Package ‘STAARpipelineSummary’

November 11, 2023

Type Package

Title Summarization and Visualization of Analysis Results Generated by STAARpipeline

Version 0.9.7

Date 2023-11-11

Author Xihao Li [aut, cre], Zilin Li [aut, cre]

Maintainer Xihao Li <xihaoli@unc.edu>, Zilin Li <li@nenu.edu.cn>

Description An R package for summarizing analysis results generated by STAARpipeline.

License GPL-3

Copyright See COPYRIGHTS for details.

Imports Rcpp, STAAR, MultiSTAAR, STAARpipeline, SCANG, dplyr, SeqArray, SeqVarTools, GenomicFeatures, TxDb.Hsapiens.UCSC.hg38.knownGene, GMMAT, Matrix, methods, lattice

Encoding UTF-8

LazyData true

Depends R (>= 3.2.0)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

Annotate_Single_Variants . . . 2
Dynamic_Window_Results_Summary . . . 3
fit_nullmodel_genome_cond_spa . . . 5
Gene_Centric_Coding_Info . . . 8
Gene_Centric_Coding_Results_Summary . . . 9
Gene_Centric_Coding_Results_Summary_incl_ptv . . . 13
Gene_Centric_Noncoding_Info . . . 17
Gene_Centric_Noncoding_Results_Summary . . . 18
Individual_Analysis_Results_Summary . . . 23
Single_Variants_List_Analysis . . . 25
Sliding_Window_Info . . . 26
Sliding_Window_Results_Summary . . . 27

Index **30**

Annotate_Single_Variants

Functionally annotate a list of variants

Description

The Annotate_Single_Variants function takes in a list of variants to functionally annotate the input variants

Usage

```
Annotate_Single_Variants(
  agds_dir,
  single_variants_list,
  QC_label = "annotation/filter",
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Annotation_name
)
```

Arguments

agds_dir file directory of annotated GDS (aGDS) files for all chromosomes (1-22).

single_variants_list
a data frame containing the information of variants to be functionally annotated. The data frame must include 4 columns with the following names: "CHR" (chromosome number), "POS" (position), "REF" (reference allele), and "ALT" (alternative allele).

QC_label channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

Annotation_dir channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

Annotation_name_catalog
a data frame containing the annotation names and the corresponding channel names in the aGDS file.

Annotation_name
a vector of qualitative/quantitative annotation names user wants to extract.

Value

A data frame containing the basic information (chromosome, position, reference allele and alternative allele) and annotation scores for the input variants.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Dynamic_Window_Results_Summary

Summarize the results of dynamic window analysis generated by STAARpipeline *package and perform conditional analysis for (unconditionally) significant genetic regions by adjusting for a given list of known variants*

Description

The Dynamic_Window_Results_Summary function takes in the results of dynamic window analysis generated by STAARpipeline package, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the dynamic window analysis results and analyze the conditional association between a quantitative/dichotomous phenotype and the rare variants in the unconditional significant genetic regions.

Usage

```
Dynamic_Window_Results_Summary(
  agds_dir,
  jobs_num,
  input_path,
  output_path,
  dynamic_window_results_name,
  obj_nullmodel,
  known_loci = NULL,
  method_cond = c("optimal", "naive"),
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  variant_type = c("SNV", "Indel", "variant"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Use_annotation_weights = FALSE,
  Annotation_name = NULL,
  alpha = 0.05
)
```

Arguments

agds_dir
: a vector containing file directory of annotated GDS (aGDS) files for all chromosomes (1-22).

jobs_num
: a data frame containing the number of jobs for association analysis. The data frame must include a column with the name "scang_num"

input_path
: file directory of the input dynamic window analysis results.

output_path
: file directory of the output summary results.

dynamic_window_results_name
: file names of the input dynamic window analysis results.

obj_nullmodel
: an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

`known_loci` a data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`geno_missing_imputation`
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`variant_type` type of variant included in the conditional analysis. Choice includes "SNV", "Indel", or "variant" (default = "SNV").

`Annotation_dir` channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog`
a data frame containing the name and the corresponding channel name in the aGDS file.

`Use_annotation_weights`
use annotations as weights or not (default = FALSE).

`Annotation_name`
a vector of annotation names used in SCANG-STAAR (default = NULL).

`alpha` threshod to control the genome-wise (family-wise) error rate (default = 0.05).

Value

The function returns the following analysis results:

`SCANG_S_res_uncond_cond.Rdata` and `SCANG_S_res_uncond_cond.csv`: A matrix that summarized the unconditional and conditional results of the significant regions (GWER<alpha) detected by the SCANG-STAAR-S procedure (conditional results available if known_loci is not a NULL), including chromosome ("chr"), start position ("start_pos"), end position ("end_pos"), number of variants ("SNV_nos"), family-wise/genome-wide error rate (GWER), unconditional STAAR-S p-value ("STAAR_S"), conditional STAAR-S p-value ("STAAR_S_cond"), conditional ACAT-V p-value ("ACAT_V_cond"), conditional Burden p-value ("Burden_cond"), conditional SKAT p-value ("SKAT_cond"), and conditional STAAR-O p-value ("STAAR_O_cond").

`SCANG_B_res_uncond_cond.Rdata` and `SCANG_B_res_uncond_cond.csv`: A matrix that summarized the unconditional and conditional results of the significant regions detected by the SCANG-STAAR-B procedure (conditional results available if known_loci is not a NULL). Details see SCANG-STAAR-S.

`SCANG_O_res_uncond_cond.Rdata` and `SCANG_O_res_uncond_cond.csv`: A matrix that summarized the unconditional and conditional results of the significant regions detected by the SCANG-STAAR-O procedure (conditional results available if known_loci is not a NULL). Details see SCANG-STAAR-S.

`results_dynamic_window.Rdata`: A Rdata file that summarized the significant regions detected by SCANG-STAAR procedure.

