Package 'rsnps'

March 3, 2015

Title Get SNP (Single-Nucleotide Polymorphism) Data on the Web

Description A programmatic interface to various SNP datasets on the web: openSNP, NBCI's dbSNP database, and Broad Institute SNP Annotation and Proxy Search. Functions are included for searching for SNPs for the Broad Institute and NCBI. For OpenSNP, functions are included for getting SNPs, and data for genotypes, phenotypes, annotations, and bulk downloads of data by user.

Version 0.1.6

Date 2015-03-03

License MIT + file LICENSE

URL https://github.com/ropensci/rsnps

BugReports https://github.com/ropensci/rsnps/issues

VignetteBuilder knitr

Imports plyr, stringr, httr, RCurl, XML, jsonlite

Suggests roxygen2, testthat, rjson, knitr

Author Scott Chamberlain [aut, cre], Kevin Ushey [aut]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-03 17:34:08

R topics documented:

`allgensnp` *Get genotype data for all users at a particular snp.*

Description

Get genotype data for all users at a particular snp.

Usage

```
allgensnp(snp = NA, df = FALSE, ...)
```

Arguments

`snp`	SNP name.
`df`	Return data.frame (TRUE) or not (FALSE) - default = FALSE.
`...`	Curl options passed on to `GET`.

Value

List of genotypes for all users at a certain SNP, or data.frame

Examples

```
## Not run:
allgensnp(snp='rs7412')
allgensnp('rs7412', df=TRUE)

## End(Not run)
```

`allphenotypes` *Get all phenotypes, their variations, and how many users have data available for a given phenotype.*

Description

Either return data.frame with all results, or output a list, then call the charicteristic by id (paramater = "id") or name (paramater = "characteristic").

Usage

```
allphenotypes(df = FALSE, ...)
```

Arguments

`df`	Return a data.frame of all data. The column known_variations can take multiple values, so the other columns id, characteristic, and number_of_users are replicated in the data.frame. (default = FALSE)
`...`	Curl options passed on to `GET`.

Value

Data.frame of results, or list if df=FALSE

Examples

```
## Not run:
# Get all data
allphenotypes(df = TRUE)

# Output a list, then call the characterisitc of interest by 'id' or 'characteristic'
datalist <- allphenotypes()
names(datalist) # get list of all characteristics you can call
datalist[["ADHD"]] # get data.frame for 'ADHD'
datalist[c("mouth size","SAT Writing")] # get data.frame for 'ADHD'

## End(Not run)
```

`annotations` *Get all phenotypes, their variations, and how many users have data available for a given phenotype.*

Description

Either return data.frame with all results, or output a list, then call the charicteristic by id (paramater = "id") or name (paramater = "characteristic").

Usage

```
annotations(snp = NA, output = c("all", "plos", "mendeley", "snpedia",
  "metadata"), ...)
```

Arguments

snp	SNP name.
output	Name the source or sources you want annotations from (options are: 'plos', 'mendeley', 'snpedia', 'metadata'). 'metadata' gives the metadata for the response.
...	Curl options passed on to GET.

Value

Data.frame of results.

Examples

```
## Not run:
# Get all data
annotations(snp = 'rs7903146', output = 'metadata') # get just the metadata
annotations(snp = 'rs7903146', output = 'plos') # just from plos
annotations(snp = 'rs7903146', output = 'snpedia') # just from snpedia
annotations(snp = 'rs7903146', output = 'all') # get all annotations

## End(Not run)
```

---

download_users *Download openSNP user files.*

Description

Download openSNP user files.

Usage

```
download_users(name = NULL, id = NULL, dir = "~/", ...)
```

Arguments

name	User name
id	User id
dir	Directory to save file to
...	Curl options passed on to GET.

Value

File downloaded to directory you specify (or default), nothing returned in R.

Examples

```
## Not run:
# Download a single user file, by id
download_users(id = 14)

# Download a single user file, by user name
download_users(name = 'kevinmcc')

# Download many user files
lapply(c(14,22), function(x) download_users(id=x))
read_users(id=14, nrows=5)

## End(Not run)
```

`fetch_genotypes` *Download genotype data for a user from 23andme or other repo.*

Description

Download genotype data for a user from 23andme or other repo.

