapimave package itself. Experiment objects are generated by the `getAllExperiments()` and `getExperiment()` methods of a rapimave object (which itself is generated using the `new.rapimave()` constructor). However, this section explains the functions that are available for these objects.

## Usage

```
new.experiment(data)
```

## Details

A Experiment object offers the following getter functions:

- `getContributors()` returns a list of users that contributed to this record.
- `getURN()` returns the URN of this record.
- `getCreationDate()` returns the date when this record was originally created.
- `getModificationDate()` returns the date when this record was last modified.
- `getPublicationDate()` returns the date when this record was published.
- `getCreatedBy()` returns the user who originally created this record.
- `getModifiedBy()` returns the user who last modified this record.
- `getMetadata()` returns the metadata for this record.
- `getTitle()` returns the title of this record.
- `getShortDescription()` returns a short description text of this record.
- `getAbstract()` returns the abstract text of this record.
- `getMethods()` returns the methods description text of this record.
- `getKeywords()` returns the list of keywords associated with this record.
- `getDOIs()` returns the list of Digital Object Identifiers (DOIs) associated with this record.
- `getXrefSRA()` returns cross-references to the NCBI Short Read Archive (SRA).
- `getXRefPubmed()` returns cross-references to Pubmed.
- `getExperimentSet()` return the accession of the ExperimentSet to which this Experiment belongs.
- `getScoreSets()` return the list of URNs of the Experiment's ScoreSets.

## Value

a new R-API MaveDB experiment object.

## Examples

```
## Not run:
mave <- new.rapimave()
exp1 <- mave$getExperiment("urn:mavedb:00000001-a")
authorIds <- exp1$getAuthors()

## End(Not run)
```

---

new.experimentSet	<i>New R-API MaveDB ExperimentSet</i>
-------------------	---------------------------------------

---

## Description

Internal constructor that creates an immutable new ExperimentSet object. This function is not exported and should not be called outside of the rapimave package itself. ExperimentSet objects are generated by the getAllExperimentSets() and getExperimentSet() methods of a rapimave object (which itself is generated using the new.rapimave() constructor). However, this section explains the functions that are available for these objects.

## Usage

```
new.experimentSet(data)
```

## Details

A ExperimentSet object offers the following getter functions:

- getContributors() returns a list of users that contributed to this record.
- getURN() returns the URN of this record.
- getCreationDate() returns the date when this record was originally created.
- getModificationDate() returns the date when this record was last modified.
- getPublicationDate() returns the date when this record was published.
- getCreatedBy() returns the user who originally created this record.
- getModifiedBy() returns the user who last modified this record.
- getMetadata() returns the metadata for this record.
- getTitle() returns the title of this record.
- getShortDescription() returns a short description text of this record.
- getAbstract() returns the abstract text of this record.
- getMethods() returns the methods description text of this record.
- getKeywords() returns the list of keywords associated with this record.
- getDOIs() returns the list of Digital Object Identifiers (DOIs) associated with this record.
- getXrefSRA() returns cross-references to the NCBI Short Read Archive (SRA).
- getXRefPubmed() returns cross-references to Pubmed.
- getExperiments() returns the URNs of the experiments belonging to this ExperimentSet.

**Value**

a new R-API MaveDB ExperimentSet object.

**Examples**

```
## Not run:
mave <- new.rapimave()
set1 <- mave$getExperimentSet("urn:mavedb:00000001")
expURNs <- set1$getExperiments()

## End(Not run)
```

---

new.genome

*New R-API MaveDB genome*


---

**Description**

Internal constructor that creates an immutable new genome object. This function is not exported and should not be called outside of the rapimave package itself. Genome objects are generated by the getGenome() method of a reference map object (which itself is generated using the get.referenceMaps() method on a target object). However, this section explains the functions that are available for these objects.

**Usage**

```
new.genome(data)
```

**Details**

A ScoreSet object offers the following getter functions:

- getShortName() returns the name of this genome object
- getOrganism() returns the species to which this genome belongs.
- getAssemblyID() returns the a cross-reference object for the assembly ID.

**Value**

a new R-API MaveDB genome object.

