

Package ‘annovarR’

January 9, 2018

Type Package

Title Integrated Framework to Annotate Genetic Variants

Version 1.0.0

Description The ‘annovarR’ package provides R functions as well as database resources which offer an integrated framework to annotate genetic variants from genome and transcriptome data. The wrapper functions of ‘annovarR’ unified the interface of many published annotation tools, such as ‘VEP’ (<<http://asia.ensembl.org/info/docs/tools/vep/index.html>>), ‘ANNOVAR’ (<<http://annovar.openbioinformatics.org/>>), ‘vcfanno’ (<<https://github.com/brentp/vcfanno>>) and ‘AnnotationDbi’ (<<http://www.bioconductor.org/packages/release/bioc/html/AnnotationDbi.html>>). It also simplified the use of some of the external annotation tools in R. Besides, massive published genetic variants annotation databases were integrated into ‘annovarR’. For example, ‘annovarR’ provides a newly RNA-seq allele frequency database, BRVar, which built from total 1,285 cases public B-progenitor acute lymphoblastic leukemia (B-ALL) transcriptome data.

Depends R (>= 3.3.0)

URL <https://github.com/JhuangLab/annovarR>

BugReports <https://github.com/JhuangLab/annovarR/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports stringr (>= 1.2.0), configr (>= 0.2.2), BioInstaller (>= 0.3.2), ngstk (>= 0.2.1), DBI (>= 0.6-1), data.table (>= 1.10.0), RSQLite (>= 2.0), futile.logger (>= 1.4.3), stringi (>= 1.1.5), RMySQL (>= 0.10.11), vcfR (>= 1.6.0), AnnotationDbi, glue, methods,

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, org.Hs.eg.db

VignetteBuilder knitr

NeedsCompilation no

Author Jianfeng Li [aut, cre]

Maintainer Jianfeng Li <lee_jianfeng@sjtu.edu.cn>

Repository CRAN

Date/Publication 2018-01-09 09:29:42 UTC

R topics documented:

annotation	3
annotation.cols.match	4
annotation.merge	6
annotation.region.match	6
annovar	8
annovarR	10
del	11
download.database	12
drop.mysql.index	13
drop.sqlite.index	14
get.annotation.dbtype	15
get.annotation.names	15
get.annotation.needcols	16
get.download.name	16
mysql.auto.build	17
mysql.auto.index	18
mysql.build	19
mysql.head	20
mysql.index	20
mysql.tb.colnames	21
mysql.tb.indexes	21
sql2sqlite	22
sqlite.auto.build	23
sqlite.auto.index	24
sqlite.build	25
sqlite.head	25
sqlite.index	26
sqlite.tb.colnames	27
sqlite.tb.indexes	27
sqlite2sql	28
vcfanno	29
vep	30

Index

32

annotation	<i>Annotation function (single name)</i>
------------	--

Description

Annotation function (single name)

Usage

```
annotation(dat = data.table(), anno.name = "", buildver = "hg19",
  annovar.anno.names = "", database.dir = Sys.getenv("annovarR_DB_DIR", ""),
  db.type = NULL, database.cfg = system.file("extdata",
  "config/databases.toml", package = "annovarR"), func = NULL,
  mysql.connect.params = list(host = "", dbname = "", table.name = "", user =
  "", password = ""), sqlite.connect.params = list(dbname = ""), ...)
```

Arguments

dat	A data.table including all of your data, eg. data.table(chr=c(1,2,3), start=c(1111,1112,1113))
anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all .etc.
buildver	Genome version, hg19, hg38, mm10 and others
annovar.anno.names	If anno.name equal perl_annovar_merge, you can use annovar.anno.names to annotate multiple database supported by ANNOVAR, the names can be found on the http://annovar.openbioinformatics.org/en/latest/user-guide/download/
database.dir	Dir of the databases
db.type	Setting the database type (sqlite or txt)
database.cfg	Configuration file of annovarR databases information
func	Function to anntate the dat data, default is to search the function in extdata/database.toml
mysql.connect.params	Connect MySQL database other parameters, e.g. list(host='11.11.11.1', port = '3306', user = "", password = '123456')
sqlite.connect.params	Connect SQLite database other paramertes, default is not need
...	Other parameters see annotation.cols.match , annotation.region.match , annovar and vep

Examples

```
library(data.table)
chr <- c('chr1', 'chr2', 'chr1')
start <- c('10020', '10020', '10020')
end <- c('10020', '10020', '10020')
ref <- c('A', 'A', 'A')
alt <- c('-', '-', '-')
```

```

database <- system.file('extdata', 'demo/hg19_avsnp147.txt', package = 'annovarR')
database.dir <- dirname(database)
dat <- data.table(chr = chr, start = start, end = end, ref = ref, alt = alt)
x <- annotation(dat, 'avsnp147', database.dir = database.dir,
return.col.names = 'avSNP147', db.type = 'txt')

```

annotation.cols.match *A position annotation utils that can be used to write a yourself annotation function*

Description

A position annotation utils that can be used to write a yourself annotation function

Usage

```

annotation.cols.match(dat = data.table(), anno.name = "",
  buildver = "hg19", database.dir = Sys.getenv("annovarR_DB_DIR", ""),
  db.col.order = 1:5, index.cols = c("chr", "start"),
  matched.cols = c("chr", "start", "end", "ref", "alt"),
  return.col.index = 6, return.col.names = "",
  return.col.names.profix = "", format.dat.fun = format.cols,
  dbname.fixed = NULL, table.name.fixed = NULL, setdb.fun = set.db,
  set.table.fun = set.table, format.db.tb.fun = format.db.tb,
  db.type = "sqlite", db.file.prefix = NULL,
  mysql.connect.params = list(), sqlite.connect.params = list(),
  fread.db.params = list(), verbose = FALSE)