Usage

```
fetch_genotypes(url, rows = 10, filepath = NULL, ...)
```

Arguments

`url`	URL for the download. See example below of function use.
`rows`	Number of rows to read in. Useful for getting a glimpse of the data. Default is 10 rows.
`filepath`	If none is given the file is saved to a temporary file, which will be lost after your session is closed. Save to a file if you want to access it later.
`...`	Further args passed on to `download.file`

Details

Beware, not setting the rows parameter means that you download the entire file, which can be large (e.g., 15MB), and so take a while to download depending on your connection speed. Therefore, rows is set to 10 by default to sort of protect the user.

Value

Dataset for a single user.

Examples

```
## Not run:
# get a data.frame of the users data
data <- users(df=TRUE)
head( data[[1]] ) # users with links to genome data
mydata <- fetch_genotypes(url = data[[1]][1,"genotypes.download_url"],
   file="~/myfile.txt", quiet=TRUE)

# see some data right away
mydata

# Or read in data later separately
read.table("~/myfile.txt", nrows=10)

## End(Not run)
```

`genotypes` *Get genotype data for one or multiple users.*

Description

Get genotype data for one or multiple users.

Usage

```
genotypes(snp = NA, userid = NA, df = FALSE, ...)
```

Arguments

`snp`	SNP name.
`userid`	ID of openSNP user.
`df`	Return data.frame (TRUE) or not (FALSE) - default = FALSE.
`...`	Curl options passed on to `GET`.

Value

List (or data.frame) of genotypes for specified user(s) at a certain SNP.

Examples

```
## Not run:
genotypes(snp='rs9939609', userid=1)
genotypes('rs9939609', userid='1,6,8', df=TRUE)
genotypes('rs9939609', userid='1-2', df=FALSE)

## End(Not run)
```

LDSearch *Search for SNPs in Linkage Disequilibrium with a set of SNPs*

Description

This function queries the SNP Annotation and Proxy tool (SNAP) for SNPs in high linkage disequilibrium with a set of SNPs, and also merges in up-to-date SNP annotation information available from NCBI.

Usage

```
LDSearch(SNPs, dataset = "onekgpilot", panel = "CEU", RSquaredLimit = 0.8,
  distanceLimit = 500, GeneCruiser = TRUE, quiet = FALSE)
```

Arguments

SNPs	A vector of SNPs (rs numbers).
dataset	The dataset to query. Must be one of: • rel21: HapMap Release 21 • rel22: HapMap Release 22 • hapmap3r2: HapMap 3 (release 2) • onekgpilot: 1000 Genomes Pilot 1
panel	The panel to use from the queried data set. Must be one of: • CEU • YRI • JPT+CHB If you are working with hapmap3r2, you can choose the additional panels: • ASW • CHD • GIH • LWK • MEK • MKK • TSI • CEU+TSI • JPT+CHB+CHD
RSquaredLimit	The R Squared limit to specify as a filter for returned SNPs; that is, only SNP pairs with R-squared greater than RSquaredLimit will be returned.
distanceLimit	The distance (in kilobases) upstream and downstream to search for SNPs in LD with each set of SNPs.
GeneCruiser	boolean; if TRUE we attempt to get gene info through GeneCruiser for each SNP. This can slow the query down substantially.
quiet	boolean; if TRUE progress updates are written to the console.

Details

For more details, please see http://www.broadinstitute.org/mpg/snap/ldsearch.php.

Information on the HapMap populations: http://ccr.coriell.org/Sections/Collections/NHGRI/hapmap.aspx?PgId=266&coll=HG

Information on the 1000 Genomes populations: http://www.1000genomes.org/category/frequently-asked-questions/population

Value

A list of data frames, one for each SNP queried, containing information about the SNPs found to be in LD with that SNP. A description of the columns follows:

