

Package ‘QBMS’

March 28, 2023

Type Package

Title Query the Breeding Management System(s)

Version 0.9.1

Date 2023-03-28

Author Khaled Al-Shamaa [aut, cre],
Mariano Omar Crimi [ctb],
Zakaria Kehel [ctb],
Johan Aparicio [ctb],
ICARDA [cph]

Maintainer Khaled Al-Shamaa <k.el-shamaa@cgiar.org>

Description Linking data management systems to analytics is an important step in breeding digitization. Breeders can use this R package to Query the Breeding Management System(s) like 'BMS' <<https://bmspro.io>>, 'BreedBase' <<https://breedbase.org>>, and 'GIGWA' <<https://southgreen.fr/content/gigwa>> (using 'BrAPI' <<https://brapi.org>> calls) and help them to retrieve phenotypic and genotypic data directly into their analyzing pipelines.

License GPL (>= 3)

URL <https://icarda-git.github.io/QBMS/>

BugReports <https://github.com/icarda-git/QBMS/issues>

Depends R (>= 3.1.0)

Imports httr, jsonlite, tcltk, utils, RNetCDF, stats

Suggests knitr, rmarkdown, async

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

RoxygenNote 7.2.3

Repository CRAN

Date/Publication 2023-03-28 12:40:05 UTC

R topics documented:

brapi_get_call	3
brapi_headers	3
build_pedigree_table	4
calc_biovars	4
debug_qbms	6
get_async_page	6
get_async_pages	7
get_germplasm_attributes	7
get_germplasm_data	8
get_germplasm_id	9
get_germplasm_list	10
get_login_details	11
get_parents	11
get_pedigree_table	12
get_program_studies	13
get_program_trials	14
get_qbms_connection	15
get_study_data	16
get_study_info	17
get_terraclimate	18
get_trial_data	20
get_trial_obs_ontology	21
gigwa_get_metadata	22
gigwa_get_samples	23
gigwa_get_variants	24
gigwa_list_dbs	25
gigwa_list_projects	25
gigwa_list_runs	26
gigwa_set_db	27
gigwa_set_project	28
gigwa_set_run	29
list_crops	30
list_locations	30
list_programs	31
list_studies	32
list_trials	33
login_bms	34
login_gigwa	35
rbindlistx	36
rbindx	36
set_crop	37
set_program	38
set_qbms_config	39
set_qbms_connection	40
set_study	41
set_trial	42

brapi_get_call

3

Index

44

brapi_get_call *Internal function used for core BrAPI GET calls*

Description

This function created for **internal use only** to call BrAPI in GET method and retrieve the rough response data and send back the results. This function take care of pagination, authentication, encoding, compress, decode JSON response, etc.

Usage

```
brapi_get_call(call_url, nested = TRUE)
```

Arguments

`call_url` BrAPI URL to call in GET method
`nested` If FALSE, then retrieved JSON data will be flatten (default is TRUE)

Value

result object returned by JSON API response

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

brapi_headers *Common HTTP headers to send*

Description

Build the list of common HTTP headers to send with each API call

Usage

```
brapi_headers()
```

Value

A list of common HTTP headers to send.

build_pedigree_table *Building Pedigree Table Recursively*

Description

Internal helping function to build the pedigree table recursively.

Usage

```
build_pedigree_table(  
  geno_list = NULL,  
  pedigree_list = NULL,  
  pedigree_df = NULL  
)
```

Arguments

geno_list List of genotypes/germplasms names.
pedigree_list List of associated pedigree strings.
pedigree_df Pedigree data.frame as per previous call/iteration.

Value

A data.frame that has three columns correspond to the identifiers for the individual, female parent and male parent, respectively.

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

calc_biovars *Calculate the Bioclimatic Variables*

Description

Bioclimatic variables are derived from the monthly temperature and rainfall values in order to generate more biologically meaningful variables. These are often used in species distribution modeling and related ecological modeling techniques. The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation) seasonality (e.g., annual range in temperature and precipitation) and extreme or limiting environmental factors (e.g., temperature of the coldest and warmest month, and precipitation of the wet and dry quarters). A quarter is a period of three months (1/4 of the year).

