Package 'gemma.R'

April 10, 2023

Title A wrapper for Gemma's Restful API to access curated gene expression data and differential expression analyses

Version 1.0.1

Description Low- and high-level wrappers for Gemma's RESTful API. They enable access to curated expression and differential expression data from over 10,000 published studies. Gemma is a web site, database and a set of tools for the meta-analysis, re-use and sharing of genomics data, currently primarily targeted at the analysis of gene expression profiles.

URL https://pavlidislab.github.io/gemma.R/, https://github.com/PavlidisLab/gemma.R

License Apache License (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

BugReports https://github.com/PavlidisLab/gemma.R/issues

Imports magrittr, glue, memoise, jsonlite, data.table, rlang, lubridate, utils, stringr, SummarizedExperiment, Biobase, tibble, tidyr, S4Vectors, httr, rappdirs, bit64

Suggests testthat (>= 2.0.0), rmarkdown, knitr, dplyr, covr, ggplot2, ggrepel, BiocStyle, microbenchmark, magick, purrr, pheatmap, viridis

Config/testthat/edition 2

VignetteBuilder knitr

biocViews Software, DataImport, Microarray, SingleCell, ThirdPartyClient, DifferentialExpression, GeneExpression, Bayesian, Annotation, ExperimentalDesign, Normalization, BatchEffect, Preprocessing

git_url https://git.bioconductor.org/packages/gemma.R

git_branch RELEASE_3_16

git_last_commit 43adbd5

git_last_commit_date 2023-01-24

Date/Publication 2023-04-10

Author Javier Castillo-Arnemann [aut]
(<https://orcid.org/0000-0002-5626-9004>),
Jordan Sicherman [aut] (<https://orcid.org/0000-0001-8160-4567>),
Ogan Mancarci [cre, aut] (<https://orcid.org/0000-0002-1452-0889>),
Guillaume Poirier-Morency [aut]
(<https://orcid.org/0000-0002-6554-0441>)

Maintainer Ogan Mancarci <ogan.mancarci@gmail.com>

R topics documented:

forget_gemma_memoised . . . 3
gemma.R . . . 3
gemma_call . . . 4
get_datasets_by_ids . . . 4
get_dataset_annotations . . . 7
get_dataset_design . . . 8
get_dataset_differential_expression_analyses . . . 9
get_dataset_expression . . . 10
get_dataset_object . . . 11
get_dataset_platforms . . . 12
get_dataset_samples . . . 14
get_differential_expression_values . . . 15
get_genes . . . 16
get_gene_go_terms . . . 18
get_gene_locations . . . 19
get_gene_probes . . . 20
get_platforms_by_ids . . . 21
get_platform_annotations . . . 23
get_platform_datasets . . . 24
get_platform_element_genes . . . 26
get_taxa . . . 27
get_taxon_datasets . . . 28
search_annotations . . . 30
search_datasets . . . 31
search_gemma . . . 34
set_gemma_user . . . 35

Index **36**

`forget_gemma_memoised` *Clear gemma.R cache*

Description

Forget past results from memoised calls to the Gemma API (ie. using functions with memoised = TRUE)

Usage

```
forget_gemma_memoised()
```

Value

TRUE to indicate cache was cleared.

Examples

```
forget_gemma_memoised()
```

`gemma.R` *gemma.R package: Access curated gene expression data and differential expression analyses*

Description

This package contains wrappers and convenience functions for Gemma's RESTful API that enables access to curated expression and differential expression data from over 15,000 published studies (as of mid-2022). Gemma (https://gemma.msl.ubc.ca) is a web site, database and a set of tools for the meta-analysis, re-use and sharing of transcriptomics data, currently primarily targeted at the analysis of gene expression profiles.

Details

Most users will want to start with the high-level functions like `get_dataset_object`, `get_differential_expression_valu` and `get_platform_annotations` Additional lower-level methods are available that directly map to the Gemma RESTful API methods.

For more information and detailed usage instructions check the README, the function reference and the vignette.

All software-related questions should be posted to the Bioconductor Support Site: `https://support.bioconductor.org`

Author(s)

Javier Castillo-Arnemann, Jordan Sicherman, Ogan Mancarci, Guillaume Poirier-Morency

References

Lim, N. et al., Curation of over 10 000 transcriptomic studies to enable data reuse, Database, 2021. https://doi.org/10.1093/database/baab006

`gemma_call` *Custom gemma call*

Description

A minimal function to create custom calls. Can be used to acquire unimplemented endpoints and/or raw output without any processing. Refer to the API documentation.