- `Proxy`: The proxy SNP matched to the queried SNP.
- `SNP`: The SNP queried.
- `Distance`: The distance, in base pairs, between the queried SNP and the proxy SNP. This distance is calculated according to up-to-date position information returned from NCBI.
- `RSquared`: The measure of LD between the SNP and the proxy.
- `DPrime`: Another measure of LD between the SNP and the proxy.
- `GeneVariant`: Present if `GeneCruiser` is `TRUE`. This will identify where the SNP lies relative to its 'parent' SNP.
- `GeneName`: Present if `GeneCruiser` is `TRUE`. If the proxy SNP found lies within a gene, the name of that gene will be returned here. Otherwise, the field is `N/A`.
- `GeneDescription`: Present if `GeneCruiser` is `TRUE`. If the proxy SNP lies within a gene, information about that gene (as obtained from GeneCruiser) will be available here.
- `Major`: The major allele, as reported by SNAP.
- `Minor`: The minor allele, as reported by SNAP.
- `MAF`: The minor allele frequency corresponding to the reference panel queried, as obtained through SNAP.
- `NObserved`: The number of individuals from which the MAF information is generated, for column `MAF`.
- `Chromosome_NCBI`: The chromosome that the marker lies on.
- `Marker_NCBI`: The name of the marker. If the rs ID queried has been merged, the up-to-date name of the marker is returned here, and a warning is issued.
- `Class_NCBI`: The marker's 'class'. See http://www.ncbi.nlm.nih.gov/projects/SNP/snp_legend.cgi?legend=snpClass for more details.
- `Gene_NCBI`: If the marker lies within a gene (either within the exon or introns of a gene), the name of that gene is returned here; otherwise, `NA`. Note that the gene may not be returned if the marker lies too far upstream or downstream of the particular gene of interest.
- `Alleles_NCBI`: The alleles associated with the SNP if it is a SNV; otherwise, if it is an INDEL, microsatellite, or other kind of polymorphism the relevant information will be available here.
- `Major_NCBI`: The major allele of the SNP, on the forward strand, given it is an SNV; otherwise, `NA`.

- `Minor_NCBI:` The minor allele of the SNP, on the forward strand, given it is an SNV; otherwise, `NA`.
- `MAF_NCBI:` The minor allele frequency of the SNP, given it is an SNV. This is drawn from the current global reference population used by NCBI.
- `BP_NCBI:` The chromosomal position, in base pairs, of the marker, as aligned with the current genome used by dbSNP.

Examples

```
## Not run:
LDSearch("rs420358")
LDSearch('rs2836443')
LDSearch('rs113196607')

## End(Not run)
```

`NCBI_snp_query` *Query NCBI's dbSNP for information on a set of SNPs*

Description

This function queries NCBI's dbSNP for information related to the latest dbSNP build and latest reference genome for information on the vector of SNPs submitted.

Usage

```
NCBI_snp_query(SNPs, ...)
```

Arguments

`SNPs`	A vector of SNPs (rs numbers).
`...`	Further named parameters passed on to `config` to debug curl. See examples.

Details

Note that you are limited in the number of SNPs you pass in to one request because URLs can only be so long. Around 600 is likely the max you can pass in, though may be somewhat more. Break up your vector of SNP codes into pieces of 600 or less and do repeated requests to get all data.

Value

A dataframe with columns:

- `Query:` The rs ID that was queried.
- `Chromosome:` The chromosome that the marker lies on.
- `Marker:` The name of the marker. If the rs ID queried has been merged, the up-to-date name of the marker is returned here, and a warning is issued.

- `Class`: The marker's 'class'. See http://www.ncbi.nlm.nih.gov/projects/SNP/snp_legend.cgi?legend=snpClass for more details.
- `Gene`: If the marker lies within a gene (either within the exon or introns of a gene), the name of that gene is returned here; otherwise, `NA`. Note that the gene may not be returned if the marker lies too far upstream or downstream of the particular gene of interest.
- `Alleles`: The alleles associated with the SNP if it is a SNV; otherwise, if it is an INDEL, microsatellite, or other kind of polymorphism the relevant information will be available here.
- `Major`: The major allele of the SNP, on the forward strand, given it is an SNV; otherwise, `NA`.
- `Minor`: The minor allele of the SNP, on the forward strand, given it is an SNV; otherwise, `NA`.
- `MAF`: The minor allele frequency of the SNP, given it is an SNV. This is drawn from the current global reference population used by NCBI.
- `BP`: The chromosomal position, in base pairs, of the marker, as aligned with the current genome used by dbSNP.

