Package 'rGREAT'

April 12, 2022

Type Package

Title Client for GREAT Analysis

Version 1.26.0

Date 2021-09-30

Author Zuguang Gu

Depends R (>= 3.1.2), GenomicRanges, IRanges, methods

Imports rjson, GetoptLong (>= 0.0.9), RCurl, utils, stats

Suggests testthat (>= 0.3), knitr, circlize (>= 0.4.8), rmarkdown

VignetteBuilder knitr

biocViews GeneSetEnrichment, GO, Pathways, Software, Sequencing, WholeGenome, GenomeAnnotation, Coverage

Description This package makes GREAT (Genomic Regions Enrichment of Annotations Tool) analysis automatic by constructing a HTTP POST request according to user's input and automatically retrieving results from GREAT web server.

URL https://github.com/jokergoo/rGREAT,
http://great.stanford.edu/public/html/

License MIT + file LICENSE

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

git_branch RELEASE_3_14

git_last_commit 28aedeb

git_last_commit_date 2021-10-26

Date/Publication 2022-04-12

Maintainer Zuguang Gu <z.gu@dkfz.de>

ℝ topics documented:

availableCategories-GreatJob-method

Available ontology categories

Description

Available ontology categories

Usage

```
## S4 method for signature 'GreatJob'
availableCategories(job)
```

Arguments

job   a `GreatJob-class` instance

Details

The values of the supported categories sometime change. You should run the function to get the real-time values. The meaning of categories returned is quite self-explained by the name.

Value

The returned value is a vector of categories.

Author(s)

Zuguang gu <z.gu@dkfz.de>

Examples

```
# note the `job` was generated from GREAT 3.0.0
job = readRDS(system.file("extdata", "job.rds", package = "rGREAT"))
availableCategories(job)
```

---

availableOntologies-GreatJob-method
All available ontology names

Description

All available ontology names

Usage

```
## S4 method for signature 'GreatJob'
availableOntologies(job, category = NULL)
```

Arguments

job	a GreatJob-class instance
category	one or multiple categories. All available categories can be get by availableCategories

Details

The values of the supported ontologies sometime change. You should run the function to get the real-time values. The meaning of ontology returned is quite self-explained by the name.

Value

The returned values is a vector of ontologies.

Author(s)

Zuguang gu <z.gu@dkfz.de>

Examples

```
# note the `job` was generated from GREAT 3.0.0
job = readRDS(system.file("extdata", "job.rds", package = "rGREAT"))
availableOntologies(job)
availableOntologies(job, category = "Pathway Data")
```

---

getEnrichmentTables-GreatJob-method
Get enrichment tables from GREAT web server

Description

Get enrichment tables from GREAT web server

Usage

```
## S4 method for signature 'GreatJob'
getEnrichmentTables(job, ontology = NULL, category = "GO",
    request_interval = 10, max_tries = 100, download_by = c("json", "tsv"),
    verbose = TRUE)
```

Arguments

job	a GreatJob-class instance
ontology	ontology names. Valid values are in availableOntologies. ontology is prior to category argument.
category	Pre-defined ontology categories. One category can contain more than one ontologies. Valid values are in availableCategories
request_interval	
	time interval for two requests. Default is 300 seconds.
max_tries	maximum tries
download_by	Internally used.
verbose	Whether print messages.

Details

The table contains statistics for the each term in each ontology catalogue.

Please note there is no FDR column in original tables. Users should calculate by themselves by functions such as p.adjust

Value

The returned value is a list of data frames in which each one corresponds to result for a single ontology. The structure of the data frames are same as the tables available on GREAT website.

See

availableOntologies, availableCategories

Author(s)

Zuguang gu <z.gu@dkfz.de>

See Also

[availableOntologies](), [availableCategories]()

Examples

```
# note the `job` was generated from GREAT 3.0.0
job = readRDS(system.file("extdata", "job.rds", package = "rGREAT"))
tb = getEnrichmentTables(job)
names(tb)
head(tb[[1]])
job

tb = getEnrichmentTables(job, ontology = "GO Molecular Function")
tb = getEnrichmentTables(job, category = "GO")
```

GreatJob   *Constructor method for GreatJob class*

Description

Constructor method for GreatJob class

Usage

```
GreatJob(...)
```

Arguments

...   arguments.

Details

There is no public constructor method for the [GreatJob-class]().