```

Arguments

dat	A data.table including all of your data, eg. data.table(chr=c(1,2,3), start=c(1111,1112,1113))
anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases (mysql no need)
db.col.order	Using the index, you can rename the database table, and can be matched using matched.cols.
index.cols	Using the selected cols to match data with sqlite database. eg. c('chr', 'start'), 'rs'
matched.cols	Using the selected cols to match data with selected partial data by index.cols limited.
return.col.index	Setting the colnums need be returned
return.col.names	Setting the returned colnum names
return.col.names.profix	Setting the returned colnum names profix

format.dat.fun	A function to process input data. eg. as.numeric(dat\$start); as.character(dat\$chr)
dbname.fixed	Database path (txt, sqlite) or name (MySQL), default is NULL, and get from setdb.fun (Set value will fix the dbname, and will be added in sqlite.connnect.params and mysql.connect.params)
table.name.fixed	Table name, default is NULL, and get from set.table.fun (Set value will fix the table.name) (Set value will fix the table.name, and will be added in sqlite.connnect.params and mysql.connect.params)
setdb.fun	A function to process the name, buildver, database.dir and get the database path (MySQL return NULL)
set.table.fun	A function to process the name, buildver and get the final table name
format.db.tb.fun	A function to process the selected database table that can be used to matched with your data
db.type	Setting the database type (sqlite, txt or mysql)
db.file.prefix	Only be setted when db.type is local databae like sqlite or txt
mysql.connect.params	Connect MySQL database other parameters, e.g. list(host='11.11.11.1', port = '3306', user = '', password = '123456')
sqlite.connect.params	Connect SQLite database other paramertes, default is not need
fread.db.params	For text format database, you can use fread.db.params to control the fread behavior
verbose	Logical indicating wheather print the extra log infomation

Examples

```

library(data.table)
chr <- c('chr1', 'chr2', 'chr1')
start <- c('10020', '10020', '10020')
end <- c('10020', '10020', '10020')
ref <- c('A', 'A', 'A')
alt <- c('-', '-', '-')
database <- system.file('extdata', 'demo/hg19_avsnp147.txt', package = 'annovarR')
database.dir <- dirname(database)
dat <- data.table(chr = chr, start = start, end = end, ref = ref, alt = alt)
x <- annotation.cols.match(dat, 'avsnp147', database.dir = database.dir,
return.col.names = 'avSNP147', db.type = 'txt')

```

annotation.merge *Annotation function (multiple name)*

Description

Annotation function (multiple name)

Usage

```
annotation.merge(anno.names, ...)
```

Arguments

anno.names Annotation names, eg. c('avsnp138', 'avsnp147', '1000g2015aug_all')

... Other parameters see [annotation](#)

Examples

```
library(data.table)
chr <- c('chr1', 'chr2', 'chr1')
start <- c('10020', '10020', '10020')
end <- c('10020', '10020', '10020')
ref <- c('A', 'A', 'A')
alt <- c('-', '-', '-')
database <- system.file('extdata', 'demo/hg19_avsnp147.txt', package = 'annovarR')
database.dir <- dirname(database)
dat <- data.table(chr = chr, start = start, end = end, ref = ref, alt = alt)
x <- annotation.merge(dat = dat, anno.names = c('avsnp147'),
  database.dir = database.dir, db.type = 'txt')
```

annotation.region.match

A region annotation utils that can be used to write a yourself annotation function

Description

A region annotation utils that can be used to write a yourself annotation function

Usage

```

annotation.region.match(dat = data.table(), anno.name = "",
  buildver = "hg19", database.dir = Sys.getenv("annovarR_DB_DIR", ""),
  db.col.order = 1:3, index.cols = c("chr", "start", "end"),
  full.matched.cols = "chr", inferior.col = "start", superior.col = "end",
  return.col.index = 4, return.col.names = "",
  return.col.names.prefix = "", format.dat.fun = format.cols,
  dbname.fixed = NULL, table.name.fixed = NULL, setdb.fun = set.db,
  set.table.fun = set.table, format.db.tb.fun = format.db.region.tb,
  db.type = "sqlite", db.file.prefix = NULL,
  mysql.connect.params = list(), sqlite.connect.params = list(),
  fread.db.params = list(), verbose = FALSE)

```

Arguments

dat	A data.table including all of your data, eg. data.table(chr=c(1,2,3), start=c(1111,1112,1113))
anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases (mysql no need)
db.col.order	Using the index, you can rename the database table, and can be matched using matched.cols.
index.cols	Using the selected cols to match data with sqlite database. eg. c('chr', 'start'), 'rs'
full.matched.cols	Using the selected cols to match data with selected partial data by index.cols limited.
inferior.col	Inferior limit col, e.g. start
superior.col	Superior limit col, e.g. end
return.col.index	Setting the colnums need be returned
return.col.names	Setting the returned colnum names
return.col.names.prefix	Setting the returned colnum names prefix
format.dat.fun	A function to process input data. eg. as.numeric(dat\$start); as.character(dat\$chr)
dbname.fixed	Database path (txt, sqlite) or name (MySQL), default is NULL, and get from setdb.fun (Set value will fix the dbname, and will be added in sqlite.connenct.params and mysql.connect.params)
table.name.fixed	Table name, default is NULL, and get from set.table.fun (Set value will fix the table.name) (Set value will fix the table.name, and will be added in sqlite.connenct.params and mysql.connect.params)
setdb.fun	A function to process the name, buildver, database.dir and get the database path (MySQL return NULL)

`set.table.fun` A function to process the name, buildver and get the final table name
`format.db.tb.fun` A function to process the selected database table that can be used to matched with your data
`db.type` Setting the database type (sqlite, txt or mysql)
`db.file.prefix` Only be setted when db.type is local databae like sqlite or txt
`mysql.connect.params` Connect MySQL database other parameters, e.g. `list(host='11.11.11.1', port = '3306', user = "", password = '123456')`
`sqlite.connect.params` Connect SQLite database other paramertes, default is not need
`fread.db.params` For text format database, you can use `fread.db.params` to control the fread behavior
`verbose` Logical indicating wheather print the extra log infomation