`SCANG_S_top1.Rdata` and `SCANG_S_top1.csv`: A matrix that summarized the top 1 unconditional region detected by SCANG-STAAR-S, including the STAAR-S p-value ("STAAR_S"), chromosome ("chr"), start position ("start_pos"), end position ("end_pos"), family-wise/genome-wide error rate (GWER) and the number of variants ("SNV_nos").

SCANG_B_top1.Rdata and SCANG_B_top1.csv: A matrix that summarized the top 1 unconditional region detected by SCANG-STAAR-B. Details see SCANG-STAAR-S.

SCANG_O_top1.Rdata and SCANG_O_top1.csv: A matrix that summarized the top 1 unconditional region detected by SCANG-STAAR-O. Details see SCANG-STAAR-S.

SCANG_S_res.Rdata and SCANG_S_res.csv: A matrix that summarized the significant regions (GWER<alpha) detected by SCANG-STAAR-S, including the negative log transformation of STAAR-S p-value ("-logp"), chromosome ("chr"), start position ("start_pos"), end position ("end_pos"), family-wise/genome-wide error rate (GWER) and the number of variants ("SNV_num").

SCANG_B_res.Rdata and SCANG_B_res.csv: A matrix that summarized the significant regions detected by SCANG-STAAR-B. Details see SCANG-STAAR-S.

SCANG_O_res.Rdata and SCANG_O_res.csv: A matrix that summarized the significant regions detected by SCANG-STAAR-O. Details see SCANG-STAAR-S.

SCANG_S_res_cond.Rdata and SCANG_S_res_cond.csv: A matrix that summarized the conditional p-values of the significant regions (GWER<alpha) detected by SCANG-STAAR-S, including chromosome ("chr"), start position ("Start Loc"), end position ("End Loc"), the number of variants ("#SNV"), annotation-weighted ACAT-V, Burden and SKAT conditional p-values, and STAAR conditional p-values of the regions with GWER smaller than the threshold alpha (available if known_loci is not a NULL).

SCANG_B_res_cond.Rdata and SCANG_B_res_cond.csv: A matrix that summarized the conditional p-values of the significant regions (GWER<alpha) detected by SCANG-STAAR-B (available if known_loci is not a NULL), Details see SCANG-STAAR-S.

SCANG_O_res_cond.Rdata and SCANG_O_res_cond.csv: A matrix that summarized the conditional p-values of the significant regions (GWER<alpha) detected by SCANG-STAAR-O (available if known_loci is not a NULL), Details see SCANG-STAAR-S.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

fit_nullmodel_genome_cond_spa

Fitting conditional generalized linear mixed models with known relationship matrices for conditional analysis in imbalanced case-control setting.

Description

The fit_nullmodel_genome_cond_spa function fit regression models for conditional analysis in imbalanced case-control setting, which provides the preliminary step for subsequent conditional variant-set tests in conditional analysis. Each chromosome has a separate null model for conditional analysis. See fit_nullmodel for more details.

Usage

```
fit_nullmodel_genome_cond_spa(
  fixed,
  data = parent.frame(),
  kins,
  use_sparse = TRUE,
  use_SPA = TRUE,
  agds_dir,
  known_loci,
  geno_missing_imputation = c("mean", "minor"),
  MAC_cutoff = 20,
  output_path,
  cond_null_model_name = NULL,
  phenotype_id,
  phenotype,
  kins_cutoff = 0.022,
  id,
  random.slope = NULL,
  groups = NULL,
  family = binomial(link = "logit"),
  method = "REML",
  method.optim = "AI",
  maxiter = 500,
  tol = 1e-05,
  taumin = 1e-05,
  taumax = 1e+05,
  tauregion = 10,
  verbose = FALSE,
  ...
)
```

Arguments

`fixed` an object of class `formula` (or one that can be coerced to that class): a symbolic description of the fixed effects model to be fitted. For multiple phenotype analysis, `formula` recognized by `lm`, such as `cbind(y1,y2,y3) ~ x1 + x2`, can be used in `fixed` as fixed effects.

`data` a data frame or list (or object coercible by `as.data.frame` to a data frame) containing the variables in the model.

`kins` a known positive semi-definite relationship matrix (e.g. kinship matrix in genetic association studies) or a list of known positive semi-definite relationship matrices. The rownames and colnames of these matrices must at least include all samples as specified in the `id` column of the data frame `data`. If `kins` is NULL, `fit_nullmodel` will switch to the generalized linear model with no random effects.

`use_sparse` a logical switch of whether the provided dense `kins` matrix should be transformed to a sparse matrix (default = TRUE).

`use_SPA` a logical switch determines if the null model fitting occurs in an imbalanced case-control setting (default = TRUE).

`agds_dir` file directory of annotated GDS (aGDS) files for all chromosomes (1-22)

`known_loci` the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`geno_missing_imputation`
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`MAC_cutoff` the cutoff of the minimum minor allele count of known variants adjusted in conditional analysis (default = 20).

`output_path` the directory for the output files.

`cond_null_model_name` the file name of conditional null models (default = NULL).

`phenotype_id` id of samples.

`phenotype` outcome in regression.

`kins_cutoff` the cutoff value for clustering samples to make the output matrix sparse block-diagonal (default = 0.022).

`id` a column in the data frame `data`, indicating the id of samples. When there are duplicates in `id`, the data is assumed to be longitudinal with repeated measures.

`random.slope` an optional column indicating the random slope for time effect used in a mixed effects model for longitudinal data. It must be included in the names of `data`. There must be duplicates in `id` and `method.optim` must be "AI" (default = NULL).

`groups` an optional categorical variable indicating the groups used in a heteroscedastic linear mixed model (allowing residual variances in different groups to be different). This variable must be included in the names of `data`, and `family` must be "gaussian" and `method.optim` must be "AI" (default = NULL).

`family` a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See `family` for details of family functions).

`method` method of fitting the generalized linear mixed model. Either "REML" or "ML" (default = "REML").