Usage

```
gemma_call(call, ..., json = TRUE)
```

Arguments

`call`	Gemma API endpoint.
`...`	parameters included in the call
`json`	If `TRUE` will parse the content as a list

Value

A list if `json = TRUE` and an httr response if `FALSE`

Examples

```
# get singular value decomposition for the dataset
gemma_call('datasets/{dataset}/svd',dataset = 1)
```

`get_datasets_by_ids` *Retrieve datasets by their identifiers*

Description

Retrieve datasets by their identifiers

Usage

```
get_datasets_by_ids(
  datasets = NA_character_,
  offset = 0L,
  limit = 20L,
  sort = "+id",
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`datasets`	Numerical dataset identifiers or dataset short names. If not specified, all datasets will be returned instead
`offset`	The offset of the first retrieved result.
`limit`	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.
`sort`	Order results by the given property and direction. The '+' sign indicate ascending order whereas the '-' indicate descending.
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the queried dataset(s). A list if `raw = TRUE`. Returns an empty list if no datasets matched. A successful response may contain 'Geeq' information, which aims to provide a unified metric to measure experiments by the quality of their data, and their suitability for use in Gemma. You can read more about the geeq properties here.

The fields of the output data.table are:

- `experiment.ShortName`: Shortname given to the dataset within Gemma. Often corresponds to accession ID

- `experiment.Name`: Full title of the dataset
- `experiment.ID`: Internal ID of the dataset.
- `experiment.Description`: Description of the dataset
- `experiment.Troubled`: Did an automatic process within gemma or a curator mark the dataset as "troubled"
- `experiment.Accession`: Accession ID of the dataset in the external database it was taken from
- `experiment.Database`: The name of the database where the dataset was taken from
- `experiment.URI`: URI of the original database
- `experiment.SampleCount`: Number of samples in the dataset
- `experiment.batchEffect`: A text field describing whether the dataset has batch effects
- `geeq.batchCorrected`: Whether batch correction has been performed on the dataset.
- `geeq.batchConfound`: 0 if batch info isn't available, -1 if batch counfoud is detected, 1 if batch information is available and no batch confound found
- `geeq.batchEffect`: -1 if batch p value < 0.0001, 1 if batch p value > 0.1, 0 if otherwise and when there is no batch information is available or when the data is confounded with batches.
- `geeq.rawData`: -1 if no raw data available, 1 if raw data was available. When available, Gemma reprocesses raw data to get expression values and batches
- `geeq.qScore`: Data quality score given to the dataset by Gemma.
- `geeq.sScore`: Suitability score given to the dataset by Gemma. Refers to factors like batches, platforms and other aspects of experimental design
- `taxon.Name`: Name of the species
- `taxon.Scientific`: Scientific name for the taxon
- `taxon.ID`: Internal identifier given to the species by Gemma
- `taxon.NCBI`: NCBI ID of the taxon
- `taxon.Database.Name`: Underlying database used in Gemma for the taxon
- `taxon.Database.ID`: ID of the underyling database used in Gemma for the taxon

Examples

```
get_datasets_by_ids("GSE2018")
get_datasets_by_ids(c("GSE2018", "GSE2872"))
```

`get_dataset_annotations` *Retrieve the annotations analysis of a dataset*

Description

Retrieve the annotations analysis of a dataset

Usage

```
get_dataset_annotations(
  dataset,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`dataset`	A numerical dataset identifier or a dataset short name
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the annotations of the queried dataset. A list if `raw = TRUE`.A `404 error` if the given identifier does not map to any object.

The fields of the output data.table are:

- `class.Type`: Type of the annotation class
- `class.Name`: Name of the annotation class (e.g. organism part)
- `class.URI`: URI for the annotation class
- `term.Name`: Name of the annotation term (e.g. lung)
- `term.URI`: URI for the annotation term

Examples

```
get_dataset_annotations("GSE2018")
```

`get_dataset_design` *Retrieve the design of a dataset*

Description

Retrieve the design of a dataset

Usage

```
get_dataset_design(
  dataset,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`dataset`	A numerical dataset identifier or a dataset short name
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table of the design matrix for the queried dataset. A `404 error` if the given identifier does not map to any object

Examples

```
head(get_dataset_design("GSE2018"))
```

`get_dataset_differential_expression_analyses` *Retrieve the differential analyses of a dataset*

Description

Retrieve the differential analyses of a dataset

Usage

```
get_dataset_differential_expression_analyses(
  dataset,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`dataset`	A numerical dataset identifier or a dataset short name
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the differential expression analysis of the queried dataset. Note that this funciton does not return differential expression values themselves. Use `get_differential_expression_values` to get differential expression values (see examples).

The fields of the output data.table are:

- `result.ID`: Result set ID of the differential expression analysis. May represent multiple factors in a single model.