They are coded as follows:

- BIO1 = Annual Mean Temperature

- BIO2 = Mean Diurnal Range (Mean of monthly (max temp - min temp))
- BIO3 = Isothermality (BIO2/BIO7) (* 100)
- BIO4 = Temperature Seasonality (standard deviation *100)
- BIO5 = Max Temperature of Warmest Month
- BIO6 = Min Temperature of Coldest Month
- BIO7 = Temperature Annual Range (BIO5-BIO6)
- BIO8 = Mean Temperature of Wettest Quarter
- BIO9 = Mean Temperature of Driest Quarter
- BIO10 = Mean Temperature of Warmest Quarter
- BIO11 = Mean Temperature of Coldest Quarter
- BIO12 = Annual Precipitation
- BIO13 = Precipitation of Wettest Month
- BIO14 = Precipitation of Driest Month
- BIO15 = Precipitation Seasonality (Coefficient of Variation)
- BIO16 = Precipitation of Wettest Quarter
- BIO17 = Precipitation of Driest Quarter
- BIO18 = Precipitation of Warmest Quarter
- BIO19 = Precipitation of Coldest Quarter

This work is derivative from the [dismo R package](#)

Usage

```
calc_biovars(data)
```

Arguments

`data` Data.frame has 4 mandatory columns (year, ppt, tmin, and tmax), and 12 rows (months) for each year sorted from Jan to Dec.

Value

Data.frame has 19 columns for "bioclim" variables (bio1-bio19) and last column for year (you will get one row per year).

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Robert Hijmans, Museum of Vertebrate Zoology, UC Berkeley

References

Nix, 1986. A biogeographic analysis of Australian elapid snakes. In: R. Longmore (ed.). Atlas of elapid snakes of Australia. Australian Flora and Fauna Series 7. Australian Government Publishing Service, Canberra.

`debug_qbms`*Debug internal QBMS status object*

Description

Return the internal QBMS status object for debugging

Usage

```
debug_qbms()
```

Value

an environment object for the package config and status

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
obj <- debug_qbms()
obj$config
obj$state
```

`get_async_page`*Async version of HTTP GET request*

Description

A small helper function to create 'async' version of the original HTTP GET request.

Usage

```
get_async_page(full_url, nested)
```

Arguments

<code>full_url</code>	URL to retrieve
<code>nested</code>	Don't flatten nested data.frames

Value

Async version of HTTP GET request.

get_async_pages *Run all supplied pages*

Description

A small helper function to create a deferred value that is resolved when all listed pages are resolved.

Usage

```
get_async_pages(pages, nested)
```

Arguments

pages	List of URLs to retrieve
nested	Don't flatten nested data.frames

Value

Async deferred object.

get_germplasm_attributes
Get germplasm attributes for a given germplasm in a crop

Description

Get germplasm attributes for a given germplasm in a crop

Usage

```
get_germplasm_attributes(germplasm_name = "")
```

Arguments

germplasm_name the name of the germplasm

Value

a data frame of the germplasm attributes

Author(s)

Johan Steven Aparicio, <j.aparicio@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [get_germplasm_data](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
  
  # retrieve attributes data of a given germplasm in a crop  
  germplasm_attributes <- get_germplasm_attributes("BASF-CORN-2-1")  
}
```

get_germplasm_data *Get the observations data of a given germplasm name in a crop*

Description

This function will retrieve all the observations data available for a given germplasm in the current crop database regardless of the programs/trials nested structure.

Usage

```
get_germplasm_data(germplasm_name = "")
```

Arguments

germplasm_name the name of the germplasm

Value

a data frame of the germplasm observations data aggregate from all trials

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [get_germplasm_attributes](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")

# select a breeding program by name
set_program("MC Maize")

# retrieve observations data of a given germplasm aggregated from all trials
germplasm_observations <- get_germplasm_data("BASFCORN-2-1")
}
```

get_germplasm_id	<i>Get Germplasm ID</i>
------------------	-------------------------

Description

Get the germplasm id for the given germplasm name in the current crop

Usage

```
get_germplasm_id(germplasm_name = "")
```

Arguments

germplasm_name the name of the germplasm

Value

a string of the germplasm id

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_crop](#), [get_germplasm_data](#), [get_germplasm_attributes](#)

get_germplasm_list *Get the germplasm list of the current active study*

Description

This function will retrieve the germplasm list of the current active study as configured in the internal state object using 'set_study()' function.

Usage

```
get_germplasm_list()
```

Value

a data frame of the study germplasm list

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#), [set_study](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
  
  # select a specific study/trial by name  
  set_trial("2018 PVT")  
  
  # select a specific environment/location dataset  
  set_study("2018 PVT Environment Number 1")  
  
  # retrieve the germplasm list of the selected environment/location  
  germplasm <- get_germplasm_list()  
}
```

get_login_details *Login pop-up window*

Description

Build a GUI pop-up window using Tcl/Tk to insert BMS username and password

Usage

```
get_login_details()
```

Value

a vector of inserted username and password

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

get_parents *Get Direct Parents*

Description

Internal helping function to split the given pedigree string that provides the parentage through which a cultivar was obtained, and get the pedigrees of the direct parents.