---

new.mave.filter	<i>New MaveDB Filter Object</i>
-----------------	---------------------------------

---

## Description

Constructor that creates a MaveDB filter object

## Usage

```
new.mave.filter(data, verbose = TRUE)
```

## Arguments

data	A data.frame containing a MaveDB data set containing at least the two columns "hgvs" and "score".
verbose	Logical. If TRUE, prints status messages while loading data.

## Details

The resulting object offers a number of filter functions. All these functions return vectors of type logical, which can be combined with & and | to form more complex filters before being applied to the data. (See example below). The following functions are available.

- `mutationCount()` filters by number of mutations in the variant. Parameters `min` and `max` default to 0 and `Inf` respectively.
- `position()` filters by the (start) position of the mutations. Parameters `min` and `max` default to `-Inf` and `Inf` respectively. Parameter `multi` determines whether in case of multi-mutants, the criterium must match any or all of the individual mutations. Allowed values are "any" and "all".
- `residues()` filters by amino acid residues. Parameter `from` and `to` are vectors of allowed ancestral and variant amino acid residues. Parameter `multi` determines whether in case of multi-mutants, the criterium must match any or all of the individual mutations. Allowed values are "any" and "all".
- `numerical()` filters by numerical columns. Parameter `col` is the name of the numerical column to filter by. Parameters `min` and `max` default to `-Inf` and `Inf` respectively.
- `categorical()` filters by categorical columns. Parameter `col` is the name of the categorical column (of type character or factor) to filter by. Parameter `values` is a vector of allowed values in that column to filter for.

## Value

a new MaveDB filter object

## Examples

```
## Not run:
mave <- new.rapimave()
data <- mave$getScores("SCS000001A.2")
mfilter <- new.mave.filter(data)
filter <- with(mfilter,
  position(min=5,multi="all") & residues(to="Ala",multi="all") & numerical("score",min=0.1)
)
filtered.data <- data[filter,]

## End(Not run)
```

---

new.rapimave

*MaveDB R-API client constructor*


---

## Description

Creates an object of type rapimave.

## Usage

```
new.rapimave(
  baseURL = "https://www.mavedb.org/api/",
  certifySSL = FALSE,
  encoding = "UTF-8"
)
```

## Arguments

baseURL	MaveDB API base-URL. Defaults to "https://www.mavedb.org/api/"
certifySSL	Logical indicating whether to certify SSL certificates when querying the database. Defaults to FALSE.
encoding	A string specifying the expected character encoding used by the webservice. Defaults to "UTF-8"

## Details

The resulting object offers the following functions:

- `getAllUsers()` returns a list of all user objects. See [new.user](#) for available methods on user objects.
- `getUser(username)` returns the user object for the given username. Usernames are usually ORCID IDs. See <https://orcid.org/> for information on ORCID. See [new.user](#) for available methods on user objects.
- `findExperimentSets(filter)` returns a list of all ExperimentSet objects that match the given filter word. See [new.experimentSet](#) for available methods on ExperimentSet objects.

- `getAllExperimentSets()` returns a list of all `ExperimentSet` objects. See [new.experimentSet](#) for available methods on `ExperimentSet` objects.
- `getExperimentSet(urn)` returns the `ExperimentSet` object for the given URN. `ExperimentSet` URNs usually follow the syntax `/^urn:mavedb:\d+$/`. See [new.experimentSet](#) for available methods on `ExperimentSet` objects.
- `getAllExperiments()` returns a list of all `Experiment` objects. See [new.experiment](#) for available methods on `Experiment` objects.
- `findExperiments(filter)` returns a list of all `Experiment` objects that match the given filter word. See [new.experiment](#) for available methods on `Experiment` objects.
- `getExperiment(urn)` returns the `Experiment` object for the given URN. `Experiment` URNs usually follow the syntax `/^urn:mavedb:\d+-\w+$/`. See [new.experiment](#) for available methods on `Experiment` objects.
- `getAllScoreSets()` returns a list of all `ScoreSet` objects. See [new.scoreSet](#) for available methods on `ScoreSet` objects.
- `findScoreSets(filter)` returns a list of all `ScoreSet` objects that match the given filter word. See [new.scoreSet](#) for available methods on `ScoreSet` objects.
- `getScoreSet(urn)` returns the `ScoreSet` object for the given URN. `ScoreSet` URNs usually follow the syntax `/^urn:mavedb:\d+-\w+-\d+$/`. See [new.scoreSet](#) for available methods on `ScoreSet` objects.
- `getScores(urn)` returns a `data.frame` with the scores for the given `ScoreSet` URN
- `getCounts(urn)` returns a `data.frame` with the counts for the given `ScoreSet` URN
- `getMetadata(urn)` returns a `data.frame` with the metadata for the given `ScoreSet` URN
- `getKeywords()` returns all currently used keywords across `MaveDB`.
- `getAllPubmed()` returns all currently used Pubmed IDs across `MaveDB`.
- `getAllDOI()` returns all currently used DOIs across `MaveDB`.
- `getAllSRA()` returns all currently used SRA IDs across `MaveDB`.
- `getAllRefseq()` returns all currently used Refseq IDs across `MaveDB`.
- `getAllUniprot()` returns all currently used Uniprot IDs across `MaveDB`.
- `getAllEnsembl()` returns all currently used Ensembl IDs across `MaveDB`.
- `getAllAssemblies()` returns all currently used Genome assemblies across `MaveDB`.
- `getAllTargets()` returns all currently used target sequences across `MaveDB`.
- `getReferenceGenomes()` returns all currently used reference genomes across `MaveDB`.