- `contrast.ID`: Id of the specific contrast factor. Together with the result.ID they uniquely represent a given contrast.
- `experiment.ID`: Id of the source experiment
- `baseline.category`: Category for the contrast
- `baseline.categoryURI`: URI for the baseline category
- `baseline.factorValue`: Factor value assigned as the baseline in the contrast. Typically represent control samples
- `baseline.factorValueURI`: URI for the baseline.factorValue
- `experimental.factorValue`: Factor value assigned to the experimental group.
- `experimental.factorValueURI`: URI for the experimental.factorValue
- `subsetFactor.subset`: TRUE if the result set belong to a subset, FALSE if not. Subsets are created when performing differential expression to avoid unhelpful comparisons.
- `subsetFactor.category`: Category of the subset
- `subsetFactor.categoryURI`: URI of the subset category
- `subsetFactor.factorValue`: Factor Value of the subset
- `subsetFactor.factorValueURI`: URI of the subset factor value
- `probes.Analyzed`: Number of probesets represented in the contrast
- `genes.Analyzed`: Number of genes represented in the contrast
- `platform.ID`: Platform id for the contrast

Examples

```
result <- get_dataset_differential_expression_analyses("GSE2018")
get_differential_expression_values(resultSet = result$result.ID)
```

---

get_dataset_expression

Retrieve the expression data of a dataset

Description

Retrieve the expression data of a dataset

Usage

```
get_dataset_expression(
  dataset,
  filter = FALSE,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

dataset	A numerical dataset identifier or a dataset short name
filter	The filtered version (filter = TRUE) corresponds to what is used in most Gemma analyses, removing some probes/elements. Unfiltered includes all elements.
raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

If raw is FALSE (default), a data table of the expression matrix for the queried dataset. If raw is TRUE, returns the binary file in raw form.

Examples

```
get_dataset_expression("GSE2018")
```

---

get_dataset_object *Compile gene expression data and metadata*

Description

Return an annotated Bioconductor-compatible data structure or a long form tibble of the queried dataset, including expression data and the experimental design.

Usage

```
get_dataset_object(
  dataset,
  filter = FALSE,
  type = "se",
  memoised = getOption("gemma.memoised", FALSE)
)
```

Arguments

dataset	A dataset identifier.
filter	The filtered version corresponds to what is used in most Gemma analyses, removing some probes/elements. Unfiltered includes all elements.
type	"se"for a SummarizedExperiment or "eset" for Expression Set. We recommend using SummarizedExperiments which are more recent. See the Summarized experiment vignette or the ExpressionSet vignette for more details.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.

Value

A SummarizedExperiment, ExpressionSet or tibble containing metadata and expression data for the queried dataset.

Examples

```
get_dataset_object("GSE2018")
```

---

get_dataset_platforms *Retrieve the platform of a dataset*

Description

Retrieve the platform of a dataset

Usage

```
get_dataset_platforms(
  dataset,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

dataset	A numerical dataset identifier or a dataset short name
raw	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.

memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the platform(s). A list if raw = TRUE. A 404 error if the given identifier does not map to any object

The fields of the output data.table are:

- platform.ID: Internal identifier of the platform
- platform.ShortName: Shortname of the platform.
- platform.Name: Full name of the platform.
- platform.Description: Free text description of the platform
- platform.Troubled: Whether or not the platform was marked "troubled" by a Gemma process or a curator
- platform.ExperimentCount: Number of experiments using the platform within Gemma
- platform.Type: Technology type for the platform.
- taxon.Name: Name of the species platform was made for
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal identifier given to the species by Gemma
- taxon.NCBI: NCBI ID of the taxon
- taxon.Database.Name: Underlying database used in Gemma for the taxon
- taxon.Database.ID: ID of the underyling database used in Gemma for the taxon

Examples

```
get_dataset_platforms("GSE2018")
```

`get_dataset_samples` *Retrieve the samples of a dataset*

Description

Retrieve the samples of a dataset

Usage

```
get_dataset_samples(
  dataset,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`dataset`	A numerical dataset identifier or a dataset short name
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the samples of the queried dataset. A list if `raw = TRUE`. A `404 error` if the given identifier does not map to any object.

The fields of the output data.table are:

- `sample.Name`: Internal name given to the sample.
- `sample.ID`: Internal ID of the sample
- `sample.Description`: Free text description of the sample
- `sample.Outlier`: Whether or not the sample is marked as an outlier

- `sample.Accession`: Accession ID of the sample in it's original database
- `sample.Database`: Database of origin for the sample
- `sample.Characteristics`: Characteristics of the sample. This field is a data table
- `sample.FactorValues`: Experimental factor values of the sample. This field is a data table

Examples

```
head(get_dataset_samples("GSE2018"))
```

`get_differential_expression_values`
Retrieve differential expression results

Description

Retrieves the differential expression result set(s) associated with the dataset. To get more information about the contrasts in individual resultSets and annotation terms associated them, use `get_dataset_differential_expression_analyses()`

Usage