Usage

```
get_parents(pedigree)
```

Arguments

pedigree String provide the parentage through which a cultivar was obtained.

Value

Vector of two items, the direct female and male parents.

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

get_pedigree_table *Get the Pedigree Table*

Description

Get the pedigree table starting from current germplasm list and associated pedigree string that provides the parentage through which a cultivar was obtained.

Usage

```
get_pedigree_table(  
  data,  
  geno_column = "germplasmName",  
  pedigree_column = "pedigree"  
)
```

Arguments

data germplasm dataset as a data.frame.
geno_column name of the column that identifies the genotype/germplasm names.
pedigree_column name of the column that identifies the pedigree strings.

Value

A data.frame that has three columns correspond to the identifiers for the individual, female parent and male parent, respectively. The row giving the pedigree of an individual appears before any row where that individual appears as a parent. Founders use NA in the parental columns.

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
}
```

```

# select a specific study/trial by name
set_trial("2018 PVT")

# select a specific environment/location dataset
set_study("2018 PVT Environment Number 1")

# retrieve the germplasm list of the selected environment/location
germplasm <- get_germplasm_list()

pedigree_table <- get_pedigree_table(germplasm, "germplasmName", "pedigree")

#####
# nadiv package way
# library(nadiv)

# get additive relationship matrix in sparse matrix format
# A <- nadiv::makeA(pedigree_table)

# get A inverse matrix using base R function
# AINV <- solve(as.matrix(A))

#####
# ASreml-R package way
# library(asreml)

# represent A inverse matrix in efficient way using i,j index and Ainverse value
# actual genotype names of any given index are in the attr(ainv, "rowNames")
# ainv <- asreml::ainverse(pedigree_table)

#####
# dummy data set for testing
test <- data.frame(genotype = c("X", "Y"),
                  pedigree = c("A//B/D/2/C", "B/C/3/A//B/C/2/D"))

pedigree_table <- get_pedigree_table(test, "genotype", "pedigree")
}

```

get_program_studies *Get the list of trials studies locations information of the current selected program*

Description

This function will retrieve all environments/locations information of the trials studies in the current active program as configured in the internal state object using 'set_program()' function.

Usage

```
get_program_studies()
```

Value

a data frame of locations information for each study in the program trials

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")

# select a breeding program by name
set_program("MC Maize")

# retrieve all environments/locations information in the selected program studies/trials
program_studies <- get_program_studies()
}
```

get_program_trials *Internal function used to retrieve the rough list of trials*

Description

This function created for **internal use only** to retrieve the rough list of trials from the pre-selected (i.e. currently active) crop and breeding program combination as already configured in the internal state object using 'set_crop()' and 'set_program()' functions respectively.

Usage

```
get_program_trials()
```

Value

a list of trials information

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [list_trials](#)

get_qbms_connection *Get the QBMS connection*

Description

Get the QBMS connection object from the current environment

Usage

```
get_qbms_connection()
```

Value

a list of the current connection config and status

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_connection](#)

Examples

```
if(interactive()) {
# configure QBMS to connect the phenotypics server
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench/controller/auth/login")

# login, set the crop, and program
login_bms()
set_crop("maize")
set_program("MC Maize")

# get germplasm data
df1 <- get_germplasm_data("BASFCORN-2-1")

# save current connection (phenotypic server)
con1 <- get_qbms_connection()

# configure QBMS to connect the genotypic server
set_qbms_config("https://gigwa.southgreen.fr/gigwa/", engine = "gigwa", no_auth = TRUE)
```

```
# set the db, project, and run
gigwa_set_db("3kG_10M")
gigwa_set_project("3003_ind")
gigwa_set_run("1")

# get associated metadata
df2 <- gigwa_get_metadata()

# save current connection (before switch)
con2 <- get_qbms_connection()

# load the saved phenotypic server connection
set_qbms_connection(con1)

# continue retrieving germplasm attributes from the phenotypic server
df3 <- get_germplasm_attributes("BASFCORN-2-1")
}
```

get_study_data

Get the observations data of the current active study

Description

This function will retrieve the observations data of the current active study as configured in the internal state object using ‘set_study()’ function.