## Value

`MaveDB` R-API object with functions accessible via the `$` operator.

## Examples

```
## Not run:
mave <- new.rapimave()

## End(Not run)
```

---

new.refmap

*New R-API MaveDB ReferenceMap*


---

### Description

Internal constructor that creates an immutable new ReferenceMap object. This function is not exported and should not be called outside of the rapimave package itself. ReferenceMap objects are generated by the getReferenceMap() method of a target object (which itself is generated using the getTarget() method on a scoreset object). However, this section explains the functions that are available for these objects.

### Usage

```
new.refmap(data)
```

### Details

A ScoreSet object offers the following getter functions:

- getGenome() returns the genome object for this reference map

### Value

a new R-API MaveDB ReferenceMap object.

---

new.scoreSet

*New R-API MaveDB ScoreSet*


---

### Description

Internal constructor that creates an immutable new ScoreSet object. This function is not exported and should not be called outside of the rapimave package itself. ScoreSet objects are generated by the getAllScoreSets() and getScoreSet() methods of a rapimave object (which itself is generated using the new.rapimave() constructor). However, this section explains the functions that are available for these objects.

### Usage

```
new.scoreSet(data)
```



## Details

A ScoreSet object offers the following getter functions:

- `getContributors()` returns a list of users that contributed to this ScoreSet.
- `getURN()` returns the URN of this ScoreSet.
- `getCreationDate()` returns the date when this record was originally created.
- `getModificationDate()` returns the date when this record was last modified.
- `getPublicationDate()` returns the date when this record was published.
- `getCreatedBy()` returns the user who originally created this record.
- `getModifiedBy()` returns the user who last modified this record.
- `getMetadata()` returns the metadata for this record.
- `getTitle()` returns the title of this record.
- `getShortDescription()` returns a short description text of this record.
- `getAbstract()` returns the abstract text of this record.
- `getMethods()` returns the methods description text of this record.
- `getKeywords()` returns the list of keywords associated with this record.
- `getDOIs()` returns the list of Digital Object Identifiers (DOIs) associated with this record.
- `getXRefPubmed()` returns cross-references to Pubmed.
- `getTarget()` returns the Target object for this scoreset, which describes the molecule that that was targeted in the experiment.
- `getLicence()` returns the licencing information governing the use of this ScoreSet.
- `getCurrentVersion()` If a more up-to-date version of this ScoreSet exists, that has since replaced it, this function will return the accession of the most recent version.
- `getPreviousVersion()` If this ScoreSet replaces an older ScoreSet, this function returns the accession of that older ScoreSet.
- `getNextVersion()` If this ScoreSet was replaced by newer ScoreSet, this function returns the accession of that newer ScoreSet in versioning history. To obtain the most recent version, use `getCurrentVersion()`.
- `getCountColumns()` returns the column names in the count table for this ScoreSet.
- `getScoreColumns()` returns the column names in the score table for this ScoreSet.
- `getVariantCount()` returns the number of variants in this scoreset, i.e. the number of rows to be expected in the scores and counts tables.
- `getExperiment()` returns the URN of the experiment record to which this scoreset belongs.
- `isMetaAnalysis()` returns whether this dataset represents a meta-analysis of existing data.
- `getDataUsagePolicy()` returns any data usage policy text for this ScoreSet.