Examples

```

library(data.table)
bed.file <- system.file('extdata', 'demo/example.bed', package = 'annovarR')
bed.sqlite <- sprintf('%s/%s.sqlite', tempdir(), basename(bed.file))
connect.params <- list(dbname = bed.sqlite, table.name = 'bed')
sqlite.build(bed.file, connect.params)
chr <- c('chr10', 'chr1')
start <- c('100188904', '100185955')
end <- c('100188904', '100185955')
dat <- data.table(chr = chr, start = start, end = end)
## Not run:
x <- annotation.region.match(dat = dat, database.dir = tempdir(),
dbname.fixed = bed.sqlite, table.name.fixed = 'bed',
db.type = 'sqlite', format.dat.fun = function(...) {
params = list(...);return(params[[1]])})

## End(Not run)
file.remove(bed.sqlite)
  
```

annovar

R function to run ANNOVAR.

Description

R function to run ANNOVAR.

Usage

```
annovar(perl = Sys.which("perl"), cmd.pool = list(script1.downdb =
  paste(c("{perl}", "{script}{extra.params}", "-downdb", "{buildver}",
    "{webfrom}", "{down.dbname}", "{database.dir}"), collapse = " "),
  script1.gene.based = paste(c("{perl}", "{script}{extra.params}", "{buildver}",
    "{input.file}", "{database.dir}"), collapse = " "), script1.region.based =
  paste(c("{perl}", "{script}", " -regionanno{extra.params}", "{buildver}",
    "{anno.names}", "{input.file}", "{database.dir}"), collapse = " "),
  script1.filter.based = paste(c("{perl}", "{script}",
    "-filter{extra.params}", "{buildver}", "{anno.names}", "{input.file}",
    "{database.dir}"), collapse = " "), script2 = paste(c("{perl}", "{script}",
    "{input.file}", "{database.dir}", "{buildver}", "{out}",
    "-remove{extra.params}", "-protocol {anno.names}", "-operation",
    "{operation}", "{nastring}", "{otherinfo}", "{vcfinput}"), collapse = " "),
  script3 = paste("{perl}", "{script}{extra.params}", "-format", "{format}",
    "{input.file}", "> {convert.out}", collapse = " ")),
cmd.used = "script1.downdb", down.dbname = "", input.file = "",
annovar.dir = "", buildver = "hg19",
database.dir = "{annovar.dir}/humandb", webfrom = "annovar",
anno.names = "", out = "", convert.out = "", format = "vcf4",
operation.type = list(gene.based = c("refGene", "knownGene", "ensGene",
  "ccdsGene"), region.based = c("cytoBand", "genomicSuperDups")),
cmd.profix.flag = list(buildver = "-buildver", anno.names = "-dbtype",
  webfrom = "-webfrom", out = "-out", nastring = "-nastring"),
otherinfo = FALSE, nastring = "NA", vcfinput = FALSE,
extra.params = "", debug = FALSE)
```

Arguments

perl	Executable file of perl
cmd.pool	Un-parsed commands of ANNOVAR
cmd.used	Name in cmd.pool that used to parse final run command
down.dbname	Need to download database name, e.g. avsnp147,cosmic70,1000g2015aug
input.file	Input file name, e.g. example.avinput, example.vcf
annovar.dir	ANNOVAR source code directory
buildver	Genome version e.g. hg19, mm10
database.dir	Database directory, e.g. /opt/annovar/humandb
webfrom	Database resource warehouse, e.g. ucsc, annovar
anno.names	ANNOVAR annotation names
out	ANNOVAR -out parameter value
convert.out	ANNOVAR convert2annovar.pl output file, e.g. out.avinput
format	ANNOVAR convert2annovar.pl input format option
operation.type	Operation types used in table_annovar.pl

cmd.prefix.flag	Prefix used in ANNOVAR command
otherinfo	Used in table_annovar.pl, -otherinfo
nastring	ANNOVAR -nastring value, default is NA.
vcfinput	Specify that input is in VCF format and output will be in VCF format, table_annovar.pl, -vcfinput
extra.params	Extra paramters in ANNOVAR command
debug	If set TRUE, only print the command

Examples

```
# original ANNOVAR download.database
down.dbname <- 'refGene'
annovar('perl', cmd.used = 'script1.downdb', down.dbname = 'avsnp147',
        annovar.dir = '/opt/annovar', debug = TRUE)

# ANNOVAR gene-based annotation
annovar('perl', cmd.used = 'script1.gene.based', input.file = 'example.avinput',
        annovar.dir = '/opt/annovar', debug = TRUE)

# ANNOVAR gene-based annotation
annovar('perl', cmd.used = 'script1.region.based', anno.names = 'cytoBand',
        input.file = 'example.avinput', annovar.dir = '/opt/annovar', debug = TRUE)

# ANNOVAR filter-based annotation
annovar('perl', cmd.used = 'script1.filter.based', anno.names = 'avsnp147',
        input.file = 'example.avinput', annovar.dir = '/opt/annovar', debug = TRUE)

# ANNOVAR table_annovar.pl
anno.names <- c('refGene', 'cytoBand', 'genomicSuperDups', 'esp6500siv2_all',
               '1000g2015aug_all', '1000g2015aug_afr', '1000g2015aug_eas', '1000g2015aug_eur',
               'snp138', 'avsnp142', 'avsnp144', 'avsnp147', 'ljb26_all', 'cosmic70', 'cosmic81')
annovar('perl', cmd.used = 'script2',
        anno.names = anno.names,
        input.file = 'example.avinput', annovar.dir = '/opt/annovar', debug = TRUE)

# ANNOVAR convert2annovar.pl
annovar('perl', cmd.used = 'script3', input.file = 'example.vcf', format = 'vcf4old',
        convert.out = 'example.avinput', annovar.dir = '/opt/annovar', debug = TRUE)
```