```
get_differential_expression_values(
  dataset = NA_character_,
  resultSet = NA_integer_,
  readableContrasts = FALSE,
  memoised = getOption("gemma.memoised", FALSE)
)
```

Arguments

`dataset`	A dataset identifier.
`resultSet`	A resultSet identifier.
`readableContrasts`	If `FALSE` (default), the returned columns will use internal constrasts IDs as names. Details about the contrasts can be accessed using `get_dataset_differential_expression_ana` If TRUE IDs will be replaced with human readable contrast information.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.

Details

In Gemma each result set corresponds to the estimated effects associated with a single factor in the design, and each can have multiple contrasts (for each level compared to baseline). Thus a dataset with a 2x3 factorial design will have two result sets, one of which will have one contrast, and one having two contrasts.

The methodology for differential expression is explained in Curation of over 10000 transcriptomic studies to enable data reuse. Briefly, differential expression analysis is performed on the dataset based on the annotated experimental design with up two three potentially nested factors. Gemma attempts to automatically assign baseline conditions for each factor. In the absence of a clear control condition, a baseline is arbitrarily selected. A generalized linear model with empirical Bayes shrinkage of t-statistics is fit to the data for each platform element (probe/gene) using an implementation of the limma algorithm. For RNA-seq data, we use weighted regression, applying the voom algorithm to compute weights from the mean–variance relationship of the data. Contrasts of each condition are then computed compared to the selected baseline. In some situations, Gemma will split the data into subsets for analysis. A typical such situation is when a 'batch' factor is present and confounded with another factor, the subsets being determined by the levels of the confounding factor.

Value

A list of data tables with differential expression values per result set.

Examples

```
get_differential_expression_values("GSE2018")
```

---

get_genes *Retrieve genes matching a gene identifier*

Description

Retrieve genes matching a gene identifier

Usage

```
get_genes(
  genes,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

genes	An ensembl gene identifier which typically starts with ensg or an ncbi gene identifier or an official gene symbol approved by hgnc
raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the querried gene(s) A list if raw = TRUE.

The fields of the output data.table are:

- gene.Symbol: Symbol for the gene
- gene.Ensembl: Ensembl ID for the gene
- gene.NCBI: NCBI id for the gene
- gene.Name: Name of the gene
- gene.MFX.Rank: Multifunctionality rank for the gene
- taxon.Name: Name of the species
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal identifier given to the species by Gemma
- taxon.NCBI: NCBI ID of the taxon
- taxon.Database.Name: Underlying database used in Gemma for the taxon
- taxon.Database.ID: ID of the underyling database used in Gemma for the taxon

Examples

```
get_genes("DYRK1A")
get_genes(c("DYRK1A", "PTEN"))
```

`get_gene_go_terms` *Retrieve the GO terms associated to a gene*

Description

Retrieve the GO terms associated to a gene

Usage

```
get_gene_go_terms(
  gene,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`gene`	An ensembl gene identifier which typically starts with ensg or an ncbi gene identifier or an official gene symbol approved by hgnc
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the GO terms assigned to the queried gene. A list if `raw = TRUE`. A `404 error` if the given identifier does not map to any object.

The fields of the output data.table are:

- `term.Name`: Name of the term
- `term.ID`: ID of the term
- `term.URI`: URI of the term

Examples

```
get_gene_go_terms("DYRK1A")
```

`get_gene_locations` *Retrieve the physical locations of a given gene*

Description

Retrieve the physical locations of a given gene

Usage

```
get_gene_locations(
  gene,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`gene`	An ensembl gene identifier which typically starts with ensg or an ncbi gene identifier or an official gene symbol approved by hgnc
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
`memoised`	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.
`file`	The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.
`overwrite`	Whether or not to overwrite if a file exists at the specified filename.
`attributes`	If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the physical location of the queried gene. A list if `raw = TRUE`. A `404 error` if the given identifier does not map to any object.

The fields of the output data.table are:

- `chromosome`: Name of the chromosome the gene is located

- strand: Which strand the gene is located
- nucleotide: Nucleotide number for the gene
- length: Gene length
- taxon.name: Name of the taxon
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal ID for the taxon given by Gemma
- taxon.NCBI: NCBI ID for the taxon
- taxon.Database.Name: Name of the database used in Gemma for the taxon

Examples

```
get_gene_locations("DYRK1A")
```

`get_gene_probes` *Retrieve the probes associated to a genes*

Description

Retrieve the probes associated to a genes

Usage

```
get_gene_probes(
  gene,
  offset = 0L,
  limit = 20L,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

gene	An ensembl gene identifier which typically starts with ensg or an ncbi gene identifier or an official gene symbol approved by hgnc
offset	The offset of the first retrieved result.
limit	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with offset and the totalElements attribute in the output to compile all data if needed.
raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.

memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the probes representing a gene across all platrofms. A list if raw = TRUE. A 404 error if the given identifier does not map to any genes.

The fields of the output data.table are:

- mapping.name: Name of the mapping. Typically the probeset name
- mappping.description: A free text field providing optional information about the mapping
- platform.ShortName: Shortname of the platform given by Gemma. Typically the GPL identifier.
- platform.Name: Full name of the platform
- platform.ID: Id number of the platform given by Gemma
- platform.Taxon: Species the platform was designed for
- platform.TaxonID: Id number of the species given by Gemma
- platform.Type: Type of the platform.
- platform.Description: Free text field describing the platform.
- platform.Troubled: Whether the platform is marked as troubled by a Gemma curator.

Examples

