Package 'echoannot'

September 18, 2021

Type Package

Title echoverse module: Annotate fine-mapping results

Version 0.99.1

Description echoverse module: Annotate fine-mapping results.

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

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

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

SystemRequirements Python (>= 3.7.0)

biocViews

Imports echodata,
echotabix,
dplyr,
data.table,
utils,
stats,
tidyr,
parallel,
haploR,
ggplot2,
patchwork,
ggbio,
RColorBrewer,
scales,
GenomicRanges,
DescTools,
pheatmap,
grDevices,
rtracklayer,
S4Vectors,
GenomeInfoDb,
biomaRt,
IRanges

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

Remotes github::RajLabMSSM/echodata,
github::RajLabMSSM/echotabix

RoxygenNote 7.1.2

VignetteBuilder knitr

License GPL-3

Config/testthat/edition 3

R topics documented:

`annotate_missense` *Annotate any missense variants*

Description

Annotate any missense variants

Usage

```
annotate_missense(merged_DT, snp_filter = "Support>0")
```

See Also

Other annotate: biomart_geneInfo(), biomart_snp_info(), biomart_snps_to_geneInfo(), haplor_epigenetics_enrichment(), haplor_epigenetics_summary(), haplor_haploreg(), haplor_regulomedb(), plot_missense(), snps_by_mutation_type()

Examples

```
## Not run:
annotated_DT <- annotate_missense(
    merged_DT = echodata::Nalls2019_merged,
    snp_filter = "Support>0"
)

## End(Not run)
```

CORCES_2020.bulkATACseq_peaks

bulkATACseq peaks from human brain tissue

Description

Each row represents an individual peak identified in the bulk ATAC-seq data.

Usage

```
CORCES_2020.bulkATACseq_peaks
```

Format

An object of class `data.table` (inherits from `data.frame`) with 186559 rows and 10 columns.

Details

Data originally from Corces et al. (bioRxiv), as of May 2020. Specifically: *STable2_Features_bulkATAC-seq_Peaks*

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.get_HiChIP_FitHiChIP_overlap(), CORCES_2020.prepa
CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks, CORCES_2020.scATA

Examples

```
## Not run:
dat <- readxl::read_excel(
    file.path(
        "~/Desktop/Fine_Mapping/echolocatoR",
        "annotations/Coceres_2020",
        "STable2_Features_bulkATAC-seq_Peaks.xlsx"
    ),
    skip = 18
)
CORCES_2020.bulkATACseq_peaks <- data.table::data.table(dat)
usethis::use_data(CORCES_2020.bulkATACseq_peaks, overwrite = TRUE)

## End(Not run)
```

`CORCES_2020.cicero_coaccessibility`

Cicero_coaccessibility from human brain tissue

Description

Cicero coaccessibility analysis for peaks that overlap SNPs derived from analysis of scATAC-seq data. Each row represents an individual peak identified from the feature binarization analysis (see methods).

Usage

```
CORCES_2020.cicero_coaccessibility
```

Format

An object of class `data.table` (inherits from `data.frame`) with 9795 rows and 14 columns.

Details

Data originally from Corces et al. (bioRxiv), as of May 2020. Specifically: *STable10_Coacessibility_Peak_loop_connect* *Cicero Coaccessibility* sheet. Peak_ID_Peak1 - A unique number that identifies the peak across supplementary tables.

Column dictionary:

hg38_Chromosome_Peak1 The hg38 chromosome of the first loop Peak.

hg38_Start_Peak1 The hg38 start position of the first loop Peak.

hg38_Stop_Peak1 The hg38 stop position of the first loop Peak.

Width_Peak1 The width of the first loop Peak.

Peak_ID_Peak2 A unique number that identifies the peak across supplementary tables.

hg38_Chromosome_Peak2 The hg38 chromosome of the second loop Peak.

hg38_Start_Peak2 The hg38 start position of the second loop Peak.

hg38_Stop_Peak2 The hg38 stop position of the second loop Peak.

Width_Peak2 The width of the second loop Peak.

Coaccessibility The coaccessibility correlation for the given peak pair.

Peak1_hasSNP A boolean variable determining whether the first peak overlaps a SNP from our AD/PD GWAS analyses.

Peak2_hasSNP A boolean variable determining whether the second peak overlaps a SNP from our AD/PD GWAS analyses.