Value

No value is returned.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
# There is no example
NULL
```

GreatJob-class   *Class to store and retrieve GREAT results*

Description

Class to store and retrieve GREAT results

Details

After submitting request to GREAT server, the generated results will be available on GREAT server for some time. The GreatJob-class is defined to store parameters that user has set and result tables what were retrieved from GREAT server.

Constructor

Users don't need to construct by hand, submitGreatJob is used to generate a GreatJob-class instance.

Workflow

After submitting request to GREAT server, users can perform following steps:

- call getEnrichmentTables to get enrichment tables for selected ontologies catalogues.
- call plotRegionGeneAssociationGraphs to get associations between regions and genes as well as making plots.

Author(s)

Zuguang gu <z.gu@dkfz.de>

Examples

```
# please refer to page of `submitGreatJob`
NULL
```

---

plotRegionGeneAssociationGraphs-GreatJob-method
                Plot region-gene association figures

Description

Plot region-gene association figures

Usage

```
## S4 method for signature 'GreatJob'
plotRegionGeneAssociationGraphs(job, type = 1:3, ontology = NULL,
    termID = NULL, request_interval = 10, max_tries = 100, verbose = TRUE,
    plot = TRUE)
```

Arguments

job	a `GreatJob-class` instance
type	type of plots, should be in 1,2,3. See details section for explanation
ontology	ontology name
termID	term id which corresponds to the selected ontology
request_interval	
	time interval for two requests. Default is 300 seconds.
max_tries	maximum tries
verbose	whether show message
plot	whether make plots

Details

Generated figures are:

- association between regions and genes
- distribution of distance to TSS
- distribution of absolute distance to TSS

If `ontology` and `termID` are set, only regions and genes corresponding to selected ontology term will be used. Valid value for `ontology` is in `availableOntologies` and valid value for `termID` is from 'id' column in the table which is returned by `getEnrichmentTables`.

Value

a `GRanges` object. Columns in metadata are:

gene genes that are associated with corresponding regions

distTSS distance from the regions to TSS of the associated gene

The returned values corresponds to whole input regions or only regions in specified ontology term, depending on user's setting.

If there is no gene associated with the region, corresponding gene and `distTSS` columns will be `NA`.

Author(s)

Zuguang gu <z.gu@dkfz.de>

Examples

```
# note the `job` was generated from GREAT 3.0.0
job = readRDS(system.file("extdata", "job.rds", package = "rGREAT"))

res = plotRegionGeneAssociationGraphs(job)
res

plotRegionGeneAssociationGraphs(job, type = 1)

res = plotRegionGeneAssociationGraphs(job, ontology = "GO Molecular Function",
    termID = "GO:0004984")
res
```

---

submitGreatJob   *Send requests to GREAT web server*

Description

Send requests to GREAT web server

Usage