`method.optim` optimization method of fitting the generalized linear mixed model. Either "AI", "Brent" or "Nelder-Mead" (default = "AI").

`maxiter` a positive integer specifying the maximum number of iterations when fitting the generalized linear mixed model (default = 500).

`tol` a positive number specifying tolerance, the difference threshold for parameter estimates below which iterations should be stopped (default = 1e-5).

`taumin` the lower bound of search space for the variance component parameter τ (default = 1e-5), used when `method.optim` = "Brent". See Details.

`taumax` the upper bound of search space for the variance component parameter τ (default = 1e5), used when `method.optim` = "Brent". See Details.

`tauregion` the number of search intervals for the REML or ML estimate of the variance component parameter τ (default = 10), used when `method.optim` = "Brent". See Details.

`verbose` a logical switch for printing detailed information (parameter estimates in each iteration) for testing and debugging purpose (default = FALSE).

`...` additional arguments that could be passed to `glm`.

Value

The function returns objects of the null models fit from `fit_nullmodel` and whether the `kins` matrix is sparse when fitting the null model, each chromosome has one output. See `fit_nullmodel` for more details.

References

Chen, H., et al. (2016). Control for population structure and relatedness for binary traits in genetic association studies via logistic mixed models. *The American Journal of Human Genetics*, *98*(4), 653-666. (pub)

Chen, H., et al. (2019). Efficient variant set mixed model association tests for continuous and binary traits in large-scale whole-genome sequencing studies. *The American Journal of Human Genetics*, *104*(2), 260-274. (pub)

Chen, H. (2021). GMMAT: Generalized linear Mixed Model Association Tests Version 1.3.2. (web)

Gene_Centric_Coding_Info

Functionally annotate rare variants in a coding mask

Description

The Gene_Centric_Coding_Info function takes in a coding mask of a gene to functionally annotate the rare variants in the mask.

Usage

```
Gene_Centric_Coding_Info(
  category = c("plof", "plof_ds", "missense", "disruptive_missense", "synonymous", "ptv",
    "ptv_ds"),
  chr,
  genofile,
  obj_nullmodel,
  gene_name,
  known_loci = NULL,
  rare_maf_cutoff = 0.01,
  method_cond = c("optimal", "naive"),
  QC_label = "annotation/filter",
  variant_type = c("SNV", "Indel", "variant"),
  geno_missing_imputation = c("mean", "minor"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Annotation_name
)
```

Arguments

category the coding functional category of rare variants to be functionally annotated. Choices include plof, plof_ds, missense, disruptive_missense, synonymous, ptv, ptv_ds (default = plof).

chr chromosome.

genofile an object of opened annotated GDS (aGDS) file.

obj_nullmodel an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

`gene_name` name of the gene to be annotated.

`known_loci` the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`rare_maf_cutoff` the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`QC_label` channel name of the QC label in the GDS/aGDS file.

`variant_type` type of variant included in the conditional analysis. Choices include "SNV", "Indel", or "variant" (default = "SNV").

`geno_missing_imputation` method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`Annotation_dir` channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog` a data frame containing the name and the corresponding channel name in the aGDS file.

`Annotation_name` a vector of qualitative/quantitative annotation names user wants to extract.

Value

A data frame containing the basic information (chromosome, position, reference allele and alternative allele), unconditional and conditional the score test p-values (not provided for imbalanced case-control setting), and annotation scores for the rare variants of the input coding mask.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

`Gene_Centric_Coding_Results_Summary`

Summarize gene-centric coding analysis results generated by `STAARpipeline` *package and perform conditional analysis for (unconditionally) significant coding masks by adjusting for a given list of known variants*

Description

The `Gene_Centric_Coding_Results_Summary` function takes in the objects of gene-centric coding analysis results generated by `STAARpipeline` package, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the gene-centric coding analysis results and analyze the conditional association between a quantitative/dichotomous phenotype (including imbalanced case-control design) and the rare variants in the unconditional significant coding masks.

Usage

```
Gene_Centric_Coding_Results_Summary(
  agds_dir,
  gene_centric_coding_jobs_num,
  input_path,
  output_path,
  gene_centric_results_name,
  obj_nullmodel,
  known_loci = NULL,
  cMAC_cutoff = 0,
  method_cond = c("optimal", "naive"),
  rare_maf_cutoff = 0.01,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  variant_type = c("SNV", "Indel", "variant"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Use_annotation_weights = FALSE,
  Annotation_name = NULL,
  alpha = 2.5e-06,
  manhattan_plot = FALSE,
  QQ_plot = FALSE,
  cond_null_model_name = NULL,
  cond_null_model_dir = NULL,
  SPA_p_filter = FALSE,
  p_filter_cutoff = 0.05
)
```

Arguments

`agds_dir` file directory of annotated GDS (aGDS) files for all chromosomes (1-22)

`gene_centric_coding_jobs_num`
the number of gene-centric coding analysis results generated by `STAARpipeline` package.

`input_path` the directory of gene-centric coding analysis results that generated by `STAARpipeline` package.

`output_path` the directory for the output files.

`gene_centric_results_name`
file name of gene-centric coding analysis results generated by `STAARpipeline` package.

`obj_nullmodel` an object from fitting the null model, which is either the output from `fit_nullmodel` function in the `STAARpipeline` package, or the output from `fitNullModel` function in the `GENESIS` package and transformed using the `genesis2staar_nullmodel` function in the `STAARpipeline` package.

`known_loci` the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`cMAC_cutoff` the cutoff of the minimum number of the cumulative minor allele of variants in the masks when summarizing the results (default = 0).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`rare_maf_cutoff`
the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`geno_missing_imputation`
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`variant_type` type of variant included in the analysis. Choices include "SNV", "Indel", or "variant" (default = "SNV").

`Annotation_dir` channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog`
a data frame containing the name and the corresponding channel name in the aGDS file.

`Use_annotation_weights`
use annotations as weights or not (default = FALSE).

`Annotation_name`
a vector of annotation names used in STAAR (default = NULL).