annovarR *The 'annovarR' package provides R functions as well as database resources which offer an integrated framework to annotate genetic variants from genome and transcriptome data. The wrapper functions of 'annovarR' unified the interface of many published annotation tools, such as 'VEP' (<http://asia.ensembl.org/info/docs/tools/vep/index.html>), 'ANNOVAR' (<http://annovar.openbioinformatics.org/>), 'vcfanno' (<https://github.com/brentp/vcfanno>) and 'AnnotationDbi' (<http://www.bioconductor.org/packages/release/bioc/html/AnnotationDbi.html>). It also simplified the use of some of the external annotation tools in R. Besides, massive published genetic variants annotation databases were integrated into 'annovarR'. For example, 'annovarR' provides a newly RNA-seq allele frequency database, BRVar, which built from total 1,285 cases public B-progenitor acute lymphoblastic leukemia (B-ALL) transcriptome data.*

Description

The 'annovarR' package provides R functions as well as database resources which offer an integrated framework to annotate genetic variants from genome and transcriptome data. The wrapper functions of 'annovarR' unified the interface of many published annotation tools, such as 'VEP' (<http://asia.ensembl.org/info/docs/tools/vep/index.html>), 'ANNOVAR' (<http://annovar.openbioinformatics.org/>), 'vcfanno' (<https://github.com/brentp/vcfanno>) and 'AnnotationDbi' (<http://www.bioconductor.org/packages/release/bioc/html/AnnotationDbi.html>). It also simplified the use of some of the external annotation tools in R. Besides, massive published genetic variants annotation databases were integrated into 'annovarR'. For example, 'annovarR' provides a newly RNA-seq allele frequency database, BRVar, which built from total 1,285 cases public B-progenitor acute lymphoblastic leukemia (B-ALL) transcriptome data.

Author(s)

Li Jianfeng lee_jianfeng@sjtu.edu.cn

See Also

Useful links:

<https://github.com/JhuangLab/annovarR>

Report bugs at <https://github.com/JhuangLab/annovarR/issues>

del

Delete table or database (text file, sqlite, mysql)

Description

Delete table or database (text file, sqlite, mysql)

Usage

```
del(filename = "", sqlite.connect.params = list(),
     mysql.connect.params = list(), del.type = "database",
     db.type = "sqlite", verbose = FALSE, ...)
```

Arguments

filename	Path of text format data
sqlite.connect.params	Connect to sqlite database params [dbname, table.name]
mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
del.type	file (for filename), database (for sqlite and MySQL), table (for sqlite and MySQL) be supported (delete file, delete sqlite db file, delete table in sqlite db, delete table in MySQL db and delete database in MySQL database)
db.type	txt, sqlite or mysql
verbose	Logical indicating wheather show the log message
...	Other parameters be used in dbSendQuery

Examples

```
db <- tempfile()
file.create(db)
del(db, del.type = 'file')
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,
table.name = 'snp_test'))
del(sqlite.connect.params = list(dbname = test.sqlite,
table.name = 'snp_test'), del.type = 'table')
del(sqlite.connect.params = list(dbname = test.sqlite,
table.name = 'snp_test'))
```

download.database	<i>Download annovarR databases</i>
-------------------	------------------------------------

Description

Download annovarR databases

Usage

```
download.database(download.name = NULL, database.dir = tempdir(),
                 version = c(), buildver = "hg19", download.cfg = system.file("extdata",
"config/db/db_annovar.toml", package = "BioInstaller"),
                 show.all.versions = FALSE, show.all.names = FALSE,
                 show.all.buildvers = FALSE, verbose = FALSE, ...)
```

Arguments

download.name	Download name, default is NULL and get value from database.cfg dependence_db
database.dir	Destination directory to store databases
version	Version of download database, supported version can be get by download.database('db_annovar_avsnp', show.all.versions = TRUE)
buildver	Genome version, e.g hg19, hg38, mm10
download.cfg	Configuration file for download, most of annovarR database resources can be found in system.file('extdata', 'config/db/db_annovar.toml', package = 'BioInstaller')
show.all.versions	Logical indicating wheather show all avaiable versions
show.all.names	Logical indicating wheather show all avaiable name
show.all.buildvers	Logical indicating wheather show all avaiable buildver
verbose	Logical indicating wheather print the extra log infomation
...	Other parameters pass to install.bioinfo

Examples

```
download.database('db_annovar_avsnp', database.dir = sprintf('%s/databases/', tempdir()),
show.all.versions = TRUE)
```

drop.mysql.index	<i>Drop search index of annovarR database table in mysql</i>
------------------	--

Description

Drop search index of annovarR database table in mysql

Usage

```
drop.mysql.index(mysql.connect.params = list(host = "", dbname = "",
table.name = "", user = "user", password = "password"), index = "",
verbose = FALSE, ...)
```

Arguments

mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
index	Index name in mysql
verbose	Ligical indicating wheather show the log message
...	Other parameters be used in dbSendQuery

Examples

```
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
## Not run:
drop.mysql.index(list(host = 'host', dbname = 'db', user = 'user', password = 'password'),
index = 'index_name')

## End(Not run)
```

drop.sqlite.index *Drop search index of annovarR database table in sqlite*

Description

Drop search index of annovarR database table in sqlite

Usage

```
drop.sqlite.index(sqlite.connect.params = list(dbname = "", table.name = ""),
index = "", verbose = FALSE, ...)
```

Arguments

sqlite.connect.params	Connect to sqlite database params [dbname]
index	Index name in sqlite
verbose	Logical indicating wheather show the log message
...	Other parameters be used in dbSendQuery