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.bulkATACseq_peaks, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.get_HiChIP_FitHiChIP_overlap(), CORCES_2020.prepa CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks, CORCES_2020.scATA

Examples

```
## Not run:
dat <- readxl::read_excel(
    file.path(
        "~/Desktop/Fine_Mapping/echolocatoR/annotations",
        "Coceres_2020/STable10_Coacessibility_Peak_loop_connection.xlsx"
    ),
    skip = 21, sheet = 2
)
CORCES_2020.cicero_coaccessibility <- data.table::data.table(dat)
usethis::use_data(CORCES_2020.cicero_coaccessibility)

## End(Not run)
```

CORCES_2020.get_ATAC_peak_overlap

Get overlap between datatable of SNPs and scATAC peaks

Description

Can optionally add `Cicero` coaccessibility scores, which are also derived from scATAC-seq data.

Usage

```
CORCES_2020.get_ATAC_peak_overlap(
  finemap_dat,
  FDR_filter = NULL,
  add_cicero = TRUE,
  cell_type_specific = TRUE,
  verbose = TRUE
)
```

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.bulkATACseq_peaks, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_HiChIP_FitHiChIP_overlap(), CORCES_2020.prepar CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks, CORCES_2020.scATA

CORCES_2020.get_HiChIP_FitHiChIP_overlap

Get overlap between data table of SNPs and HiChIP_FitHiChIP coaccessibility anchors

---

Description

Anchors are the genomic regions that have evidence of being functionally connected to one another (coaccessible), e.g. enhancer-promoter interactions.

Usage

```
CORCES_2020.get_HiChIP_FitHiChIP_overlap(finemap_dat, verbose = TRUE)
```

Arguments

finemap_dat	Fine-mapping results.
verbose	Print messages.

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.bulkATACseq_peaks, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.prepare_bulkA CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks, CORCES_2020.scATA

CORCES_2020.HiChIP_FitHiChIP_loop_calls

FitHiChIP loop calls from human brain tissue

Description

FitHiChIP loop calls that overlap SNPs derived from analysis of H3K27ac HiChIP data. Each row represents an individual peak identified from the feature binarization analysis (see methods).

Usage

```
CORCES_2020.HiChIP_FitHiChIP_loop_calls
```

Format

An object of class data.table (inherits from data.frame) with 11542 rows and 11 columns.

Details

Data originally from Corces et al. (bioRxiv), as of May 2020. Specifically: *STable10_Coacessibility_Peak_loop_connect*
HiChIP FitHiChIP Loop Calls sheet.

Column dictionary

hg38_Chromosome_Anchor1 The hg38 chromosome of the first loop Anchor.

hg38_Start_Anchor1 The hg38 start position of the first loop Anchor.

hg38_Stop_Anchor1 The hg38 stop position of the first loop Anchor.

Width_Anchor1 The width of the first loop Anchor.

hg38_Chromosome_Anchor2 The hg38 chromosome of the second loop Anchor.

hg38_Start_Anchor2 The hg38 start position of the second loop Anchor.

hg38_Stop_Anchor2 The hg38 stop position of the second loop Anchor.

Width_Anchor2 The width of the second loop Anchor.

Score The -log10(q-value) of the loop call from FitHiChIP.

Anchor1_hasSNP A boolean variable determining whether the first anchor overlaps a SNP from our AD/PD GWAS analyses.

Anchor2_hasSNP A boolean variable determining whether the second anchor overlaps a SNP from our AD/PD GWAS analyses.

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.bulkATACseq_peaks, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.get_HiChIP_FitHiChIP_overlap(), CORCES_2020.prepa CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks, CORCES_2020.scATA

Examples

```
## Not run:
dat <- readxl::read_excel(
    file.path(
        "~/Desktop/Fine_Mapping/echolocatoR/annotations",
        "Coceres_2020/STable10_Coacessibility_Peak_loop_connection.xlsx"
    ),
    skip = 19, sheet = 1
)
CORCES_2020.HiChIP_FitHiChIP_loop_calls <- data.table::data.table(dat)
usethis::use_data(CORCES_2020.HiChIP_FitHiChIP_loop_calls)

## End(Not run)
```

CORCES_2020.scATACseq_celltype_peaks
scATACseq cell type-specific peaks from human brain tissue

Description

Each row represents an individual peak identified from the feature binarization analysis (see methods).

Usage

```
CORCES_2020.scATACseq_celltype_peaks
```

Format

An object of class `data.table` (inherits from `data.frame`) with 221062 rows and 13 columns.