`alpha` p-value threshold of significant results (default = 2.5E-06).

`manhattan_plot` output manhattan plot or not (default = FALSE).

`QQ_plot` output Q-Q plot or not (default = FALSE).

`cond_null_model_name`
the null model name for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`cond_null_model_dir`
the directory of storing the null model for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`SPA_p_filter` logical: are only the variants with a normal approximation based p-value smaller than a pre-specified threshold use the SPA method to recalculate the p-value, only used for imbalanced case-control setting (default = FALSE).

`p_filter_cutoff`
threshold for the p-value recalculation using the SPA method, only used for imbalanced case-control setting (default = 0.05).

Value

The function returns the following analysis results:

`coding_sig.csv`: a matrix that summarizes the unconditional significant coding masks detected by STAAR-O or STAAR-B in imbalanced case-control setting (STAAR-O/-B pvalue smaller than the threshold alpha), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and unconditional p-values of set-based tests SKAT ("SKAT(1,25)"), Burden ("Burden(1,1)"), ACAT-V ("ACAT-V(1,25)") and STAAR-O ("STAAR-O") or unconditional p-values of set-based tests Burden ("Burden(1,1)") and STAAR-B ("STAAR-B") for imbalanced case-control setting.

coding_sig_cond.csv: a matrix that summarized the conditional analysis results of unconditional significant coding masks detected by STAAR-O or STAAR-B in imbalanced case-control setting (available if known_loci is not a NULL), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and conditional p-values of set-based tests SKAT ("SKAT(1,25)"), Burden ("Burden(1,1)"), ACAT-V ("ACAT-V(1,25)") and STAAR-O ("STAAR-O") or conditional p-values of set-based tests Burden ("Burden(1,1)") and STAAR-B ("STAAR-B") for imbalanced case-control setting.

results_plof_genome.Rdata: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the putative loss of function variants (plof) for all protein-coding genes across the genome.

plof_sig.csv: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof masks.

plof_sig_cond.csv: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof masks (available if known_loci is not a NULL).

results_plof_ds_genome.Rdata: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the putative loss of function variants and disruptive missense variants (plof_ds) for all protein-coding genes across the genome.

plof_ds_sig.csv: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof_ds masks.

plof_ds_sig_cond.csv: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof_ds masks (available if known_loci is not a NULL).

results_disruptive_missense_genome.Rdata: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the disruptive missense variants (disruptive_missense) for all protein-coding genes across the genome.

disruptive_missense_sig.csv: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant disruptive_missense masks.

disruptive_missense_sig_cond.csv: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant disruptive_missense masks (available if known_loci is not a NULL).

results_missense_genome.Rdata: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the missense variants (missense) for all protein-coding genes across the genome.

missense_sig.csv: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant missense masks.

missense_sig_cond.csv: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant missense masks (available if known_loci is not a NULL).

results_synonymous_genome.Rdata: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the synonymous variants (synonymous) for all protein-coding genes across the genome.

synonymous_sig.csv: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant synonymous masks.

synonymous_sig_cond.csv: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant synonymous masks (available if known_loci is not a NULL).

manhattan plot (optional) and Q-Q plot (optional) of the gene-centric coding analysis results.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Gene_Centric_Coding_Results_Summary_incl_ptv

Summarize gene-centric coding analysis results generated by STAARpipeline *package and perform conditional analysis for (unconditionally) significant coding masks (including masks ptv and ptv_ds) by adjusting for a given list of known variants*

Description

The Gene_Centric_Coding_Results_Summary_incl_ptv function takes in the objects of gene-centric coding analysis results generated by STAARpipeline package, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the gene-centric coding analysis results and analyze the conditional association between a quantitative/dichotomous phenotype (including imbalanced case-control design) and the rare variants in the unconditional significant coding masks.

Usage

```
Gene_Centric_Coding_Results_Summary_incl_ptv(
  agds_dir,
  gene_centric_coding_jobs_num,
  input_path,
  output_path,
  gene_centric_results_name,
  obj_nullmodel,
  known_loci = NULL,
  cMAC_cutoff = 0,
  method_cond = c("optimal", "naive"),
  rare_maf_cutoff = 0.01,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  variant_type = c("SNV", "Indel", "variant"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Use_annotation_weights = FALSE,
  Annotation_name = NULL,
  alpha = 2.5e-06,
  manhattan_plot = FALSE,
  QQ_plot = FALSE,
  cond_null_model_name = NULL,
  cond_null_model_dir = NULL,
  SPA_p_filter = FALSE,
  p_filter_cutoff = 0.05
)
```

Arguments

`agds_dir` file directory of annotated GDS (aGDS) files for all chromosomes (1-22)

`gene_centric_coding_jobs_num`
the number of gene-centric coding analysis results generated by `STAARpipeline` package.

`input_path` the directory of gene-centric coding analysis results that generated by `STAARpipeline` package.

`output_path` the directory for the output files.

`gene_centric_results_name`
file name of gene-centric coding analysis results generated by `STAARpipeline` package.

`obj_nullmodel` an object from fitting the null model, which is either the output from `fit_nullmodel` function in the `STAARpipeline` package, or the output from `fitNullModel` function in the `GENESIS` package and transformed using the `genesis2staar_nullmodel` function in the `STAARpipeline` package.

`known_loci` the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`cMAC_cutoff` the cutoff of the minimum number of the cumulative minor allele of variants in the masks when summarizing the results (default = 0).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`rare_maf_cutoff`
the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`geno_missing_imputation`
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`variant_type` type of variant included in the analysis. Choices include "SNV", "Indel", or "variant" (default = "SNV").

`Annotation_dir` channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog`
a data frame containing the name and the corresponding channel name in the aGDS file.

`Use_annotation_weights`
use annotations as weights or not (default = FALSE).

`Annotation_name`
a vector of annotation names used in STAAR (default = NULL).

`alpha` p-value threshold of significant results (default = 2.5E-06).

