Package 'echotabix'

August 31, 2021

Type Package

Title echoverse module: Tabix indexing and querying.

Version 0.99.0

Description echoverse module: Tabix indexing and querying.

URL https://github.com/RajLabMSSM/echotabix

BugReports https://github.com/RajLabMSSM/echotabix/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

SystemRequirements Python (>= 3.7.0)

biocViews

Imports magrittr,
dplyr,
utils,
R.utils,
stats,
methods,
Matrix,
parallel,
BiocManager,
data.table,
GenomeInfoDb,
IRanges,
GenomicRanges,
VariantAnnotation,
Rsamtools,
seqminer,
rtracklayer

Suggests markdown,
rmarkdown,
remotes,
knitr,
BiocStyle,
covr,
testthat (>= 3.0.0)

RoxygenNote 7.1.1

VignetteBuilder knitr

License GPL (>= 3) + file LICENSE

Config/testthat/edition 3

R topics documented:

BST1 **echolocatoR** *output example: BST1 locus*

Description

An example results file after running `finemap_loci` on the *BST1* locus.

Usage

```
data("BST1")
```

Format

data.table

SNP SNP RSID

CHR Chromosome

POS Genomic position (in basepairs)

... Optional: extra columns

Nalls2019

data.table

Details

Data originally comes from the Parkinson's disease GWAS by Nalls et al., (bioRxiv).

Source

```
root_dir <-"~/Desktop/Fine_Mapping/Data/GWAS/Nalls23andMe_2019/BST1/Multi-finemap"
BST1 <-data.table::fread(file.path(root_dir,"Multi-finemap_results.txt")) BST1 <-update_cols(dat
= BST1) BST1 <-find_consensus_SNPs(dat = BST1) usethis::use_data(BST1,overwrite = TRUE)
```

convert *Convert summary stats file to tabix format*

Description

Convert summary stats file to tabix format

Usage

```
convert(fullSS_path, chrom_col = "CHR", position_col = "POS", verbose = TRUE)
```

See Also

Other tabix: `convert_and_query()`

Examples

```
## Not run:
fullSS_path <- echolocatoR::example_fullSS()
fullSS_tabix <- convert(fullSS_path = fullSS_path, position_col = "BP")

## End(Not run)
```

convert_and_query *Convert and query*

Description

If it is not tabix format already (determined by checking for a .tbi file of the same name in the same directory), the full summary statistics file is converted into tabix format for super fast querying. A query is then made using the min/max genomic positions to extract a locus-specific summary stats file.

Usage

```
convert_and_query(
  fullSS_path,
  study_dir = NULL,
  subset_path = tempfile(".tsv.gz"),
  is_tabix = FALSE,
  chrom_col = "CHR",
  position_col = "BP",
  min_POS,
  max_POS,
  chrom,
  save_subset = TRUE,
  nThread = 1,
  conda_env = "echoR",
  verbose = TRUE
)
```

Value

data.table of locus subset summary statistics

See Also

Other tabix: `convert()`

Examples

```
## Not run:
data("BST1")
fullSS_path <- echolocatoR::example_fullSS()

subset_path <- file.path(tempdir(), "BST1_Nalls23andMe_2019_subset.tsv.gz")
dat <- convert_and_query(
    fullSS_path = fullSS_path,
    subset_path = subset_path,
    min_POS = min(BST1$POS),
    max_POS = max(BST1$POS),
    chrom = BST1$CHR[1]
)

## End(Not run)
```

`liftover` *Genome build liftover*

Description

Transfer genomic coordinates from one genome build to another.

Usage

```
liftover(
  dat,
  chrom_col = "CHR",
  start_col = "POS",
  end_col = start_col,
  build_conversion = c("hg38ToHg19", "hg19ToHg38"),
  as_granges = FALSE,
  verbose = TRUE
)
```

Arguments

`dat`	SNP-level data table.
`chrom_col`	Name of the chromosome column.
`start_col`	Name of the start position column.
`end_col`	Name of the end position column (can be same as `start_col` if all data is SNP-level).

build_conversion
: From which to which genome build to lift over dat.

as_granges
: Return lifted dat as GenomicRanges object.

verbose
: Print messages.

Source

liftOver

UCSC chain files

Examples

```
data("BST1")
dat_lifted <- liftover(dat = BST1, build_conversion = "hg19ToHg38")
```

---

locus_dir *Example results path for BST1 locus*

Description

Example results path for BST1 locus

Usage

```
data("locus_dir")
```

Format

path string

Source

```
locus_dir <-"results/GWAS/Nalls23andMe_2019/BST1" usethis::use_data(locus_dir,overwrite
= T)
```

---

query_tabular *Query a tabix file*

Description

Query by genomic coordinates.

Usage

```
query_tabular(fullSS_tabix, chrom, start_pos, end_pos, verbose = TRUE)
```

Examples

```
## Not run:
data("BST1")
fullSS_path <- echolocatoR::example_fullSS()
fullSS_tabix <- convert(fullSS_path = fullSS_path, position_col = "BP")
tab <- query_tabular(
    fullSS_tabix = fullSS_tabix,
    chrom = BST1$CHR[1],
    start_pos = min(BST1$POS),
    end_pos = max(BST1$POS)
)

## End(Not run)
```

---

query_vcf *Query VCF file*

Description

Query a (remote) Variant Call Format (VCF) file.

Usage

```
query_vcf(
  dat,
  vcf_url,
  locus_dir = file.path(tempdir(), "LD"),
  vcf_name = gsub(".vcf|.gz.|bgz", "", basename(vcf_url)),
  ref_genome = "GRCh37",
  samples = NULL,
  force_new_vcf = FALSE,
  verbose = TRUE
)
```

Arguments

dat	SNP-level data table.
vcf_url	URL or path to VCF.
locus_dir	Directory to store LD in.
vcf_name	VCF reference name (e.g. "1KGphase1").
ref_genome	Genome build of the VCF file.
samples	Sample names to subset the VCF by before computing LD.
force_new_vcf	Force the creation of a new LD file even if one exists.
verbose	Print messages.

Value

VCF object.

See Also

Other LD: construct_subset_vcf_name(), get_locus_vcf_folder()

Index