```
get_gene_probes("DYRK1A")
```

---

get_platforms_by_ids *Retrieve all platforms matching a set of platform identifiers*

Description

Retrieve all platforms matching a set of platform identifiers

Usage

```
get_platforms_by_ids(
  platforms = NA_character_,
  offset = 0L,
  limit = 20L,
  sort = "+id",
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

platforms	Platform numerical identifiers or platform short names. If not specified, all platforms will be returned instead
offset	The offset of the first retrieved result.
limit	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with offset and the totalElements attribute in the output to compile all data if needed.
sort	Order results by the given property and direction. The '+' sign indicate ascending order whereas the '-' indicate descending.
raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the platform(s). A list if raw = TRUE. A 404 error if the given identifier does not map to any object

The fields of the output data.table are:

- platform.ID: Internal identifier of the platform
- platform.ShortName: Shortname of the platform.
- platform.Name: Full name of the platform.

- platform.Description: Free text description of the platform
- platform.Troubled: Whether or not the platform was marked "troubled" by a Gemma process or a curator
- platform.ExperimentCount: Number of experiments using the platform within Gemma
- platform.Type: Technology type for the platform.
- taxon.Name: Name of the species platform was made for
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal identifier given to the species by Gemma
- taxon.NCBI: NCBI ID of the taxon
- taxon.Database.Name: Underlying database used in Gemma for the taxon
- taxon.Database.ID: ID of the underyling database used in Gemma for the taxon

Examples

```
get_platforms_by_ids("GPL1355")
get_platforms_by_ids(c("GPL1355", "GPL96"))
```

---

get_platform_annotations

Retrieve Platform Annotations by Gemma

Description

Gets Gemma's platform annotations including mappings of microarray probes to genes.

Usage

```
get_platform_annotations(
  platform,
  annotType = c("noParents", "allParents", "bioProcess"),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  memoised = getOption("gemma.memoise", FALSE),
  unzip = FALSE
)
```

Arguments

platform	A platform identifier @seealso getPlatforms
annotType	Which GO terms should the output include
file	Where to save the annotation file to, or empty to just load into memory
overwrite	Whether or not to overwrite an existing file
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
unzip	Whether or not to unzip the file (if @param file is not empty)

Value

A table of annotations

- `ProbeName`: Probeset names provided by the platform. Gene symbols for generic annotations
- `GeneSymbols`: Genes that were found to be aligned to the probe sequence. Note that it is possible for probes to be non-specific. Alignment to multiple genes are indicated with gene symbols separated by "|"s
- `GeneNames`: Name of the gene
- `GOTerms`: GO Terms associated with the genes. `annotType` argument can be used to choose which terms should be included.
- `GemmaIDs` and `NCBIids`: respective IDs for the genes.

Examples

```
head(get_platform_annotations("GPL96"))
head(get_platform_annotations('Generic_human'))
```

---

get_platform_datasets *Retrieve all experiments within a given platform*

Description

Retrieve all experiments within a given platform

Usage

```
get_platform_datasets(
  platform,
  offset = 0L,
  limit = 20L,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`platform`	A platform numerical identifier or a platform short name
`offset`	The offset of the first retrieved result.
`limit`	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.

raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the queried dataset(s). A list if raw = TRUE. Returns an empty list if no datasets matched. A successful response may contain 'Geeq' information, which aims to provide a unified metric to measure experiments by the quality of their data, and their suitability for use in Gemma. You can read more about the geeq properties here.

The fields of the output data.table are:

- experiment.ShortName: Shortname given to the dataset within Gemma. Often corresponds to accession ID
- experiment.Name: Full title of the dataset
- experiment.ID: Internal ID of the dataset.
- experiment.Description: Description of the dataset
- experiment.Troubled: Did an automatic process within gemma or a curator mark the dataset as "troubled"
- experiment.Accession: Accession ID of the dataset in the external database it was taken from
- experiment.Database: The name of the database where the dataset was taken from
- experiment.URI: URI of the original database
- experiment.SampleCount: Number of samples in the dataset
- experiment.batchEffect: A text field describing whether the dataset has batch effects
- geeq.batchCorrected: Whether batch correction has been performed on the dataset.
- geeq.batchConfound: 0 if batch info isn't available, -1 if batch counfoud is detected, 1 if batch information is available and no batch confound found
- geeq.batchEffect: -1 if batch p value < 0.0001, 1 if batch p value > 0.1, 0 if otherwise and when there is no batch information is available or when the data is confounded with batches.
- geeq.rawData: -1 if no raw data available, 1 if raw data was available. When available, Gemma reprocesses raw data to get expression values and batches
- geeq.qScore: Data quality score given to the dataset by Gemma.

- `geeq.sScore`: Suitability score given to the dataset by Gemma. Refers to factors like batches, platforms and other aspects of experimental design
- `taxon.Name`: Name of the species
- `taxon.Scientific`: Scientific name for the taxon
- `taxon.ID`: Internal identifier given to the species by Gemma
- `taxon.NCBI`: NCBI ID of the taxon
- `taxon.Database.Name`: Underlying database used in Gemma for the taxon
- `taxon.Database.ID`: ID of the underyling database used in Gemma for the taxon

Examples