Details

Data originally from Corces et al. (bioRxiv), as of May 2020. Specifically: *STable6_Features_scATAC-seq_celltype_Peaks*

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.bulkATACseq_peaks, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.get_HiChIP_Fi CORCES_2020.prepare_bulkATAC_peak_overlap(), CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_peaks

Examples

```
## Not run:
dat <- readxl::read_excel(
    file.path(
        "~/Desktop/Fine_Mapping/echolocatoR/annotations",
        "Coceres_2020/STable6_Features_scATAC-seq_celltype_Peaks.xlsx"
    ),
    skip = 15
)
CORCES_2020.scATACseq_celltype_peaks <- data.table::data.table(dat)
usethis::use_data(CORCES_2020.scATACseq_celltype_peaks, overwrite = TRUE)

## End(Not run)
```

CORCES_2020.scATACseq_peaks

scATACseq peaks from human brain tissue

Description

Each row represents an individual peak identified in the single-cell ATAC-seq data.

Usage

```
CORCES_2020.scATACseq_peaks
```

Format

An object of class `data.table` (inherits from `data.frame`) with 359022 rows and 10 columns.

Details

Data originally from Corces et al. (bioRxiv), as of May 2020. Specifically: *STable5_Features_scATAC-seq_Peaks_all*

Source

https://doi.org/10.1038/s41588-020-00721-x

See Also

Other CORCES_2020: CORCES_2020.HiChIP_FitHiChIP_loop_calls, CORCES_2020.bulkATACseq_peaks, CORCES_2020.cicero_coaccessibility, CORCES_2020.get_ATAC_peak_overlap(), CORCES_2020.get_HiChIP_Fi CORCES_2020.prepare_bulkATAC_peak_overlap(), CORCES_2020.prepare_scATAC_peak_overlap(), CORCES_2020.scATACseq_celltype_peaks

Examples

```
## Not run:
dat <- readxl::read_excel(
    file.path(
        "~/Desktop/Fine_Mapping/echolocatoR/annotations",
        "Coceres_2020/STable5_Features_scATAC-seq_Peaks_all.xlsx"
    ),
    skip = 18
)
CORCES_2020.scATACseq_peaks <- data.table::data.table(dat)
usethis::use_data(CORCES_2020.scATACseq_peaks, overwrite = TRUE)

## End(Not run)
```

CS_bin_plot *Plot CS bin counts*

Description

Plot CS bin counts

Usage

```
CS_bin_plot(merged_DT, show_plot = TRUE)
```

See Also

Other summarise: CS_counts_plot(), get_CS_bins(), get_CS_counts(), get_SNPgroup_counts(), peak_overlap_plot(), plot_dataset_overlap(), results_report(), super_summary_plot()

Examples

```
bin_plot <- CS_bin_plot(merged_DT = echodata::Nalls2019_merged)
```

CS_counts_plot *Bar plot of tool-specific CS sizes*

Description

Loci ordered by UCS size (smallest to largest).

Usage

```
CS_counts_plot(
  merged_DT,
  show_numbers = TRUE,
  ylabel = "Locus",
  legend_nrow = 3,
  label_yaxis = TRUE,
  top_CS_only = FALSE,
  show_plot = TRUE
)
```

See Also

Other summarise: CS_bin_plot(), get_CS_bins(), get_CS_counts(), get_SNPgroup_counts(), peak_overlap_plot(), plot_dataset_overlap(), results_report(), super_summary_plot()

Examples

```
gg_CS <- CS_counts_plot(merged_DT = echodata::Nalls2019_merged)
```

get_SNPgroup_counts *Tally locus-specific SNP group sizes*

Description

Tally locus-specific SNP group sizes

Usage

```
get_SNPgroup_counts(merged_DT, grouping_vars = "Locus")
```

See Also

Other summarise: CS_bin_plot(), CS_counts_plot(), get_CS_bins(), get_CS_counts(), peak_overlap_plot(), plot_dataset_overlap(), results_report(), super_summary_plot()

Examples

```
data("merged_DT")
snp_groups <- get_SNPgroup_counts(merged_DT = echodata::Nalls2019_merged)
```

merge_celltype_specific_epigenomics *Merge all cell-type-specific epigenomics*

Description

Merges multiple cell-type-specific epigenomic datasets (Nott 2019, Corces 2020) into a single GRanges object.

Usage

```
merge_celltype_specific_epigenomics(keep_extra_cols = FALSE)
```

Examples

```
gr.merged <- merge_celltype_specific_epigenomics()
```

`merge_finemapping_results` *Merge fine-mapping results from all loci*

Description

Gather fine-mapping results from *echolocatoR* across all loci and merge into a single data.frame.