## Value

a new R-API MaveDB ScoreSet object.

## Examples

```
## Not run:
mave <- new.rapimave()
set1 <- mave$getScoreSet("urn:mavedb:00000001-a-1")
scoreColNames <- set1$getScoreColumns()

## End(Not run)
```

---

new.target

*New R-API MaveDB Target*


---

## Description

Internal constructor that creates an immutable new Target object. This function is not exported and should not be called outside of the rapimave package itself. Target objects are generated by the `getTarget()` method of a scoreset object (which itself is generated using the `getAllScoreSets()` and `getScoreSet()` methods of a rapimave object). However, this section explains the functions that are available for these objects.

## Usage

```
new.target(data)
```

## Details

A ScoreSet object offers the following getter functions:

- `getName()` returns the name of the target
- `getSequence()` returns wildtype sequence of the target
- `getSequenceType()` returns whether the sequence is at DNA or Protein level
- `getXrefUniprot()` returns a Xref object for Uniprot
- `getXrefEnsembl()` returns a Xref object for Ensembl
- `getXrefRefseq()` returns a Xref object for Refseq
- `getReferenceMaps()` returns a list of reference map objects that describe how this target maps to different reference genomes.
- `getScoreset()` returns the URN of the scoreset to which this target belongs.

## Value

a new R-API MaveDB Target object.

---

`new.user`*New R-API MaveDB User*

---

## Description

Internal constructor that creates an immutable new user object. This function is not exported and should not be called outside of the rapimave package itself. User objects are generated by the `getAllUsers()` and `getUser()` methods of a rapimave object (which itself is generated using the `new.rapimave()` constructor). However, this section explains the functions that are available for these objects.

## Usage

```
new.user(data)
```

## Details

A user object offers the following getter functions:

- `getUsername()` return the username.
- `getFirstName()` return the user's first name.
- `getLastName()` return the user's last name.
- `getDisplayName()` return the user's preferred display name.
- `getExperimentSets()` return the ExperimentSets attributed to the user.
- `getExperiments()` return the Experiments attributed to the user.
- `getScoreSets()` return the ScoreSets attributed to the user.

## Value

a new R-API MaveDB user object.

## Examples

```
## Not run:
mave <- new.rapimave()
user <- mave$getUser("0000-0003-1474-605X")
cat(user$getFirstName(),user$getLastName())

## End(Not run)
```

---

new.xref	<i>New R-API MaveDB cross-reference</i>
----------	---

---

### Description

Internal constructor that creates an immutable new Xref object. This function is not exported and should not be called outside of the rapimave package itself. Xref objects are generated by various methods of a rapimave object (which itself is generated using the new.rapimave() constructor). However, this section explains the functions that are available for these objects.

### Usage

```
new.xref(data)
```

### Details

A ScoreSet object offers the following getter functions:

- getID() returns the Identifier or Accession of this cross-reference
- getURL() returns the database URL
- getDB() returns the database name
- getDBVersion() returns the database version
- getOffset() returns the numerical sequence offset compared to the database entry. Only applicable for some databases. Returns NULL where not defined.

### Value

a new R-API MaveDB Xref object.

---

print.rapimave	<i>Print R-API MaveDB object</i>
----------------	----------------------------------

---

### Description

Prints a human-readable summary of a R-API MaveDB object.

### Usage

```
## S3 method for class 'rapimave'
print(obj)
```

### Arguments

obj	the object to print
-----	---------------------

**Details**

Prints a list of available functions for the object.