`manhattan_plot` output manhattan plot or not (default = FALSE).

`QQ_plot` output Q-Q plot or not (default = FALSE).

`cond_null_model_name`
the null model name for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`cond_null_model_dir`
the directory of storing the null model for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`SPA_p_filter` logical: are only the variants with a normal approximation based p-value smaller than a pre-specified threshold use the SPA method to recalculate the p-value, only used for imbalanced case-control setting (default = FALSE).

`p_filter_cutoff`
threshold for the p-value recalculation using the SPA method, only used for imbalanced case-control setting (default = 0.05).

Value

The function returns the following analysis results:

`coding_sig.csv`: a matrix that summarizes the unconditional significant coding masks detected by STAAR-O or STAAR-B in imbalanced case-control setting (STAAR-O/-B pvalue smaller than the threshold alpha), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and unconditional p-values of set-based tests SKAT ("SKAT(1,25)"), Burden ("Burden(1,1)"), ACAT-V ("ACAT-V(1,25)") and STAAR-O ("STAAR-O") or unconditional p-values of set-based tests Burden ("Burden(1,1)") and STAAR-B ("STAAR-B") for imbalanced case-control setting.

`coding_sig_cond.csv`: a matrix that summarized the conditional analysis results of unconditional significant coding masks detected by STAAR-O or STAAR-B in imbalanced case-control setting (available if known_loci is not a NULL), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and conditional p-values of set-based tests SKAT ("SKAT(1,25)"), Burden ("Burden(1,1)"), ACAT-V ("ACAT-V(1,25)") and STAAR-O ("STAAR-O") or conditional p-values of set-based tests Burden ("Burden(1,1)") and STAAR-B ("STAAR-B") for imbalanced case-control setting.

`results_plof_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the putative loss of function variants (plof) for all protein-coding genes across the genome.

`plof_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof masks.

`plof_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof masks (available if known_loci is not a NULL).

`results_plof_ds_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the putative loss of function variants and disruptive missense variants (plof_ds) for all protein-coding genes across the genome.

`plof_ds_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof_ds masks.

`plof_ds_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant plof_ds masks (available if known_loci is not a NULL).

`results_ptv_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the protein-truncating variants (ptv) for all protein-coding genes across the genome.

`ptv_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ptv masks.

`ptv_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ptv masks (available if known_loci is not a NULL).

`results_ptv_ds_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the protein-truncating variants and disruptive missense variants (ptv_ds) for all protein-coding genes across the genome.

`ptv_ds_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ptv_ds masks.

`ptv_ds_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ptv_ds masks (available if known_loci is not a NULL).

`results_disruptive_missense_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the disruptive missense variants (disruptive_missense) for all protein-coding genes across the genome.

`disruptive_missense_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant disruptive_missense masks.

`disruptive_missense_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant disruptive_missense masks (available if known_loci is not a NULL).

`results_missense_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the missense variants (missense) for all protein-coding genes across the genome.

`missense_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant missense masks.

`missense_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant missense masks (available if known_loci is not a NULL).

`results_synonymous_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the coding mask defined by the synonymous variants (synonymous) for all protein-coding genes across the genome.

`synonymous_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant synonymous masks.

`synonymous_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant synonymous masks (available if known_loci is not a NULL).

manhattan plot (optional) and Q-Q plot (optional) of the gene-centric coding analysis results.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Gene_Centric_Noncoding_Info *Functionally annotate rare variants in a noncoding mask*

Description

The Gene_Centric_Noncoding_Info function takes in a noncoding mask of a gene to functionally annotate the rare variants in the mask.

Usage

```
Gene_Centric_Noncoding_Info(
  category = c("downstream", "upstream", "UTR", "promoter_CAGE", "promoter_DHS",
    "enhancer_CAGE", "enhancer_DHS", "ncRNA"),
  chr,
  genofile,
  obj_nullmodel,
  gene_name,
  known_loci = NULL,
  rare_maf_cutoff = 0.01,
  method_cond = c("optimal", "naive"),
  QC_label = "annotation/filter",
  variant_type = c("SNV", "Indel", "variant"),
  geno_missing_imputation = c("mean", "minor"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Annotation_name
)
```

Arguments

category
: the noncoding functional category to be functionally annotated. Choices include downstream, upstream, UTR, promoter_CAGE, promoter_DHS, enhancer_CAGE, enhancer_DHS, ncRNA (default = downstream).

chr
: chromosome.

genofile
: an object of opened annotated GDS (aGDS) file.

obj_nullmodel
: an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

gene_name
: name of the gene to be annotated.

known_loci
: the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

rare_maf_cutoff
: the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`variant_type` type of variant included in the conditional analysis. Choices include "SNV", "Indel", or "variant" (default = "SNV").

`geno_missing_imputation` method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`Annotation_dir` channel name of the annotations in the aGDS file (default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog` a data frame containing the name and the corresponding channel name in the aGDS file.

`Annotation_name` a vector of qualitative/quantitative annotation names user wants to extract.

Value

a data frame containing the basic information (chromosome, position, reference allele and alternative allele), unconditional and conditional the score test p-values (not provided for imbalanced case-control setting), and annotation scores for the rare variants of the input noncoding mask.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

`Gene_Centric_Noncoding_Results_Summary`

Summarize gene-centric noncoding analysis results generated by `STAARpipeline` *package*

Description

The `Gene_Centric_Noncoding_Results_Summary` function takes in the objects of gene-centric noncoding analysis results generated by `STAARpipeline` package, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the gene-centric noncoding analysis results and analyze the conditional association between a quantitative/dichotomous phenotype (including imbalanced case-control design) and the rare variants in the unconditional significant noncoding masks.