Usage

```
merge_finemapping_results(
  dataset = "./Data/GWAS",
  minimum_support = 1,
  include_leadSNPs = TRUE,
  LD_reference = NULL,
  save_path = tempfile(fileext = "merged_results.csv.gz"),
  from_storage = TRUE,
  haploreg_annotation = FALSE,
  regulomeDB_annotation = FALSE,
  biomart_annotation = FALSE,
  PP_threshold = 0.95,
  consensus_threshold = 2,
  exclude_methods = NULL,
  top_CS_only = FALSE,
  verbose = TRUE,
  nThread = 1
)
```

Arguments

`dataset`	Path to the folder you want to recursively search for results files within (e.g. `"Data/GWAS/Nalls23andMe_2019"`). Set this to a path that includes multiple subfolders if you want to gather results from multiple studies at once (e.g. `"Data/GWAS"`).
`minimum_support`	Filter SNPs by the minimum number of fine-mapping tools that contained the SNP in their Credible Set.
`include_leadSNPs`	Include lead GWAS/QTL SNPs per locus (regardless of other filtering criterion).
`from_storage`	Search for stored results files.
`haploreg_annotation`	Annotate SNPs with HaploReg (using `HaploR`).
`regulomeDB_annotation`	Annotate SNPs with regulaomeDB (using `HaploR`).
`biomart_annotation`	Annotate SNPs with `biomart`.
`PP_threshold`	Mean posterior probability threshold to include SNPs in mean PP Credible Set (averaged across all fine-mapping tools).

exclude_methods
: Exclude certain fine-mapping methods when estimating **mean.CS** and **Consensus_SNP**.

verbose
: Print messages.

xlsx_path
: Save merged data.frame as excel file.

consensus_thresh
: The minimum number of tools that have the SNPs in their Credible Set to classify it as a **Consensus_SNP**.

`NOTT_2019.bigwig_metadata` *Metadata and links to data*

Description

Metadata for cell type-specific epigenomic bigWig files hosted on UCSC Genome Browser. bigWig files contain the genomic ranges from each epigenomic assay, as well as a Score column which describes the peaks of the aggregate reads.

Usage

```
NOTT_2019.bigwig_metadata
```

Format

An object of class `data.table` (inherits from `data.frame`) with 18 rows and 14 columns.

Source

https://science.sciencemag.org/content/366/6469/1134

See Also

Other NOTT_2019: `NOTT_2019.epigenomic_histograms()`, `NOTT_2019.get_epigenomic_peaks()`, `NOTT_2019.get_interactions()`, `NOTT_2019.get_interactome()`, `NOTT_2019.get_promoter_celltypes()`, `NOTT_2019.get_promoter_interactome_data()`, `NOTT_2019.get_regulatory_regions()`, `NOTT_2019.interacto`
`NOTT_2019.plac_seq_plot()`, `NOTT_2019.superenhancer_interactome`, `NOTT_2019.superenhancers()`

Examples

```
## Not run:
NOTT_2019.bigwig_metadata <- data.table::data.table(
    readxl::read_excel(
        file.path(
            "~/Desktop/Fine_Mapping/echolocatoR/annotations",
            "Nott_2019/Nott_2019.snEpigenomics.xlsx"
        )
    )
)
usethis::use_data(NOTT_2019.bigwig_metadata, overwrite = TRUE)

## End(Not run)
```

NOTT_2019.interactome *Brain cell type-specific enhancers, promoters, and interactomes*

Description

Originally from Nott et al. (2019). Specifically: *aay0793-Nott-Table-S5.xlsx*.

Usage

```
NOTT_2019.interactome
```

Format

An object of class `list` of length 12.

Source

https://science.sciencemag.org/content/366/6469/1134

See Also

Other NOTT_2019: NOTT_2019.bigwig_metadata, NOTT_2019.epigenomic_histograms(), NOTT_2019.get_epiger
NOTT_2019.get_interactions(), NOTT_2019.get_interactome(), NOTT_2019.get_promoter_celltypes(),
NOTT_2019.get_promoter_interactome_data(), NOTT_2019.get_regulatory_regions(), NOTT_2019.plac_seq_
NOTT_2019.superenhancer_interactome, NOTT_2019.superenhancers()