Usage

```
get_study_data()
```

Value

a data frame of the study observations data

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#), [set_study](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
```



```
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")

# select a breeding program by name
set_program("MC Maize")

# select a specific study/trial by name
set_trial("2018 PVT")

# select a specific environment/location dataset
set_study("2018 PVT Environment Number 1")

# retrieve the data of the selected environment/location
data <- get_study_data()
}
```

get_study_info

Get the details/metadata of the current active study

Description

This function will retrieve the details/metadata of the current active study as configured in the internal state object using 'set_study()' function.

Usage

```
get_study_info()
```

Value

a data frame of the study details/metadata

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#), [set_study](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
```

```

login_bms()

set_crop("maize")

# select a breeding program by name
set_program("MC Maize")

# select a specific study/trial by name
set_trial("2018 PVT")

# select a specific environment/location dataset
set_study("2018 PVT Environment Number 1")

# retrieve the general information of the selected environment/location
info <- get_study_info()
}

```

```

get_terraclimate      Get TerraClimate data for a given coordinate(s)

```

Description

TerraClimate is a monthly climate dataset for global terrestrial surfaces from 1958-2021. This function enables you to extract **climate variables** from the **hosting server** provided by the **Idaho University** for a given coordinate(s) without a need to download the whole raster files in the netCDF format (~100MB per variable for each year) and provide them in a standard data frame format ready to use in your code. It also calculates the **bioclimatic variables** using the **calc_biovars** function derivative from the **dismo R package**.

TerraClimate vs. **WorldClim**

- 1958-2021 vs. 1970-2000
- 14 vs. 7 climate variables
- ~4 km vs. ~1 km spatial resolution
- need to calculate vs. pre-calculated 19 bioclimatic variables

Usage

```

get_terraclimate(
  lat,
  lon,
  from = "1958-01-01",
  to = "2020-12-31",
  clim_vars = NULL,
  month_mask = NULL
)

```

Arguments

lat	Vector of Latitude(s) in decimal degree format.
lon	Vector of Longitude(s) in decimal degree format.
from	Start date as a string in the 'YYYY-MM-DD' format.
to	End date as a string in the 'YYYY-MM-DD' format.
clim_vars	List of all climate variables to be imported. Valid list includes: <i>aet</i> , <i>def</i> , <i>pet</i> , <i>ppt</i> , <i>q</i> , <i>soil</i> , <i>srad</i> , <i>swe</i> , <i>tmax</i> , <i>tmin</i> , <i>vap</i> , <i>ws</i> , <i>vpd</i> , and <i>PDSI</i> . Default is NULL for all.
month_mask	A list of all months of interest (e.g., planting season: <code>c(10:12, 1:5)</code>). Default is NULL for all.

Value

A list of two data.frame(s) for each pair of coordinates:

- **climate:** includes the climate variables ([reference](#)).
- **biovars:** includes the calculated bioclimatic variables ([reference](#)).

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

References

Abatzoglou, J., Dobrowski, S., Parks, S. *et al.* TerraClimate, a high-resolution global dataset of monthly climate and climatic water balance from 1958-2015. *Sci Data* **5**, 170191 (2018). [doi:10.1038/sdata.2017.191](https://doi.org/10.1038/sdata.2017.191)

Examples

```
if(interactive()) {
# data <- get_terraclimate(36.016, 36.943,
#                          '1979-09-01', '2012-06-30',
#                          c('ppt', 'tmin', 'tmax'), c(10:12,1:5))
data <- get_terraclimate(36.016, 36.943, '1979-09-01', '2012-06-30')

View(data$climate[[1]])

View(data$biovars[[1]])
}
```

get_trial_data	<i>Get the observations data of the current active trial</i>
----------------	--

Description

This function will retrieve the observations data of the current active trial (i.e. including all studies within) as configured in the internal state object using 'set_trial()' function.