Usage

```
Gene_Centric_Noncoding_Results_Summary(
  agds_dir,
  gene_centric_noncoding_jobs_num,
  input_path,
  output_path,
  gene_centric_results_name,
  ncRNA_jobs_num,
  ncRNA_input_path,
  ncRNA_output_path,
  ncRNA_results_name,
  obj_nullmodel,
  known_loci = NULL,
  cMAC_cutoff = 0,
  method_cond = c("optimal", "naive"),
  rare_maf_cutoff = 0.01,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  variant_type = c("SNV", "Indel", "variant"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Use_annotation_weights = FALSE,
  Annotation_name = NULL,
  alpha = 2.5e-06,
  alpha_ncRNA = 2.5e-06,
  ncRNA_pos = NULL,
  manhattan_plot = FALSE,
  QQ_plot = FALSE,
  cond_null_model_name = NULL,
  cond_null_model_dir = NULL,
  SPA_p_filter = FALSE,
  p_filter_cutoff = 0.05
)
```

Arguments

`agds_dir` a data farme containing directory of GDS/aGDS files.

`gene_centric_noncoding_jobs_num`
the number of results for gene-centric noncoding analysis of protein-coding genes generated by `STAARpipeline` package.

`input_path` the directory of gene-centric noncoding analysis results for protein-coding genes that generated by `STAARpipeline` package.

`output_path` the directory for the output files of the summary of gene-centric noncoding analysis results for protein-coding genes.

`gene_centric_results_name`
the file name of gene-centric noncoding analysis results for protein-coding genes generated by `STAARpipeline` package.

`ncRNA_jobs_num` the number of results for gene-centric noncoding analysis of ncRNA genes generated by `STAARpipeline` package..

`ncRNA_input_path`
the directory of gene-centric noncoding analysis results for ncRNA genes that generated by `STAARpipeline` package.

`ncRNA_output_path`
the directory for the output files of the summary of gene-centric noncoding analysis results for ncRNA genes.

ncRNA_results_name
file name of gene-centric noncoding analysis results for ncRNA genes that generated by STAARpipeline package.

obj_nullmodel
an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

known_loci
the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

cMAC_cutoff
the cutoff of the minimum number of the cumulative minor allele of variants in the masks when summarizing the results (default = 0).

method_cond
a character value indicating the method for conditional analysis. optimal refers to regressing residuals from the null model on known_loci as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; naive refers to regressing residuals from the null model on known_loci and taking the residuals (default = optimal).

rare_maf_cutoff
the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

QC_label
channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

geno_missing_imputation
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

variant_type
type of variant included in the analysis. Choices include "SNV", "Indel", or "variant" (default = "SNV").

Annotation_dir
channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

Annotation_name_catalog
a data frame containing the name and the corresponding channel name in the aGDS file.

Use_annotation_weights
use annotations as weights or not (default = FALSE).

Annotation_name
a vector of annotation names used in STAAR (default = NULL).

alpha
p-value threshold of significant results of protein coding genes (default = 2.5E-06).

alpha_ncRNA
p-value threshold of significant results of ncRNA genes (default = 2.5E-06).

ncRNA_pos
positions of ncRNA genes, required for generating the Manhattan plot and Q-Q plot of the results of ncRNA genes (default=NULL).

manhattan_plot
output manhattan plot or not (default = FALSE).

QQ_plot
output Q-Q plot or not (default = FALSE).

cond_null_model_name
the null model name for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

cond_null_model_dir
the directory of storing the null model for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`SPA_p_filter` logical: are only the variants with a normal approximation based p-value smaller than a pre-specified threshold use the SPA method to recalculate the p-value, only used for imbalanced case-control setting (default = FALSE).

`p_filter_cutoff` threshold for the p-value recalculation using the SPA method, only used for imbalanced case-control setting (default = 0.05).

Value

The function returns the following analysis results:

`noncoding_sig.csv`: a matrix that summarized the unconditional significant region detected by STAAR-O or STAAR-B in imbalanced case-control setting (STAAR-O/-B pvalue smaller than the threshold alpha), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting).

`noncoding_sig_cond.csv`: a matrix that summarized the conditional analysis results of the unconditional significant region detected by STAAR-O or STAAR-B in imbalanced case-control setting (available if known_loci is not a NULL), including gene name ("Gene name"), chromosome ("chr"), coding functional category ("Category"), number of variants ("#SNV"), and the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting).

`results_UTR_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by UTR variants (UTR) for all protein-coding genes across the genome.

`UTR_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant UTR masks.

`UTR_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant UTR masks (available if known_loci is not a NULL).

`results_upstream_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by upstream variants (upstream) for all protein-coding genes across the genome.

`upstream_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant upstream masks.

`upstream_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant upstream masks (available if known_loci is not a NULL).

`results_downstream_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by downstream variants (downstream) for all protein-coding genes across the genome.

`downstream_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant downstream masks.

`downstream_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant downstream masks (available if known_loci is not a NULL).

`results_promoter_CAGE_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by variants overlaid with CAGE sites in the promoter (promoter_CAGE) for all protein-coding genes across the genome.

`promoter_CAGE_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant promoter_CAGE masks.

`promoter_CAGE_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant promoter_CAGE masks (available if known_loci is not a NULL).

`results_promoter_DHS_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by variants overlaid with DHS sites in the promoter (promoter_DHS) for all protein-coding genes across the genome.

`promoter_DHS_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant promoter_DHS masks.

`promoter_DHS_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant promoter_DHS masks (available if known_loci is not a NULL).

`results_enhancer_CAGE_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by variants overlaid with CAGE sites in the enhancer (enhancer_CAGE) for all protein-coding genes across the genome.

`enhancer_CAGE_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant enhancer_CAGE masks.

`enhancer_CAGE_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant enhancer_CAGE masks (available if known_loci is not a NULL).

`results_enhancer_DHS_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by variants overlaid with DHS sites in the enhancer (enhancer_DHS) for all protein-coding genes across the genome.

`enhancer_DHS_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant enhancer_DHS masks.

`enhancer_DHS_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant enhancer_DHS masks (available if known_loci is not a NULL).