Examples

```
## Not run:
file <- file.path(
    "~/Desktop/Fine_Mapping/echolocatoR/annotations",
    "Nott_2019/aay0793-Nott-Table-S5.xlsx"
)
sheets <- readxl::excel_sheets(file)
enh_prom_sheets <- grep("enhancers|promoters", sheets, value = TRUE)
other_sheets <- grep("enhancers|promoters", sheets,
    value = TRUE,
    invert = TRUE
)
NOTT_2019.interactome <- lapply(other_sheets, function(s) {
    readxl::read_excel(file, sheet = s, skip = 2)
})
NOTT_2019.interactome <- append(
    NOTT_2019.interactome,
    lapply(enh_prom_sheets, function(s) {
        readxl::read_excel(file,
            sheet = s, skip = 2,
            col_names = c("chr", "start", "end")
        )
    })
)
names(NOTT_2019.interactome) <- c(other_sheets, enh_prom_sheets)
usethis::use_data(NOTT_2019.interactome, overwrite = TRUE)
```

```
## End(Not run)
```

NOTT_2019.superenhancer_interactome
Brain cell type-specific interactomes with superenhancers

Description

Originally from Nott et al. (2019). Specifically: *aay0793-Nott-Table-S6.xlsx*.

Usage

```
NOTT_2019.superenhancer_interactome
```

Format

An object of class `data.table` (inherits from `data.frame`) with 2954 rows and 29 columns.

Source

https://science.sciencemag.org/content/366/6469/1134

See Also

Other NOTT_2019: NOTT_2019.bigwig_metadata, NOTT_2019.epigenomic_histograms(), NOTT_2019.get_epiger
NOTT_2019.get_interactions(), NOTT_2019.get_interactome(), NOTT_2019.get_promoter_celltypes(),
NOTT_2019.get_promoter_interactome_data(), NOTT_2019.get_regulatory_regions(), NOTT_2019.interacto
NOTT_2019.plac_seq_plot(), NOTT_2019.superenhancers()

Examples

```
## Not run:
NOTT_2019.superenhancer_interactome <- data.table::data.table(
    readxl::read_excel(
        file.path(
            "~/Desktop/Fine_Mapping/echolocatoR",
            "annotations/Nott_2019/aay0793-Nott-Table-S6.xlsx"
        ),
        skip = 2
    )
)
usethis::use_data(NOTT_2019.superenhancer_interactome)

## End(Not run)
```

plot_dataset_overlap *Plot inter-study SNP overlap*

Description

Cross-tabulate SNP overlap (after applying filter) between each pair of studies.

Usage

```
plot_dataset_overlap(
  merged_DT,
  snp_filter = "!is.na(SNP)",
  filename = NA,
  formula_str = "~ SNP + Dataset",
  triangle = FALSE,
  proxies = NULL
)
```

See Also

Other summarise: CS_bin_plot(), CS_counts_plot(), get_CS_bins(), get_CS_counts(), get_SNPgroup_counts()
peak_overlap_plot(), results_report(), super_summary_plot()

plot_missense *Plot any missense variants*

Description

Plot any missense variants

Usage

```
plot_missense(
  merged_DT,
  snp_filter = "Support>0",
  label_yaxis = FALSE,
  x_label = "UCS missense\nmutations",
  show.legend = TRUE,
  show_numbers = FALSE,
  show_plot = TRUE
)
```

See Also

Other annotate: annotate_missense(), biomart_geneInfo(), biomart_snp_info(), biomart_snps_to_geneInfo()
haplor_epigenetics_enrichment(), haplor_epigenetics_summary(), haplor_haploreg(),
haplor_regulomedb(), snps_by_mutation_type()

Examples

```
## Not run:
merged_DT <- echodata::Nalls2019_merged
gg_missense <- plot_missense(
    merged_DT = merged_DT,
    snp_filter = "Support>0"
)
gg_missense <- plot_missense(
    merged_DT = merged_DT,
    snp_filter = "Consensus_SNP==TRUE"
)

## End(Not run)
```

`super_summary_plot` *Merge all summary plots into one super plot*

Description

Merge all summary plots into one super plot

Usage

```
super_summary_plot(
  merged_DT,
  snp_filter = "Consensus_SNP==TRUE",
  coloc_results = NULL,
  plot_missense = TRUE,
  show_plot = TRUE,
  save_plot = FALSE,
  height = 15,
  width = 13,
  dpi = 500
)
```

See Also

Other summarise: `CS_bin_plot()`, `CS_counts_plot()`, `get_CS_bins()`, `get_CS_counts()`, `get_SNPgroup_counts()`
`peak_overlap_plot()`, `plot_dataset_overlap()`, `results_report()`

Index