Usage

```
get_trial_data()
```

Value

a data frame of the trial observations data

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
  
  # select a specific study/trial by name  
  set_trial("2018 PVT")  
  
  # select a specific environment/location dataset  
  set_study("2018 PVT Environment Number 1")  
  
  # retrieve multi-environment trial data  
  MET <- get_trial_data()  
}
```

`get_trial_obs_ontology`*Get the traits ontology/metadata of the current active trial*

Description

This function will retrieve the traits ontology/metadata of the current active trial as configured in the internal state object using 'set_trial()' function.

Usage

```
get_trial_obs_ontology()
```

Value

a data frame of the traits ontology/metadata

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
  
  # select a specific study/trial by name  
  set_trial("2018 PVT")  
  
  # get observation variable ontology  
  ontology <- get_trial_obs_ontology()  
}
```

`gigwa_get_metadata` *Get the metadata of the current active GIGWA run*

Description

This function will retrieve the metadata of the current active run as configured in the internal state object using ‘`gigwa_set_run()`’ function.

Usage

```
gigwa_get_metadata()
```

Value

a data.frame of all metadata associated to the samples in the selected run

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_run](#)

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",  
                 time_out = 300, engine = "gigwa", no_auth = TRUE)  
  
  # select a database by name  
  gigwa_set_db("3kG_10M")  
  
  # select a project by name  
  gigwa_set_project("3003_ind")  
  
  # select a specific run by name  
  gigwa_set_run("1")  
  
  # get a list of all samples in the selected run  
  metadata <- gigwa_get_metadata()  
}
```

gigwa_get_samples *Get the samples list of the current active GIGWA run*

Description

This function will retrieve the samples list of the current active run as configured in the internal state object using 'gigwa_set_run()' function.

Usage

```
gigwa_get_samples()
```

Value

a vector of all samples in the selected run

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_run](#)

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",  
                 time_out = 300, engine = "gigwa", no_auth = TRUE)  
  
  # select a database by name  
  gigwa_set_db("Sorghum-JGI_v1")  
  
  # select a project by name  
  gigwa_set_project("Nelson_et_al_2011")  
  
  # select a specific run by name  
  gigwa_set_run("run1")  
  
  # get a list of all samples in the selected run  
  samples <- gigwa_get_samples()  
}
```

gigwa_get_variants *Get variants in the selected GIGWA run*

Description

Query the variants (e.g., SNPs markers) in the selected GIGWA run that match a given criteria.

Usage

```
gigwa_get_variants(max_missing = 1, min_maf = 0, samples = NULL)
```

Arguments

max_missing maximum missing ratio (by sample) between 0 and 1 (default is 1 for 100%).
 min_maf minimum Minor Allele Frequency (MAF) between 0 and 1 (default is 0 for 0%).
 samples a list of a samples subset (default is NULL will retrieve for all samples).

Value

A data.frame that has the first 4 columns describe attributes of the SNP (rs#: variant name, alleles: reference allele / alternative allele, chrom: chromosome name, and pos: position in bp), while the following columns describe the SNP value for a single sample line using numerical coding 0, 1, and 2 for reference, heterozygous, and alternative/minor alleles.

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
if(interactive()) {
  # config your GIGWA connection
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",
                 time_out = 300, engine = "gigwa", no_auth = TRUE)

  # select a database by name
  gigwa_set_db("Sorghum-JGI_v1")

  # select a project by name
  gigwa_set_project("Nelson_et_al_2011")

  # select a specific run by name
  gigwa_set_run("run1")

  marker_matrix <- gigwa_get_variants(max_missing = 0.2,
                                     min_maf = 0.35,
                                     samples = c("ind1", "ind3", "ind7"))
}
```

`gigwa_list_dbs`*Get the list of existing databases in the current GIGWA server*

Description

Get the list of existing databases in the current GIGWA server

Usage

```
gigwa_list_dbs()
```

Value

a list of existing databases

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#)

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",  
                 time_out = 300, engine = "gigwa", no_auth = TRUE)  
  
  # list existing databases in the GIGWA server  
  gigwa_list_dbs()  
}
```

`gigwa_list_projects`*Get the list of all projects in the selected GIGWA database*

Description

This function will retrieve the projects list from the current active database as configured in the internal configuration object using 'gigwa_set_db()' function.