Examples

```
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,
table.name = 'snp_test'))
x <- sqlite.index(list(dbname = test.sqlite, table.name = 'snp_test'),
index = 'index4', cols = c('V1', 'V2'))
x <- drop.sqlite.index(list(dbname = test.sqlite, table.name = 'snp_test'), index = 'index4')
test.sqlite <- normalizePath(test.sqlite, '/')
file.remove(test.sqlite)
```

`get.annotation.dbtype` *Get annovarR default databases type [sqlite, txt]*

Description

Get annovarR default databases type [sqlite, txt]

Usage

```
get.annotation.dbtype(anno.name = "", database.cfg = system.file("extdata",  
  "config/databases.toml", package = "annovarR"))
```

Arguments

`anno.name` Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
`database.cfg` Configuration file of annovarR databases infomation

Examples

```
get.annotation.dbtype('avsnp147')
```

`get.annotation.names` *Get all annovarR supported databases*

Description

Get all annovarR supported databases

Usage

```
get.annotation.names(database.cfg = system.file("extdata",  
  "config/databases.toml", package = "annovarR"))
```

Arguments

`database.cfg` Configuration file of annovarR databases infomation

Examples

```
cfg <- system.file('extdata', 'config/config.toml', package = 'annovarR')  
get.annotation.names(cfg)
```

```
get.annotation.needcols
```

Get annovarR annotation needed colnames according the anno.name

Description

Get annovarR annotation needed colnames according the anno.name

Usage

```
get.annotation.needcols(anno.name = "",  
  database.cfg = system.file("extdata", "config/databases.toml", package =  
  "annovarR"))
```

Arguments

anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
database.cfg	Configuration file of annovarR databases infomation

Examples

```
get.annotation.dbtype('avsnp147')
```

get.download.name	<i>Use annotation name to get download.name that can be used to download the database use download.database</i>
-------------------	---

Description

Use annotation name to get download.name that can be used to download the database use download.database

Usage

```
get.download.name(anno.name = "", database.cfg = system.file("extdata",  
  "config/databases.toml", package = "annovarR"))
```

Arguments

anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
database.cfg	Configuration file of annovarR databases infomation

Examples

```
get.download.name('avsnp147')
```

mysql.auto.build	<i>Build annovarR database in mysql (auto from extdata/config/database.toml)</i>
------------------	--

Description

Build annovarR database in mysql (auto from extdata/config/database.toml)

Usage

```
mysql.auto.build(anno.name = "", buildver = "hg19",
  database.dir = "/path/", mysql.connect.params = list(user = "", password =
  "", host = "localhost", port = "3306"), overwrite = FALSE, append = FALSE,
  index = "chr_start_index", db.type = "mysql",
  database.cfg = system.file("extdata", "config/databases.toml", package =
  "annovarR"), extra_fread_params = list(sep = "\t", header = TRUE, return_1L
  = FALSE), mysql.build.params = list(fread.params = list(sep = "\t")),
  batch_lines = 1e+07, start_index = 1, new.colnames = NULL,
  verbose = TRUE)
```

Arguments

anno.name	Annotation name, eg. avsn138, avsn147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases
mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
overwrite	Logical indicating wheather overwrite sqlite database, default is FALSE
append	Logical indicating wheather append the data to sqlite database, default is FALSE
index	Index name in sqlite
db.type	Setting the database type (sqlite, txt or mysql)
database.cfg	Configuration file of annovarR databases infomation
extra_fread_params	Pass to batch_file , default is to get value from database.cfg
mysql.build.params	Extra params pass to mysql.build
batch_lines	Parameters pass to batch_file , default is 10000000
start_index	default is 1, control the skip rows, n = (i-1) * batch_lines
new.colnames	Use the fread determined colnames or use new colnames
verbose	Logical indicating wheather print the extra log infomation

Examples

```

i <- 'hg19_avsnp147'
database <- system.file('extdata', sprintf('demo/%s.txt', i), package = 'annovarR')
file.copy(database, sprintf('%s/%s.txt', tempdir(), i))
## Not run:
mysql.auto.build(anno.name = 'avsnp147', database.dir = tempdir(),
mysql.connect.params = list(user = 'username', password = 'password',
host = 'localhost', port = 3306, dbname = 'annovarR'))

## End(Not run)

```

mysql.auto.index	<i>Index annovarR database in mysql (auto from ext-data/config/database.toml)</i>
------------------	---

Description

Index annovarR database in mysql (auto from extdata/config/database.toml)

Usage

```

mysql.auto.index(anno.name = "", buildver = "hg19",
database.dir = "/path/", mysql.connect.params = list(user = "", password =
"", host = "localhost", port = "3306"), index = "chr_start_index",
db.type = "mysql", database.cfg = system.file("extdata",
"config/databases.toml", package = "annovarR"), verbose = TRUE)

```

Arguments

anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases (mysql no need)
mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
index	Index name in sqlite
db.type	Setting the database type (sqlite, txt or mysql)
database.cfg	Configuration file of annovarR databases infomation
verbose	Logical indicating wheather print the extra log infomation

Examples

```

i <- 'hg19_avsnp147'
database <- system.file('extdata', sprintf('demo/%s.txt', i), package = 'annovarR')
file.copy(database, sprintf('%s/%s.txt', tempdir(), i))
## Not run:
mysql.auto.index(anno.name = 'avsnp147', database.dir = tempdir(),

```

```
mysql.connect.params = list(user = 'username', password = 'password',
  host = 'localhost', port = 3306, dbname = 'annovarR'))

## End(Not run)
```

mysql.build

Build annovarR database in mysql

Description

Build annovarR database in mysql

Usage

```
mysql.build(filename = "", mysql.connect.params = list(host = "", dbname =
  "", table.name = "", user = "", password = ""), dat = data.table(),
  fread.params = list(), new.colnames = NULL, verbose = FALSE, ...)
```