```
head(get_platform_datasets("GPL1355"))
```

`get_platform_element_genes` *Retrieve the genes associated to a probe in a given platform*

Description

Retrieve the genes associated to a probe in a given platform

Usage

```
get_platform_element_genes(
  platform,
  probe,
  offset = 0L,
  limit = 20L,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`platform`	A platform numerical identifier or a platform short name
`probe`	A probe name or it's numerical identifier
`offset`	The offset of the first retrieved result.
`limit`	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.

raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the querried gene(s) A list if raw = TRUE.

The fields of the output data.table are:

- gene.Symbol: Symbol for the gene
- gene.Ensembl: Ensembl ID for the gene
- gene.NCBI: NCBI id for the gene
- gene.Name: Name of the gene
- gene.MFX.Rank: Multifunctionality rank for the gene
- taxon.Name: Name of the species
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal identifier given to the species by Gemma
- taxon.NCBI: NCBI ID of the taxon
- taxon.Database.Name: Underlying database used in Gemma for the taxon
- taxon.Database.ID: ID of the underyling database used in Gemma for the taxon

Examples

```
get_platform_element_genes("GPL1355", "AFFX_Rat_beta-actin_M_at")
```

---

get_taxa *Get taxa*

Description

Returns taxa and their versions used in Gemma

Usage

```
get_taxa(memoised = getOption("gemma.memoised", FALSE))
```

Arguments

memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.

Value

A data frame including the names, IDs and database information about the taxons

Examples

```
get_taxa()
```

---

get_taxon_datasets *Retrieve the datasets for a given taxon*

Description

Retrieve the datasets for a given taxon

Usage

```
get_taxon_datasets(
  taxon,
  offset = 0L,
  limit = 20,
  sort = "+id",
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

taxon	can either be Taxon ID, Taxon NCBI ID, or one of its string identifiers: scientific name, common name. It is recommended to use Taxon ID for efficiency. Please note, that not all taxa have all the possible identifiers available. Use the `get_taxa_by_ids` function to retrieve the necessary information. For convenience, below is a list of officially supported taxa:

ID	Comm.name	Scient.name	NcbiID
1	human	Homo sapiens	9606
2	mouse	Mus musculus	10090
3	rat	Rattus norvegicus	10116

11	yeast	Saccharomyces cerevisiae	4932
12	zebrafish	Danio rerio	7955
13	fly	Drosophila melanogaster	7227
14	worm	Caenorhabditis elegans	6239

`offset` The offset of the first retrieved result.

`limit` Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.

`sort` Order results by the given property and direction. The '+' sign indicate ascending order whereas the '-' indicate descending.

`raw` `TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.

`memoised` Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised = TRUE)` will ensure that the cache is always used. Use `forget_gemma_memoised` to clear the cache.

`file` The name of a file to save the results to, or `NULL` to not write results to a file. If `raw == TRUE`, the output will be a JSON file. Otherwise, it will be a RDS file.

`overwrite` Whether or not to overwrite if a file exists at the specified filename.

`attributes` If `TRUE` additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the queried dataset(s). A list if `raw = TRUE`. Returns an empty list if no datasets matched. A successful response may contain 'Geeq' information, which aims to provide a unified metric to measure experiments by the quality of their data, and their suitability for use in Gemma. You can read more about the geeq properties here.

The fields of the output data.table are:

- `experiment.ShortName`: Shortname given to the dataset within Gemma. Often corresponds to accession ID
- `experiment.Name`: Full title of the dataset
- `experiment.ID`: Internal ID of the dataset.
- `experiment.Description`: Description of the dataset
- `experiment.Troubled`: Did an automatic process within gemma or a curator mark the dataset as "troubled"
- `experiment.Accession`: Accession ID of the dataset in the external database it was taken from
- `experiment.Database`: The name of the database where the dataset was taken from
- `experiment.URI`: URI of the original database
- `experiment.SampleCount`: Number of samples in the dataset

- `experiment.batchEffect`: A text field describing whether the dataset has batch effects
- `geeq.batchCorrected`: Whether batch correction has been performed on the dataset.
- `geeq.batchConfound`: 0 if batch info isn't available, -1 if batch counfoud is detected, 1 if batch information is available and no batch confound found
- `geeq.batchEffect`: -1 if batch p value < 0.0001, 1 if batch p value > 0.1, 0 if otherwise and when there is no batch information is available or when the data is confounded with batches.
- `geeq.rawData`: -1 if no raw data available, 1 if raw data was available. When available, Gemma reprocesses raw data to get expression values and batches
- `geeq.qScore`: Data quality score given to the dataset by Gemma.
- `geeq.sScore`: Suitability score given to the dataset by Gemma. Refers to factors like batches, platforms and other aspects of experimental design
- `taxon.Name`: Name of the species
- `taxon.Scientific`: Scientific name for the taxon
- `taxon.ID`: Internal identifier given to the species by Gemma
- `taxon.NCBI`: NCBI ID of the taxon
- `taxon.Database.Name`: Underlying database used in Gemma for the taxon
- `taxon.Database.ID`: ID of the underyling database used in Gemma for the taxon

Examples