Usage

```
gigwa_list_projects()
```

Value

a list of projects names

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_db](#)

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",  
                 time_out = 300, engine = "gigwa", no_auth = TRUE)  
  
  # select a database by name  
  gigwa_set_db("Sorghum-JGI_v1")  
  
  # list existing projects  
  gigwa_list_projects()  
}
```

`gigwa_list_runs`

Get the list of run names in the selected GIGWA project

Description

This function will retrieve the runs list from the current active project as configured in the internal configuration object using ‘`gigwa_set_project()`’ function.

Usage

```
gigwa_list_runs()
```

Value

a list of runs names

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_project](#)

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("https://gigwa.southgreen.fr/gigwa/",  
                 time_out = 300, engine = "gigwa", no_auth = TRUE)  
  
  # select a database by name  
  gigwa_set_db("Sorghum-JGI_v1")  
  
  # select a project by name  
  gigwa_set_project("Nelson_et_al_2011")  
  
  # list all runs in the selected project  
  gigwa_list_runs()  
}
```

gigwa_set_db	<i>Set the current active GIGWA database by name</i>
--------------	--

Description

This function will update the current active database in the internal configuration object (including the brapi connection object).

Usage

```
gigwa_set_db(db_name)
```

Arguments

db_name the name of the database

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_list_dbs](#)

Examples

```
if(interactive()) {
# config your GIGWA connection
set_qbms_config("https://gigwa.southgreen.fr/gigwa/",
                time_out = 300, engine = "gigwa", no_auth = TRUE)

# select a database by name
gigwa_set_db("Sorghum-JGI_v1")
}
```

gigwa_set_project	<i>Set the current active GIGWA project</i>
-------------------	---

Description

This function will update the current active project in the internal state object using the programDbId retrieved from GIGWA which is associated to the given 'project_name' parameter.

Usage

```
gigwa_set_project(project_name)
```

Arguments

project_name the name of the project

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_db](#), [gigwa_list_projects](#)

Examples

```
if(interactive()) {
# config your GIGWA connection
set_qbms_config("https://gigwa.southgreen.fr/gigwa/",
                time_out = 300, engine = "gigwa", no_auth = TRUE)

# select a database by name
gigwa_set_db("Sorghum-JGI_v1")

# select a project by name
```

```
gigwa_set_project("Nelson_et_al_2011")
}
```

gigwa_set_run	<i>Set the current active GIGWA run</i>
---------------	---

Description

This function will update the current active run in the internal state object using the ‘studyDbIds’ retrieved from GIGWA which is associated to the given run_name parameter.

Usage

```
gigwa_set_run(run_name)
```

Arguments

run_name the name of the run

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[set_qbms_config](#), [gigwa_set_project](#), [gigwa_list_runs](#)

Examples

```
if(interactive()) {
# config your GIGWA connection
set_qbms_config("https://gigwa.southgreen.fr/gigwa/",
               time_out = 300, engine = "gigwa", no_auth = TRUE)

# select a database by name
gigwa_set_db("Sorghum-JGI_v1")

# select a project by name
gigwa_set_project("Nelson_et_al_2011")

# select a specific run by name
gigwa_set_run("run1")
}
```

list_crops	<i>Get the list of supported crops</i>
------------	--

Description

Get the list of supported crops

Usage

```
list_crops()
```

Value

a list of supported crops

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  # list supported crops in the bms server  
  list_crops()  
}
```

list_locations	<i>Get the list of locations information of the current selected crop</i>
----------------	---

Description

This function will retrieve the locations information of the current active crop as configured in the internal state object using 'set_crop()' function.

Usage

```
list_locations()
```

Value

a data frame of the locations information

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#)

list_programs	<i>Get the list of breeding programs names</i>
---------------	--

Description

This function will retrieve the breeding programs list from the current active crop as configured in the internal configuration object using 'set_crop()' function.

Usage

```
list_programs()
```

Value

a list of breeding programs names

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # list existing breeding programs  
  list_programs()  
}
```

`list_studies`*Get the list of studies in the current active trial*

Description

This function will retrieve the studies list from the current active trial as configured in the internal state object using 'set_trial()' function.