Arguments

filename	Path of raw data, will be read by fread
mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
dat	Object of data.table, as the input data to build database
fread.params	Other parameters be used in fread
new.colnames	New colnames of table, default is to retain the original
verbose	Ligical indicating wheather show the log message
...	Other parameters be used in dbWriteTable

Examples

```
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
## Not run:
mysql.build(test.dat, list(host = 'host', dbname = 'db',
  table.name = 'table', user = 'user', password = 'password'))

## End(Not run)
```

mysql.head	<i>Show top n line of table of database in mysql database</i>
------------	---

Description

Show top n line of table of database in mysql database

Usage

```
mysql.head(mysql.connect.params = list(host = "", dbname = "", table.name =
"", user = "", password = ""), n = 10, extra.sql = NULL,
verbose = FALSE, ...)
```

Arguments

mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
n	n lines will be selected
extra.sql	Extra sql statement
verbose	Ligical indicating wheather show the log message
...	Other parameters pass to dbGetQuery

Examples

```
host <- '11.11.11.1'
```

mysql.index	<i>Generate search index of annovarR database table in mysql</i>
-------------	--

Description

Generate search index of annovarR database table in mysql

Usage

```
mysql.index(mysql.connect.params = list(host = "", dbname = "", table.name =
"", user = "user", password = "password"), index = "", cols = c(),
verbose = FALSE, ...)
```

Arguments

mysql.connect.params	Mysql parameters, [host, dbname, table.name, user, password etc.]
index	Index name in mysql
cols	Colnames needed with length to be index (e.g. c('V1(6)', 'V2'))
verbose	Ligical indicating wheather show the log message
...	Other parameters be used in dbSendQuery

Examples

```
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
## Not run:
mysql.index(list(host = 'host', dbname = 'db', table.name = 'table',
user = 'user', password = 'password'), index = 'index_name', cols = c('V1', 'V2'))

## End(Not run)
```

mysql.tb.colnames *Get colnames of table of database in mysql*

Description

Get colnames of table of database in mysql

Usage

```
mysql.tb.colnames(mysql.connect.params = list(host = "", dbname = "", user =
"", password = "", table.name = ""), ...)
```

Arguments

```
mysql.connect.params      Mysql parameters, [host, dbname, table.name, user, password etc.]
...                        Other parameters pass to dbConnect
```

Examples

```
## Not run:
mysql.db.colnames(list(host = 'host', dbname = 'db', user = 'user',
password = 'password', table.name = 'table'))

## End(Not run)
```

mysql.tb.indexes *Get mysql table indexes*

Description

Get mysql table indexes

Usage

```
mysql.tb.indexes(mysql.connect.params = list(dbname = "", table.name = ""))
```

Arguments

`mysql.connect.params`
 Connect to mysql database params [dbname, table.name, host, user, password]

Examples

NULL

sql2sqlite *Convert sql file to sqlite database*

Description

Convert sql file to sqlite database

Usage

```
sql2sqlite(sql.file = "", statements = "", dbname = "", verbose = FALSE,
  ...)
```

Arguments

`sql.file` SQL file of sqlite database dumped
`statements` SQL statements split by ';' (small dataset)
`dbname` Path of output sqlite database
`verbose` Logical indicating wheather show the log message
`...` Other parameters be used in `dbSendQuery`

Examples

```
sql.file <- system.file('extdata', 'demo/hg19_avsnp147.sqlite.sql', package = 'annovarR')
out.sqlite <- tempfile()
sqlite <- Sys.which(c('sqlite3', 'sqlite'))
sqlite <- sqlite[sqlite != ''][1]
sqlite <- unname(sqlite)
if (!is.na(sqlite)) {
  sql2sqlite(sql.file = sql.file, dbname = out.sqlite,
    verbose = FALSE)
  unlink(out.sqlite)
}
statements <- paste0(readLines(sql.file), collapse = '\n')
sql2sqlite(statements = statements, dbname = out.sqlite,
  verbose = FALSE)
unlink(out.sqlite)
```

sqlite.auto.build	<i>Build annovarR database in sqlite (auto from ext-data/config/database.toml)</i>
-------------------	--

Description

Build annovarR database in sqlite (auto from extdata/config/database.toml)

Usage

```
sqlite.auto.build(anno.name = "", buildver = "hg19",
  database.dir = "/path/", overwrite = FALSE, append = FALSE,
  index = "chr_start_index", db.type = "sqlite",
  database.cfg = system.file("extdata", "config/databases.toml", package =
  "annovarR"), extra_fread_params = list(sep = "\t", header = TRUE, return_1L
  = FALSE), sqlite.build.params = list(fread.params = list(sep = "\t")),
  batch_lines = 1e+07, start_index = 1, new.colnames = NULL,
  verbose = TRUE)
```

Arguments

anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases
overwrite	Logical indicating wheather overwrite sqlite database, default is FALSE
append	Logical indicating wheather append the data to sqlite database, default is FALSE
index	Index name in sqlite
db.type	Setting the database type (sqlite, txt or mysql)
database.cfg	Configuration file of annovarR databases infomation
extra_fread_params	Pass to batch_file , default is to get value from database.cfg
sqlite.build.params	Extra params pass to sqlite.build
batch_lines	Parameters pass to batch_file , default is 10000000
start_index	default is 1, control the skip rows, n = (i-1) * batch_lines
new.colnames	Use the fread determined colnames or use new colnames
verbose	Logical indicating wheather print the extra log infomation

Examples

```
i <- 'hg19_avsnp147'
database <- system.file('extdata', sprintf('demo/%s.txt', i), package = 'annovarR')
file.copy(database, sprintf('%s/%s.txt', tempdir(), i))
sqlite.auto.build('avsnp147', 'hg19', database.dir = tempdir(), verbose = TRUE)
unlink(sprintf('%s/%s.txt', tempdir(), i))
unlink(sprintf('%s/%s.sqlite', tempdir(), i))
```

sqlite.auto.index	<i>Index annovarR database in sqlite (auto from extdata/config/database.toml)</i>
-------------------	---