```
get_taxon_datasets("human")
```

`search_annotations` *Search for annotation tags*

Description

Search for annotation tags

Usage

```
search_annotations(
  query,
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

query	The search query
raw	TRUE to receive results as-is from Gemma, or FALSE to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.
memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with annotations (annotation search result value objects) matching the given identifiers. A list if raw = TRUE. A 400 error if required parameters are missing.

The fields of the output data.table are:

- category.Name: Category that the annotation belongs to
- category.URI: URI for the category.Name
- value.Name: Annotation term
- value.URI: URI for the value.Name

Examples

```
search_annotations("traumatic")
```

---

search_datasets *Retrieve datasets associated to an annotation tags search*

Description

Retrieve datasets associated to an annotation tags search

Usage

```
search_datasets(
  query,
  taxon = NA_character_,
  offset = 0L,
  limit = 20L,
  sort = "+id",
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`query` The search query. Either plain text ('traumatic'), or an ontology term URI ('http://purl.obolibrary.org/obo/UBERON_0002048'). Datasets that contain the given string in their short or full name will also be matched. Can be multiple identifiers separated by commas.

`taxon` Can either be Taxon ID, Taxon NCBI ID, or one of its string identifiers: scientific name, common name. It is recommended to use Taxon ID for efficiency. Please note, that not all taxa have all the possible identifiers available. Use the `get_taxa_by_ids` function to retrieve the necessary information. For convenience, below is a list of officially supported taxa:

ID	Comm.name	Scient.name	NcbiID
1	human	Homo sapiens	9606
2	mouse	Mus musculus	10090
3	rat	Rattus norvegicus	10116
11	yeast	Saccharomyces cerevisiae	4932
12	zebrafish	Danio rerio	7955
13	fly	Drosophila melanogaster	7227
14	worm	Caenorhabditis elegans	6239

`offset` The offset of the first retrieved result.

`limit` Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.

`sort` Order results by the given property and direction. The '+' sign indicate ascending order whereas the '-' indicate descending.

`raw` `TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.

`memoised` Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing `options(gemma.memoised`

	= TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

A data table with information about the queried dataset(s). A list if raw = TRUE. Returns an empty list if no datasets matched. A successful response may contain 'Geeq' information, which aims to provide a unified metric to measure experiments by the quality of their data, and their suitability for use in Gemma. You can read more about the geeq properties here.

The fields of the output data.table are:

- experiment.ShortName: Shortname given to the dataset within Gemma. Often corresponds to accession ID
- experiment.Name: Full title of the dataset
- experiment.ID: Internal ID of the dataset.
- experiment.Description: Description of the dataset
- experiment.Troubled: Did an automatic process within gemma or a curator mark the dataset as "troubled"
- experiment.Accession: Accession ID of the dataset in the external database it was taken from
- experiment.Database: The name of the database where the dataset was taken from
- experiment.URI: URI of the original database
- experiment.SampleCount: Number of samples in the dataset
- experiment.batchEffect: A text field describing whether the dataset has batch effects
- geeq.batchCorrected: Whether batch correction has been performed on the dataset.
- geeq.batchConfound: 0 if batch info isn't available, -1 if batch counfoud is detected, 1 if batch information is available and no batch confound found
- geeq.batchEffect: -1 if batch p value < 0.0001, 1 if batch p value > 0.1, 0 if otherwise and when there is no batch information is available or when the data is confounded with batches.
- geeq.rawData: -1 if no raw data available, 1 if raw data was available. When available, Gemma reprocesses raw data to get expression values and batches
- geeq.qScore: Data quality score given to the dataset by Gemma.
- geeq.sScore: Suitability score given to the dataset by Gemma. Refers to factors like batches, platforms and other aspects of experimental design
- taxon.Name: Name of the species
- taxon.Scientific: Scientific name for the taxon
- taxon.ID: Internal identifier given to the species by Gemma
- taxon.NCBI: NCBI ID of the taxon
- taxon.Database.Name: Underlying database used in Gemma for the taxon
- taxon.Database.ID: ID of the underyling database used in Gemma for the taxon

Examples