`results_ncRNA_genome`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the noncoding masks defined by exonic and splicing ncRNA variants (ncRNA) for all ncRNA genes across the genome.

`ncRNA_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ncRNA masks.

`ncRNA_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the unconditional significant ncRNA masks (available if known_loci is not a NULL).

manhattan plot (optional) and Q-Q plot (optional) of the gene-centric noncoding analysis results.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Individual_Analysis_Results_Summary *Summarize individual-variant analysis results generated by* STAARpipeline *package*

Description

The Individual_Analysis_Results_Summary function takes in the objects of individual analysis results generated by STAARpipeline package, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the individual analysis results and analyze the conditional association between a quantitative/dichotomous phenotype and the unconditional significant single variants.

Usage

```
Individual_Analysis_Results_Summary(
  agds_dir,
  jobs_num,
  input_path,
  output_path,
  individual_results_name,
  obj_nullmodel,
  known_loci = NULL,
  method_cond = c("optimal", "naive"),
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  alpha = 5e-09,
  manhattan_plot = FALSE,
  QQ_plot = FALSE,
  SPA_p_filter = FALSE,
  p_filter_cutoff = 0.05,
  cond_null_model_name = NULL,
  cond_null_model_dir = NULL
)
```

Arguments

agds_dir
: a data farme containing directory of GDS/aGDS files.

jobs_num
: a data frame containing the number of analysis results, including the number of individual analysis results, the number of sliding window analysis results, and the number of dynamic window analysis results.

input_path
: the directory of individual analysis results that generated by STAARpipeline package.

output_path
: the directory for the output files.

individual_results_name
: the file name of individual analysis results generated by STAARpipeline package.

obj_nullmodel
: an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel

function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

known_loci
: the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

method_cond
: a character value indicating the method for conditional analysis. optimal refers to regressing residuals from the null model on known_loci as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; naive refers to regressing residuals from the null model on known_loci and taking the residuals (default = optimal).

QC_label
: channel name of the QC label in the GDS/aGDS file.

geno_missing_imputation
: method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

alpha
: p-value threshold of significant results (default = 5E-09).

manhattan_plot
: output manhattan plot or not (default = FALSE).

QQ_plot
: output Q-Q plot or not (default = FALSE).

SPA_p_filter
: logical: are only the variants with a score-test-based p-value smaller than a pre-specified threshold use the SPA method to recalculate the p-value, only used for imbalanced case-control setting (default = FALSE).

p_filter_cutoff
: threshold for the p-value recalculation using the SPA method, only used for imbalanced case-control setting (default = 0.05)

cond_null_model_name
: the null model name for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

cond_null_model_dir
: the directory of storing the null model for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

Value

The function returns the following analysis results:

results_individual_analysis_genome.Rdata: a matrix contains the score test p-value and effect size estimation of each variant across the genome.

results_individual_analysis_sig.Rdata and results_individual_analysis_sig.csv: a matrix contains the score test p-values and effect size estimations of significant results (p-value < alpha).

results_sig_cond.Rdata and results_sig_cond.csv: a matrix contains the conditional score test p-values for each significant variant (available if known_loci is not a NULL).

manhattan plot (optional) and Q-Q plot (optional) of the individual analysis results.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Single_Variants_List_Analysis

Calculate individual-variant p-values of a list of variants

Description

The Single_Variants_List_Analysis function takes in a list of variants to calculate the p-values and effect sizes of the input variants (effect size estimations are not provided for imbalanced case-control setting), only support for null model fitting using sparse GRM.

Usage

```
Single_Variants_List_Analysis(
  agds_dir,
  single_variants_list,
  obj_nullmodel,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  p_filter_cutoff = 0.05,
  tol = .Machine$double.eps^0.25,
  max_iter = 1000
)
```

Arguments

agds_dir file directory of annotated GDS (aGDS) files for all chromosomes (1-22).

single_variants_list
name a data frame containing the information of variants to be functionally annotated. The data frame must include 4 columns with the following names: "CHR" (chromosome number), "POS" (position), "REF" (reference allele), and "ALT" (alternative allele).

obj_nullmodel an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

QC_label channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

geno_missing_imputation
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

p_filter_cutoff
threshold for the p-value recalculation using the SPA method (default = 0.05)

tol a positive number specifying tolerance, the difference threshold for parameter estimates in saddlepoint apporximation algorithm below which iterations should be stopped (default = ".Machine$double.eps^0.25").

max_iter a positive integers pecifying the maximum number of iterations for applying the saddlepoint approximation algorithm (default = "1000").

Value

a data frame containing the basic information (chromosome, position, reference allele and alternative allele) the score test p-values, and the effect sizes for the input variants.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Sliding_Window_Info *Functionally annotate rare variants in a genetic region*

Description

The Sliding_Window_Info function takes in the location of a genetic region to functionally annotate the rare variants in the region.

Usage

```
Sliding_Window_Info(
  chr,
  genofile,
  obj_nullmodel,
  start_loc,
  end_loc,
  known_loci = NULL,
  rare_maf_cutoff = 0.01,
  method_cond = c("optimal", "naive"),
  QC_label = "annotation/filter",
  variant_type = c("SNV", "Indel", "variant"),
  geno_missing_imputation = c("mean", "minor"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Annotation_name
)
```

Arguments

chr
: chromosome.

genofile
: an object of opened annotated GDS (aGDS) file.

obj_nullmodel
: an object from fitting the null model, which is either the output from fit_nullmodel function in the STAARpipeline package, or the output from fitNullModel function in the GENESIS package and transformed using the genesis2staar_nullmodel function in the STAARpipeline package.

start_loc
: starting location (position) of the genetic region to be annotated.

end_loc
: ending location (position) of the genetic region to be annotated.

known_loci
: the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

rare_maf_cutoff
: the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`variant_type` variants include in the conditional analysis. Choices include "variant", "SNV", or "Indel" (default = "SNV").

`geno_missing_imputation`
method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`Annotation_dir` channel name of the annotations in the aGDS file
(default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog`
a data frame containing the name and the corresponding channel name in the aGDS file.