References

http://www.ncbi.nlm.nih.gov/projects/SNP/

Examples

```
## Not run:
## an example with both merged SNPs, non-SNV SNPs, regular SNPs,
## SNPs not found, microsatellite
SNPs <- c("rs332", "rs420358", "rs1837253", "rs1209415715", "rs111068718")
NCBI_snp_query(SNPs)
NCBI_snp_query("123456") ##invalid: must prefix with 'rs'
NCBI_snp_query("rs420358")
NCBI_snp_query("rs332") # warning that its merged into another, try that
NCBI_snp_query("rs121909001")
NCBI_snp_query("rs1837253")
NCBI_snp_query("rs1209415715") # warning that no data available, returns 0 length data.frame
NCBI_snp_query("rs111068718") # warning that chromosomal information may be unmapped

NCBI_snp_query(SNPs='rs9970807')$BP

# Curl debugging
NCBI_snp_query("rs121909001")
library("httr")
NCBI_snp_query("rs121909001", config=verbose())
snps <- c("rs332", "rs420358", "rs1837253", "rs1209415715", "rs111068718")
NCBI_snp_query(snps, config=progress())

## End(Not run)
```

phenotypes *Get phenotype data for one or multiple users.*

Description

Get phenotype data for one or multiple users.

Usage

```
phenotypes(userid = NA, df = FALSE, ...)
```

Arguments

userid	ID of openSNP user.
df	Return data.frame (TRUE) or not (FALSE) - default = FALSE.
...	Curl options passed on to GET.

Value

List of phenotypes for specified user(s).

Examples

```
## Not run:
phenotypes(userid=1)
phenotypes(userid='1,6,8', df=TRUE)
phenotypes(userid='1-8', df=TRUE)

# coerce to data.frame
library(plyr)
df <- ldply(phenotypes(userid='1-8', df=TRUE))
head(df); tail(df)

# pass on curl options
library("httr")
phenotypes(1, config=c(verbose(), timeout(1)))
phenotypes(1, config=verbose())

## End(Not run)
```

phenotypes_byid *Get all known variations and all users sharing that phenotype for one phenotype(-ID).*

Description

Get all known variations and all users sharing that phenotype for one phenotype(-ID).

Usage

```
phenotypes_byid(phenotypeid = NA, return_ = c("description", "knownvars",
  "users"), ...)
```

Arguments

phenotypeid	ID of openSNP phenotype.
return_	Return data.frame (TRUE) or not (FALSE) - default = FALSE.
...	Curl options passed on to GET.

Value

List of description of phenotype, list of known variants, or data.frame of variants for each user with that phenotype.

Examples

```
## Not run:
phenotypes_byid(phenotypeid=12, return_ = 'desc')
phenotypes_byid(phenotypeid=12, return_ = 'knownvars')
phenotypes_byid(phenotypeid=12, return_ = 'users')

# pass on curl options
library("httr")
phenotypes_byid(phenotypeid=12, return_ = 'desc', config=c(verbose(), timeout(1)))
phenotypes_byid(phenotypeid=12, return_ = 'desc', config=verbose())

## End(Not run)
```

read_users *Read in openSNP user files from local storage.*

Description

Beware, these tables can be large. Check your RAM before executing. Or possibly read in a subset of the data. This function reads in the whole kitten kaboodle.

Usage

```
read_users(name = NULL, id = NULL, path = NULL, ...)
```

Arguments

name	User name
id	User id
path	Path to file to read from.
...	Parameters passed on to read.table.

Details

If you specify a name or id, this function reads environment variables written in the function download_users, and then searches against those variables for the path to the file saved. Alternatively, you can supply the path.

Value

A data.frame.

Examples

```
## Not run:
dat <- read_users(name = "kevinmcc")
head(dat)
dat <- read_users(id = 285)

## End(Not run)
```

`users` *Get openSNP users.*

Description

Get openSNP users.

Usage

```
users(df = FALSE, ...)
```

Arguments

`df`	Return data.frame (TRUE) or not (FALSE) - default = FALSE.
`...`	Curl options passed on to `GET`.

Value

List of openSNP users, their ID numbers, and XX if available.

Examples

```
## Not run:
# just the list
data <- users(df=FALSE)
data

# get a data.frame of the users data
data <- users(df=TRUE)
data[[1]] # users with links to genome data
data[[2]] # users without links to genome data

## End(Not run)
```

Index