Description

Index annovarR database in sqlite (auto from extdata/config/database.toml)

Usage

```
sqlite.auto.index(anno.name = "", buildver = "hg19",
  database.dir = "/path/", index = "chr_start_index", db.type = "sqlite",
  database.cfg = system.file("extdata", "config/databases.toml", package =
  "annovarR"), verbose = TRUE)
```

Arguments

anno.name	Annotation name, eg. avsnp138, avsnp147, 1000g2015aug_all
buildver	Genome version, hg19, hg38, mm10 and others
database.dir	Dir of the databases
index	Index name in sqlite
db.type	Setting the database type (sqlite, txt or mysql)
database.cfg	Configuration file of annovarR databases infomation
verbose	Logical indicating wheather print the extra log infomation

Examples

```
i <- 'hg19_avsnp147'
database <- system.file('extdata', sprintf('demo/%s.txt', i), package = 'annovarR')
file.copy(database, sprintf('%s/%s.txt', tempdir(), i))
sqlite.auto.build('avsnp147', 'hg19', database.dir = tempdir(), verbose = TRUE)
sqlite.auto.index('avsnp147', 'hg19', database.dir = tempdir(), index = 'chr_start_index2',
  verbose = TRUE)
unlink(sprintf('%s/%s.txt', tempdir(), i))
unlink(sprintf('%s/%s.sqlite', tempdir(), i))
```

sqlite.build	<i>Build annovarR database in sqlite</i>
--------------	--

Description

Build annovarR database in sqlite

Usage

```
sqlite.build(filename = NULL, sqlite.connect.params = list(dbname = "",
  table.name = ""), dat = data.table(), fread.params = list(),
  new.colnames = NULL, overwrite = TRUE, verbose = FALSE, ...)
```

Arguments

filename	Path of raw data, will be read by fread
sqlite.connect.params	Connect to sqlite database params [dbname, table.name]
dat	Object of data.table, as the input data to build database (optional)
fread.params	Other parameters be used in fread
new.colnames	New colnames of table, default is to retain the original
overwrite	Ligical indicating wheather overwrite sqlite database
verbose	Ligical indicating wheather show the log message
...	Other parameters be used in dbWriteTable

Examples

```
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,
table.name = 'snp_test'))
test.sqlite <- normalizePath(test.sqlite, '/')
file.remove(test.sqlite)
```

sqlite.head	<i>Show top n line of table of database in sqlite database</i>
-------------	--

Description

Show top n line of table of database in sqlite database

Usage

```
sqlite.head(sqlite.connect.params = list(dbname = "", table.name = ""),
  n = 10, extra.sql = NULL, verbose = FALSE, ...)
```

Arguments

sqlite.connect.params	Connect to sqlite database params [dbname, table.name]
n	n lines will be selected
extra.sql	Extra sql statement
verbose	Ligical indicating wheather show the log message
...	Other parameters pass to dbGetQuery

Examples

```
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,
table.name = 'snp_test'))
sqlite.head(list(dbname = test.sqlite, table.name = 'snp_test'))
```

 sqlite.index

Generate search index of annovarR database table in sqlite

Description

Generate search index of annovarR database table in sqlite

Usage

```
sqlite.index(sqlite.connect.params = list(dbname = "", table.name = ""),
  index = "", cols = c(), verbose = FALSE, ...)
```

Arguments

sqlite.connect.params	Connect to sqlite database params [dbname, table.name]
index	Index name in sqlite
cols	Colnames needed to be index
verbose	Ligical indicating wheather show the log message
...	Other parameters be used in dbSendQuery

Examples

```
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,
table.name = 'snp_test'))
x <- sqlite.index(list(dbname = test.sqlite, table.name = 'snp_test'),
index = 'index4', cols = c('V1', 'V2'))
test.sqlite <- normalizePath(test.sqlite, '/')
file.remove(test.sqlite)
```

sqlite.tb.colnames *Get colnames of table of database in sqlite*

Description

Get colnames of table of database in sqlite

Usage

```
sqlite.tb.colnames(sqlite.connect.params = list(dbname = "", table.name = ""))
```

Arguments

sqlite.connect.params
Connect to sqlite database params [dbname, table.name]

Examples

```
test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())  
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')  
x <- sqlite.build(filename = test.dat, list(dbname = test.sqlite,  
table.name = 'snp_test'))  
sqlite.tb.colnames(list(dbname = test.sqlite, table.name = 'snp_test'))
```

sqlite.tb.indexes *Get sqlite table indexes*

Description

Get sqlite table indexes

Usage

```
sqlite.tb.indexes(sqlite.connect.params = list(dbname = "", table.name = ""))
```

Arguments

sqlite.connect.params
Connect to sqlite database params [dbname, table.name]

Examples

```

test.sqlite <- sprintf('%s/snp.test.sqlite', tempdir())
test.dat <- system.file('extdata', 'demo/sqlite.dat.txt', package = 'annovarR')
params <- list(dbname = test.sqlite,
table.name = 'snp_test')
x <- sqlite.build(filename = test.dat, params)
x <- sqlite.index(params, index = 'index4', cols = c('V1', 'V2'))
indexes <- sqlite.tb.indexes(params)
test.sqlite <- normalizePath(test.sqlite, '/')
file.remove(test.sqlite)

```

sqlite2sql

Function to dump sqlite database (Now only use system version sqlite)

Description

Function to dump sqlite database (Now only use system version sqlite)

Usage

```

sqlite2sql(sqlite_bin = c(Sys.which("sqlite"), Sys.which("sqlite3")),
  dbname = "", out.sql = "",
  cmd = "{sqlite_bin} {dbname} '.dump' | gzip > {out.sql}", debug = FALSE,
  ...)