Usage

```
list_studies()
```

Value

a list of study and location names

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
  
  # select a specific study/trial by name  
  set_trial("2018 PVT")  
  
  # list all environments/locations information in the selected study/trial  
  list_studies()  
}
```

list_trials	<i>Get the list of trials in the current active breeding program</i>
-------------	--

Description

This function will retrieve the trials list from the current active breeding program as configured in the internal state object using 'set_program()' function.

Usage

```
list_trials(year = NULL)
```

Arguments

year the starting year to filter the list of trials (optional, default is NULL)

Value

a list of trials names

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")

# select a breeding program by name
set_program("MC Maize")

# list all studies/trials in the selected program
list_trials()

# filter listed studies/trials by year
list_trials(2020)
}
```

`login_bms`*Login to the server*

Description

Connect to the server. If username or password parameters are missing, then a login window will pop-up to insert username and password.

All other connection parameters (i.e. server IP or domain, connection port, API path, and connection protocol e.g. `http://`) will retrieve from the `qbms_config` list.

This function will update both of the `qbms_config` list (brapi connection object in the `con` key) and `qbms_state` list (token value in the `token` key).

Usage

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

Arguments

<code>username</code>	the username (optional, default is NULL)
<code>password</code>	the password (optional, default is NULL)

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
}
```

`login_gigwa`*Login to the GIGWA server*

Description

Connect to the GIGWA server. If username or password parameters are missing, then a login window will pop-up to insert username and password.

All other connection parameters (i.e. server IP or domain, connection port, API path, and connection protocol e.g. http://) will retrieve from the qbms_config list.

This function will update both of the qbms_config list (brapi connection object in the con key) and qbms_state list (token value in the token key).

Usage

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

Arguments

username	the GIGWA username (optional, default is NULL)
password	the GIGWA password (optional, default is NULL)

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
if(interactive()) {  
  # config your GIGWA connection  
  set_qbms_config("http://localhost:59395/gigwa/index.jsp", time_out = 300, engine = "gigwa")  
  
  # login using your GIGWA account (interactive mode)  
  # you can pass GIGWA username and password as parameters (batch mode)  
  login_gigwa()  
  login_gigwa("gigwadmin", "nimda")  
}
```

rbindlistx	<i>Makes one data.table from a list of many</i>
------------	---

Description

Same as `do.call("rbind", x)` on `data.frames`, but much faster.

Usage

```
rbindlistx(x)
```

Arguments

`x` A list containing `data.table`, `data.frame` or list objects.

Value

an unkeyed `data.table` containing a concatenation of all the items passed in.

rbindx	<i>Combine data.frames by row, filling in missing columns</i>
--------	---

Description

rbinds a list of data frames filling missing columns with NA

Usage

```
rbindx(..., dfs = list(...))
```

Arguments

`...` the first argument data frame.
`dfs` input data frames to row bind together.

Value

a single data frame

set_crop	<i>Set the current active crop</i>
----------	------------------------------------

Description

This function will update the current active crop in the internal configuration object (including the bapi connection object).

Usage

```
set_crop(crop_name)
```

Arguments

crop_name the name of the crop

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [list_crops](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
}
```

set_program	<i>Set the current active breeding program</i>
-------------	--

Description

This function will update the current active breeding program in the internal state object using the programDbId retrieved from BMS which is associated to the given program_name parameter.

Usage

```
set_program(program_name)
```

Arguments

program_name the name of the breeding program

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [list_programs](#)

Examples

```
if(interactive()) {  
  # config your BMS connection  
  set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")  
  
  # login using your BMS account (interactive mode)  
  # you can pass BMS username and password as parameters (batch mode)  
  login_bms()  
  
  set_crop("maize")  
  
  # select a breeding program by name  
  set_program("MC Maize")  
}
```

set_qbms_config	<i>Configure BMS server settings</i>
-----------------	--------------------------------------

Description

Set the connection configuration of the BMS server

Usage

```
set_qbms_config(  
    url = "http://localhost/ibpworkbench/controller/auth/login",  
    path = NULL,  
    page_size = 1000,  
    time_out = 120,  
    no_auth = FALSE,  
    engine = "bms",  
    verbose = TRUE  
)
```

Arguments

url	URL of the BMS login page (default is "http://localhost/ibpworkbench/")
path	BMS API path (default is NULL)
page_size	Page size (default is 1000)
time_out	Number of seconds to wait for a response until giving up (default is 10)
no_auth	TRUE if the server doesn't require authentication/login (default is FALSE)
engine	Backend database (qbms default, breedbase, gigwa)
verbose	Logical indicating if progress bar will display on the console when retrieve data from API (TRUE by default).

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

Examples