```
search_datasets("bipolar", taxon = "human")
```

`search_gemma` *Search everything in Gemma.*

Description

Search everything in Gemma.

Usage

```
search_gemma(
  query,
  taxon = NA_character_,
  platform = NA_character_,
  limit = 20,
  resultType = "experiment",
  raw = getOption("gemma.raw", FALSE),
  memoised = getOption("gemma.memoised", FALSE),
  file = getOption("gemma.file", NA_character_),
  overwrite = getOption("gemma.overwrite", FALSE),
  attributes = getOption("gemma.attributes", TRUE)
)
```

Arguments

`query`	The search query. Either plain text ('traumatic'), or an ontology term URI ('http://purl.obolibrary.org/obo/UBERON_0002048'). Datasets that contain the given string in their short of full name will also be matched ('GSE201', 'Bronchoalveolar lavage samples'.
`taxon`	A numerical taxon identifier or an ncbi taxon identifier or a taxon identifier that matches either its scientific or common name
`platform`	A platform numerical identifier or a platform short name
`limit`	Optional, defaults to 20. Limits the result to specified amount of objects. Has a maximum value of 100. Use together with `offset` and the `totalElements` attribute in the output to compile all data if needed.
`resultType`	The kind of results that should be included in the output. Can be experiment, gene, platform or a long object type name, documented in the API documentation.
`raw`	`TRUE` to receive results as-is from Gemma, or `FALSE` to enable parsing. Raw results usually contain additional fields and flags that are omitted in the parsed results.

memoised	Whether or not to save to cache for future calls with the same inputs and use the result saved in cache if a result is already saved. Doing options(gemma.memoised = TRUE) will ensure that the cache is always used. Use forget_gemma_memoised to clear the cache.
file	The name of a file to save the results to, or NULL to not write results to a file. If raw == TRUE, the output will be a JSON file. Otherwise, it will be a RDS file.
overwrite	Whether or not to overwrite if a file exists at the specified filename.
attributes	If TRUE additional information from the call will be added into the output object's attributes such as offset and available elements.

Value

If raw = FALSE and resultType is experiment, gene or platform, a data.table containing the search results. If it is any other type, a list of results. A list with additional details about the search if raw = TRUE

Examples

```
search_gemma("bipolar")
```

---

set_gemma_user *Authentication by user name*

Description

Allows the user to access information that requires logging in to Gemma. To log out, run set_gemma_user without specifying the username or password.

Usage

```
set_gemma_user(username = NULL, password = NULL)
```

Arguments

username	Your username (or empty, if logging out)
password	Your password (or empty, if logging out)

Value

TRUE if authentication is successful, FALSE if not

Index

∗ **dataset**
get_dataset_annotations, 7
get_dataset_design, 8
get_dataset_differential_expression_analyses, 9
get_dataset_expression, 10
get_dataset_object, 11
get_dataset_platforms, 12
get_dataset_samples, 14
get_datasets_by_ids, 4
get_differential_expression_values, 15
search_datasets, 31
∗ **gene**
get_gene_go_terms, 18
get_gene_locations, 19
get_gene_probes, 20
get_genes, 16
∗ **misc**
forget_gemma_memoised, 3
gemma_call, 4
get_taxa, 27
search_annotations, 30
search_gemma, 34
set_gemma_user, 35
∗ **platform**
get_platform_annotations, 23
get_platform_datasets, 24
get_platform_element_genes, 26
get_platforms_by_ids, 21
∗ **taxon**
get_taxon_datasets, 28

attribute, *5*, *20*, *22*, *24*, *26*, *29*, *32*, *34*

forget_gemma_memoised, 3, *5*, *7–9*, *11–15*, *17–19*, *21–23*, *25*, *27–29*, *31*, *33*, *35*

gemma.R, 3
gemma_call, 4
get_dataset_annotations, 7
get_dataset_design, 8
get_dataset_differential_expression_analyses, 9, *15*
get_dataset_differential_expression_analyses(), *15*
get_dataset_expression, 10
get_dataset_object, *3*, 11
get_dataset_platforms, 12
get_dataset_samples, 14
get_datasets_by_ids, 4
get_differential_expression_values, *3*, *9*, 15
get_gene_go_terms, 18
get_gene_locations, 19
get_gene_probes, 20
get_genes, 16
get_platform_annotations, *3*, 23
get_platform_datasets, 24
get_platform_element_genes, 26
get_platforms_by_ids, 21
get_taxa, 27
get_taxa_by_ids, *28*, *32*
get_taxon_datasets, 28

search_annotations, 30
search_datasets, 31
search_gemma, 34
set_gemma_user, 35