**Examples**

```
## Not run:  
mave <- new.rapimave()  
print(mave)  
  
## End(Not run)
```

---

```
print.rapimaveExperiment  
      Print R-API MaveDB Experiment
```

---

**Description**

Prints a human-readable summary of a R-API MaveDB Experiment object:

**Usage**

```
## S3 method for class 'rapimaveExperiment'  
print(obj)
```

**Arguments**

obj                    the Experiment object to print

**Examples**

```
## Not run:  
mave <- new.rapimave()  
print(mave$getExperiment("urn:mavedb:00000001-a"))  
  
## End(Not run)
```

---

```
print.rapimaveExpSet    Print R-API MaveDB ExperimentSet
```

---

**Description**

Prints a human-readable summary of a R-API MaveDB ExperimentSet object:

**Usage**

```
## S3 method for class 'rapimaveExpSet'  
print(obj)
```

**Arguments**

obj                      the ExperimentSet object to print

**Examples**

```
## Not run:
mave <- new.rapimave()
print(mave$getExperimentSet("urn:mavedb:00000001"))

## End(Not run)
```

---

print.rapimaveGenome    *Print R-API MaveDB Genome object*

---

**Description**

Prints a human-readable summary of a R-API MaveDB genome object:

**Usage**

```
## S3 method for class 'rapimaveGenome'
print(obj)
```

**Arguments**

obj                      the object to print

---

print.rapimaveRefmap    *Print R-API MaveDB reference map object*

---

**Description**

Prints a human-readable summary of a R-API MaveDB reference map object:

**Usage**

```
## S3 method for class 'rapimaveRefmap'
print(obj)
```

**Arguments**

obj                      the object to print

---

```
print.rapimaveScoreSet
```

*Print R-API MaveDB ScoreSet*

---

### Description

Prints a human-readable summary of a R-API MaveDB ScoreSet object:

### Usage

```
## S3 method for class 'rapimaveScoreSet'  
print(obj)
```

### Arguments

obj                    the ScoreSet object to print

### Examples

```
## Not run:  
mave <- new.rapimave()  
print(mave$getScoreSet("SCS000001A.2"))  
  
## End(Not run)
```

---

```
print.rapimaveTarget    Print R-API MaveDB target object
```

---

### Description

Prints a human-readable summary of a R-API MaveDB target object:

### Usage

```
## S3 method for class 'rapimaveTarget'  
print(obj)
```

### Arguments

obj                    the object to print

---

<code>print.rapimaveUser</code>	<i>Print R-API MaveDB User</i>
---------------------------------	--------------------------------

---

**Description**

Prints a human-readable summary of a R-API MaveDB User object:

**Usage**

```
## S3 method for class 'rapimaveUser'  
print(obj)
```

**Arguments**

`obj`                      the User object to print

**Examples**

```
## Not run:  
mave <- new.rapimave()  
print(mave$getUser("0000-0003-1474-605X"))  
  
## End(Not run)
```

---

<code>print.rapimaveXref</code>	<i>Print R-API MaveDB cross-reference</i>
---------------------------------	---

---

**Description**

Prints a human-readable summary of a R-API MaveDB cross-reference object:

**Usage**

```
## S3 method for class 'rapimaveXref'  
print(obj)
```

**Arguments**

`obj`                      the object to print



# Index

## \*Topic **MaveDB**

- new.rapimave, [6](#)
- print.rapimave, [12](#)
- print.rapimaveExperiment, [13](#)
- print.rapimaveExpSet, [13](#)
- print.rapimaveGenome, [14](#)
- print.rapimaveRefmap, [14](#)
- print.rapimaveScoreSet, [15](#)
- print.rapimaveTarget, [15](#)
- print.rapimaveUser, [16](#)
- print.rapimaveXref, [16](#)

## \*Topic **ScoreSet**

- print.rapimaveScoreSet, [15](#)

## \*Topic **print**

- print.rapimaveGenome, [14](#)
- print.rapimaveRefmap, [14](#)
- print.rapimaveTarget, [15](#)
- print.rapimaveXref, [16](#)

- new.experiment, [2](#), [7](#)
- new.experimentSet, [3](#), [6](#), [7](#)
- new.genome, [4](#)
- new.mave.filter, [5](#)
- new.rapimave, [6](#)
- new.refmap, [8](#)
- new.scoreSet, [7](#), [8](#)
- new.target, [10](#)
- new.user, [6](#), [11](#)
- new.xref, [12](#)

- print.rapimave, [12](#)
- print.rapimaveExperiment, [13](#)
- print.rapimaveExpSet, [13](#)
- print.rapimaveGenome, [14](#)
- print.rapimaveRefmap, [14](#)
- print.rapimaveScoreSet, [15](#)
- print.rapimaveTarget, [15](#)
- print.rapimaveUser, [16](#)
- print.rapimaveXref, [16](#)