```
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")
```

set_qbms_connection *Set the QBMS connection*

Description

Set the QBMS connection object to the current environment

Usage

```
set_qbms_connection(env)
```

Arguments

env a list of the connection config and status to load

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[get_qbms_connection](#)

Examples

```
if(interactive()) {
# configure QBMS to connect the phenotypics server
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench/controller/auth/login")

# login, set the crop, and program
login_bms()
set_crop("maize")
set_program("MC Maize")

# get germplasm data
df1 <- get_germplasm_data("BASFCORN-2-1")

# save current connection (phenotypic server)
con1 <- get_qbms_connection()

# configure QBMS to connect the genotypic server
set_qbms_config("https://gigwa.southgreen.fr/gigwa/", engine = "gigwa", no_auth = TRUE)

# set the db, project, and run
gigwa_set_db("3kG_10M")
gigwa_set_project("3003_ind")
gigwa_set_run("1")

# get associated metadata
df2 <- gigwa_get_metadata()
```



```
# save current connection (before switch)
con2 <- get_qbms_connection()

# load the saved phenotypic server connection
set_qbms_connection(con1)

# continue retrieving germplasm attributes from the phenotypic server
df3 <- get_germplasm_attributes("BASFCORN-2-1")
}
```

set_study *Set the current active study by location name*

Description

This function will update the current active study in the internal state object using the studyDbId retrieved from BMS which is associated to the given study_name parameter.

Usage

```
set_study(study_name)
```

Arguments

study_name the name of the study

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [set_trial](#), [list_studies](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")
}
```

```
# select a breeding program by name
set_program("MC Maize")

# select a specific study/trial by name
set_trial("2018 PVT")

# select a specific environment/location dataset
set_study("2018 PVT Environment Number 1")
}
```

set_trial	<i>Set the current active trial</i>
-----------	-------------------------------------

Description

This function will update the current active trial in the internal state object using the trialDbId retrieved from BMS which is associated to the given trial_name parameter.

Usage

```
set_trial(trial_name)
```

Arguments

trial_name the name of the trial

Value

no return value

Author(s)

Khaled Al-Shamaa, <k.el-shamaa@cgiar.org>

See Also

[login_bms](#), [set_crop](#), [set_program](#), [list_trials](#)

Examples

```
if(interactive()) {
# config your BMS connection
set_qbms_config("https://www.bms-uat-test.net/ibpworkbench")

# login using your BMS account (interactive mode)
# you can pass BMS username and password as parameters (batch mode)
login_bms()

set_crop("maize")
}
```

```
# select a breeding program by name
set_program("MC Maize")

# select a specific study/trial by name
set_trial("2018 PVT")
}
```

Index

brapi_get_call, 3
brapi_headers, 3
build_pedigree_table, 4

calc_biovars, 4, 18

debug_qbms, 6

get_async_page, 6
get_async_pages, 7
get_germplasm_attributes, 7, 8, 9
get_germplasm_data, 7, 8, 9
get_germplasm_id, 9
get_germplasm_list, 10
get_login_details, 11
get_parents, 11
get_pedigree_table, 12
get_program_studies, 13
get_program_trials, 14
get_qbms_connection, 15, 40
get_study_data, 16
get_study_info, 17
get_terraclimate, 18
get_trial_data, 20
get_trial_obs_ontology, 21
gigwa_get_metadata, 22
gigwa_get_samples, 23
gigwa_get_variants, 24
gigwa_list_dbs, 25, 27
gigwa_list_projects, 25, 28
gigwa_list_runs, 26, 29
gigwa_set_db, 26, 27, 28
gigwa_set_project, 26, 28, 29
gigwa_set_run, 22, 23, 29

list_crops, 30, 37
list_locations, 30
list_programs, 31, 38
list_studies, 32, 41
list_trials, 15, 33, 42

login_bms, 7, 8, 10, 14–17, 20, 21, 30–33, 34, 37, 38, 41, 42
login_gigwa, 35

rbindlistx, 36
rbindx, 36

set_crop, 7–10, 14–17, 20, 21, 31–33, 37, 38, 41, 42
set_program, 10, 14–17, 20, 21, 32, 33, 38, 41, 42
set_qbms_config, 22, 23, 25–29, 39
set_qbms_connection, 15, 40
set_study, 10, 16, 17, 41
set_trial, 10, 16, 17, 20, 21, 32, 41, 42