```

Arguments

sqlite_bin	Sqlite executable file
dbname	Sqlite database path
out.sql	Output SQL (gzip format)
cmd	The command be used to dump
debug	If set TRUE, only print the command
...	Other parameters pass to system

Examples

```

sqlite2sql('sqlite3', 'default.sqlite', debug = TRUE)

```

vcfanno	<i>R function to run vcfanno</i>
---------	----------------------------------

Description

R function to run vcfanno

Usage

```
vcfanno(vcfanno = Sys.which(c("vcfanno", "vcfanno_osx", "vcfanno_linux64")),
        vcfanno.database.cfg = system.file("extdata", "demo/vcfanno_demo/conf.toml",
        package = "annovarR"), base_path = "", lua = "", ends = FALSE,
        input.file = "input.vcf", out = "output.vcf", thread = 2,
        permissive_overlap = FALSE, debug = FALSE)
```

Arguments

vcfanno	Executable file of vcfanno (Download from https://github.com/brentp/vcfanno/releases)
vcfanno.database.cfg	vcfanno required database configuration file (Not the annovarR database.cfg)
base_path	Optional base_path to prepend to annotation files in the config
lua	Optional path to a file containing custom javascript functions to be used as ops
ends	Annotate the start and end as well as the interval itself.
input.file	Input file path (VCF only)
out	Output file path
thread	number of processes to use. (default 2)
permissive_overlap	annotate with an overlapping variant even it doesn't share the same ref and alt alleles. Default is to require exact match between variants.
debug	If set TRUE, only print the command

Examples

```
vcfanno(debug = TRUE)
```

vep *R function to run VEP*

Description

R function to run VEP

Usage

```
vep(vep = Sys.which("vep"), cache = TRUE, cache_version = 91,
    offline = TRUE, buildver = "GRCh37", dir = file.path(Sys.getenv("HOME"),
    ".vep"), input.file = "", out = "variant_effect_output.txt", fasta = "",
    everything = TRUE, extra.params = "", debug = FALSE)
```

Arguments

vep	Executable file of vep
cache	Enables use of the cache. Add <code>-refseq</code> or <code>-merged</code> to use the refseq or merged cache, (if installed).
cache_version	Use a different cache version than the assumed default (the VEP version). This should be used with Ensembl Genomes caches since their version numbers do not match Ensembl versions. For example, the VEP/Ensembl version may be 88 and the Ensembl Genomes version 35. Not used by default
offline	Enable offline mode. No database connections will be made, and a cache file or GFF/GTF file is required for annotation. Add <code>-refseq</code> to use the refseq cache (if installed). Not used by default
buildver	Select the assembly version to use if more than one available. If using the cache, you must have the appropriate assembly's cache file installed. If not specified and you have only 1 assembly version installed, this will be chosen by default. Default = use found assembly version (GRCh37)
dir	Specify the base cache/plugin directory to use. Default = <code>'\$HOME/.vep/'</code>
input.file	Input file name. If not specified, the script will attempt to read from STDIN.
out	Output file name. The script can write to STDOUT by specifying STDOUT as the output file name - this will force quiet mode. Default = <code>'variant_effect_output.txt'</code>
fasta	Specify a FASTA file or a directory containing FASTA files to use to look up reference sequence. The first time you run the script with this parameter an index will be built which can take a few minutes. This is required if fetching HGVS annotations (<code>-hgvs</code>) or checking reference sequences (<code>-check_ref</code>) in offline mode (<code>-offline</code>), and optional with some performance increase in cache mode (<code>-cache</code>). See documentation for more details. Not used by default
everything	Shortcut flag to switch on all of the following: <code>-sift b</code> , <code>-polyphen b</code> , <code>-ccds</code> , <code>-uniprot</code> , <code>-hgvs</code> , <code>-symbol</code> , <code>-numbers</code> , <code>-domains</code> , <code>-regulatory</code> , <code>-canonical</code> , <code>-protein</code> , <code>-biotype</code> , <code>-uniprot</code> , <code>-tsl</code> , <code>-appris</code> , <code>-gene_phenotype -af</code> , <code>-af_1kg</code> , <code>-af_esp</code> , <code>-af_gnomad</code> , <code>-max_af</code> , <code>-pubmed</code> , <code>-variant_class</code>
extra.params	Extra paramters in vep command
debug	If set TRUE, only print the command

vep

31

Examples

```
vep(debug = TRUE)
```

Index

annotation, [3, 6](#)
annotation.cols.match, [3, 4](#)
annotation.merge, [6](#)
annotation.region.match, [3, 6](#)
annovar, [3, 8](#)
annovarR, [10](#)
annovarR-package (annovarR), [11](#)

batch_file, [17, 23](#)

del, [11](#)
download.database, [12](#)
drop.mysql.index, [13](#)
drop.sqlite.index, [14](#)

fread, [19, 25](#)

get.annotation.dbtype, [15](#)
get.annotation.names, [15](#)
get.annotation.needcols, [16](#)
get.download.name, [16](#)

install.bioinfo, [13](#)

mysql.auto.build, [17](#)
mysql.auto.index, [18](#)
mysql.build, [17, 19](#)
mysql.head, [20](#)
mysql.index, [20](#)
mysql.tb.colnames, [21](#)
mysql.tb.indexes, [21](#)

sql2sqlite, [22](#)
sqlite.auto.build, [23](#)
sqlite.auto.index, [24](#)
sqlite.build, [23, 25](#)
sqlite.head, [25](#)
sqlite.index, [26](#)
sqlite.tb.colnames, [27](#)
sqlite.tb.indexes, [27](#)
sqlite2sql, [28](#)

system, [28](#)
vcfanno, [29](#)
vep, [3, 30](#)