`Annotation_name`
a vector of qualitative/quantitative annotation names user wants to extract.

Value

A data frame containing the basic information (chromosome, position, reference allele and alternative allele), unconditional and conditional the score test p-values, and annotation scores for the input variants.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

`Sliding_Window_Results_Summary` *Summarize the sliding window analysis results generated by* `STAARpipeline` *package*

Description

The `Sliding_Window_Results_Summary` function takes in the results of sliding window analysis, the object from fitting the null model, and the set of known variants to be adjusted for in conditional analysis to summarize the sliding window analysis results and analyze the conditional association between a quantitative/dichotomous phenotype (including imbalanced case-control setting) and the rare variants in the unconditional significant genetic region.

Usage

```
Sliding_Window_Results_Summary(
  agds_dir,
  jobs_num,
  input_path,
  output_path,
  sliding_window_results_name,
  obj_nullmodel,
  known_loci = NULL,
  cMAC_cutoff = 0,
  method_cond = c("optimal", "naive"),
  rare_maf_cutoff = 0.01,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor"),
  variant_type = c("SNV", "Indel", "variant"),
  Annotation_dir = "annotation/info/FunctionalAnnotation",
  Annotation_name_catalog,
  Use_annotation_weights = FALSE,
  Annotation_name = NULL,
  alpha = 0.05,
  manhattan_plot = FALSE,
  QQ_plot = FALSE,
  cond_null_model_name = NULL,
  cond_null_model_dir = NULL,
  SPA_p_filter = FALSE,
  p_filter_cutoff = 0.05
)
```

Arguments

`agds_dir` file directory of annotated GDS (aGDS) files for all chromosomes (1-22).

`jobs_num` a data frame containing the number of jobs for association analysis. The data frame must include a column with the name "sliding_window_num"

`input_path` file directory of the sliding window analysis results.

`output_path` file output directory of the summary results.

`sliding_window_results_name` the file name of the input sliding window analysis results.

`obj_nullmodel` an object from fitting the null model, which is either the output from `fit_nullmodel` function in the `STAARpipeline` package, or the output from `fitNullModel` function in the `GENESIS` package and transformed using the `genesis2staar_nullmodel` function in the `STAARpipeline` package.

`known_loci` the data frame of variants to be adjusted for in conditional analysis and should contain 4 columns in the following order: chromosome (CHR), position (POS), reference allele (REF), and alternative allele (ALT) (default = NULL).

`cMAC_cutoff` the cutoff of the minimum number of the cumulative minor allele of variants in the masks when summarizing the results (default = 0).

`method_cond` a character value indicating the method for conditional analysis. `optimal` refers to regressing residuals from the null model on `known_loci` as well as all covariates used in fitting the null model (fully adjusted) and taking the residuals; `naive` refers to regressing residuals from the null model on `known_loci` and taking the residuals (default = `optimal`).

`rare_maf_cutoff` the cutoff of maximum minor allele frequency in defining rare variants (default = 0.01).

`QC_label` channel name of the QC label in the GDS/aGDS file (default = "annotation/filter").

`geno_missing_imputation`
: method of handling missing genotypes. Either "mean" or "minor" (default = "mean").

`variant_type`
: variants include in the conditional analysis. Choices include "variant", "SNV", or "Indel" (default = "SNV").

`Annotation_dir`
: channel name of the annotations in the aGDS file (default = "annotation/info/FunctionalAnnotation").

`Annotation_name_catalog`
: a data frame containing the name and the corresponding channel name in the aGDS file.

`Use_annotation_weights`
: use annotations as weights or not (default = FALSE).

`Annotation_name`
: a vector of annotation names used in STAAR (default = NULL).

`alpha`
: threshod to control the genome-wise (family-wise) error rate (default = 0.05), the p-value threshold is alpha/total number of sliding windows

`manhattan_plot`
: output manhattan plot or not (default = FALSE).

`QQ_plot`
: output Q-Q plot or not (default = FALSE).

`cond_null_model_name`
: the null model name for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`cond_null_model_dir`
: the directory of storing the null model for conditional analysis in the SPA setting, only used for imbalanced case-control setting (default = NULL).

`SPA_p_filter`
: logical: are only the variants with a normal approximation based p-value smaller than a pre-specified threshold use the SPA method to recalculate the p-value, only used for imbalanced case-control setting (default = FALSE).

`p_filter_cutoff`
: threshold for the p-value recalculation using the SPA method, only used for imbalanced case-control setting (default = 0.05).

Value

The function returns the following analysis results:

`results_sliding_window_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the sliding windows across the genome.

`sliding_window_sig.Rdata` and `sliding_window_sig.csv`: a matrix contains the unconditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the significant sliding windows (unconditional p-value<alpha/total number of sliding windows).

`sliding_window_sig_cond.Rdata` and `sliding_window_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O or STAAR-B in imbalanced case-control setting) of the significant sliding windows (available if known_loci is not a NULL).

manhattan plot (optional) and Q-Q plot (optional) of the sliding window analysis results.

References

Li, Z., Li, X., et al. (2022). A framework for detecting noncoding rare-variant associations of large-scale whole-genome sequencing studies. *Nature Methods*, *19*(12), 1599-1611. (pub)

Index

Annotate_Single_Variants, 2

Dynamic_Window_Results_Summary, 3

family, *7*
fit_nullmodel, *7*
fit_nullmodel_genome_cond_spa, 5
formula, *6*

Gene_Centric_Coding_Info, 8
Gene_Centric_Coding_Results_Summary, 9
Gene_Centric_Coding_Results_Summary_incl_ptv, 13
Gene_Centric_Noncoding_Info, 17
Gene_Centric_Noncoding_Results_Summary, 18
glm, *7*

Individual_Analysis_Results_Summary, 23

lm, *6*

Single_Variants_List_Analysis, 25
Sliding_Window_Info, 26
Sliding_Window_Results_Summary, 27