```
submitGreatJob(gr, bg = NULL,
    species             = "hg19",
    includeCuratedRegDoms = TRUE,
    rule                = c("basalPlusExt", "twoClosest", "oneClosest"),
    adv_upstream        = 5.0,
    adv_downstream      = 1.0,
    adv_span            = 1000.0,
    adv_twoDistance     = 1000.0,
    adv_oneDistance     = 1000.0,
    request_interval = 60,
    max_tries = 10,
    version = DEFAULT_VERSION,
    base_url = "http://great.stanford.edu/public/cgi-bin",
    help = TRUE)
```

Arguments

gr   A GRanges object or a data frame which contains at least three columns (chr,
                start and end). Regions for test.

bg   A GRanges object or a data frame. Background regions if needed. Note gr
                should be exactly subset of bg for all columns in gr. Check [http://great.
                stanford.edu/help/display/GREAT/File+Formats#FileFormats-Whatshouldmybackgroundregion](http://great.stanford.edu/help/display/GREAT/File+Formats#FileFormats-Whatshouldmybackgroundregion)
                [3F](3F) for more explanation.

species   Species. "hg38", "hg19", "mm10", "mm9" are supported in GREAT version
                  4.x.x, "hg19", "mm10", "mm9", "danRer7" are supported in GREAT version
                  3.x.x and "hg19", "hg18", "mm9", "danRer7" are supported in GREAT version
                  2.x.x.

includeCuratedRegDoms
                  Whether to include curated regulatory domains.

rule   How to associate genomic regions to genes. See 'details' section.

adv_upstream   Unit: kb, only used when rule is basalPlusExt

adv_downstream   Unit: kb, only used when rule is basalPlusExt

adv_span   Unit: kb, only used when rule is basalPlusExt

adv_twoDistance
                  Unit: kb, only used when rule is twoClosest

adv_oneDistance
                  Unit: kb, only used when rule is oneClosest

request_interval
                  Time interval for two requests. Default is 300 seconds.

max_tries   Maximum times trying to connect to GREAT web server.

version   version of GREAT. The value should be "4.0.4", "3.0.0", "2.0.2". Shorten ver-
                  sion numbers can also be used, such as using "4" or "4.0" is same as "4.0.4".

base_url   the url of cgi-bin path, only used when explicitly specified.

help   Whether to print help messages.

Details

Note: [On Aug 19 2019 GREAT released version 4](http://great.stanford.edu/help/display/
[GREAT/Version+History](http://great.stanford.edu/help/display/GREAT/Version+History) where it supports hg38 genome and removes some ontologies such path-
ways. submitGreatJob still takes hg19 as default. hg38 can be specified by the species = "hg38"
argument. To use the older versions such as 3.0.0, specify as submitGreatJob(...,version =
"3.0.0").

Note it is not the standard GREAT API. This function directly send data to GREAT web server by
HTTP POST.

Following text is copied from GREAT web site (http://great.stanford.edu/public/html/)

Explanation of rule and settings with names started with 'adv_' (advanced settings):

basalPlusExt Mode 'Basal plus extension'. Gene regulatory domain definition: Each gene is as-
signed a basal regulatory domain of a minimum distance upstream and downstream of the TSS
(regardless of other nearby genes, controlled by adv_upstream and adv_downstream argu-
ment). The gene regulatory domain is extended in both directions to the nearest gene's basal
domain but no more than the maximum extension in one direction (controlled by adv_span).

twoClosest Mode 'Two nearest genes'. Gene regulatory domain definition: Each gene is assigned
a regulatory domain that extends in both directions to the nearest gene's TSS (controlled by
adv_twoDistance) but no more than the maximum extension in one direction.

oneClosest Mode 'Single nearest gene'. Gene regulatory domain definition: Each gene is assigned
a regulatory domain that extends in both directions to the midpoint between the gene's TSS
and the nearest gene's TSS (controlled by adv_oneDistance) but no more than the maximum
extension in one direction.

Value

A GreatJob-class class object which can be used to get results from GREAT server.

Note

takes `hg19` as default. `hg38` can be specified by the `species = "hg38"` argument. To use the older versions such as 3.0.0, specify as `submitGreatJob(...,version = "3.0.0")`.**

Note it is not the standard GREAT API. This function directly send data to GREAT web server by HTTP POST.

Following text is copied from GREAT web site (<http://great.stanford.edu/public/html/>)

Explanation of `rule` and settings with names started with 'adv_' (advanced settings):

basalPlusExt Mode 'Basal plus extension'. Gene regulatory domain definition: Each gene is assigned a basal regulatory domain of a minimum distance upstream and downstream of the TSS (regardless of other nearby genes, controlled by `adv_upstream` and `adv_downstream` argument). The gene regulatory domain is extended in both directions to the nearest gene's basal domain but no more than the maximum extension in one direction (controlled by `adv_span`).

twoClosest Mode 'Two nearest genes'. Gene regulatory domain definition: Each gene is assigned a regulatory domain that extends in both directions to the nearest gene's TSS (controlled by `adv_twoDistance`) but no more than the maximum extension in one direction.

oneClosest Mode 'Single nearest gene'. Gene regulatory domain definition: Each gene is assigned a regulatory domain that extends in both directions to the midpoint between the gene's TSS and the nearest gene's TSS (controlled by `adv_oneDistance`) but no more than the maximum extension in one direction.

Author(s)

Zuguang gu <z.gu@dkfz.de>

See Also

GreatJob-class

Examples

```
set.seed(123)
bed = circlize::generateRandomBed(nr = 1000, nc = 0)
job = submitGreatJob(bed, version = "3.0.0")
job

# more parameters can be set for the job
if(FALSE) { # suppress running it when building the package
    # current GREAT version is 4.0.1
    job = submitGreatJob(bed, species = "mm9")
    job = submitGreatJob(bed, bg, species = "mm9", bgChoise = "data")
    job = submitGreatJob(bed, adv_upstream = 10, adv_downstream = 2, adv_span = 2000)
    job = submitGreatJob(bed, rule = "twoClosest", adv_twoDistance = 2000)
    job = submitGreatJob(bed, rule = "oneClosest", adv_oneDistance = 2000)
}
```

Index