Package ‘STAARpipelineSummary’

January 31, 2022

Type Package

Title Summarization and Visualization of Analysis Results Generated by STAARpipeline

Version 0.9.6

Date 2022-01-31

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

Maintainer Xihao Li <xihaoli@g.harvard.edu>, Zilin Li <li@hsph.harvard.edu>

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

License GPL-3

Copyright See COPYRIGHTS for details.

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

Encoding UTF-8

LazyData true

Depends R (>= 3.2.0)

RoxygenNote 7.1.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

Annotate_Single_Variants 2
Dynamic_Window_Results_Summary 3
Gene_Centric_Coding_Info 5
Gene_Centric_Coding_Results_Summary 7
Gene_Centric_Noncoding_Info 9
Gene_Centric_Noncoding_Results_Summary 11
Individual_Analysis_Results_Summary 14
Single_Variants_List_Analysis 16
Sliding_Window_Info 17
Sliding_Window_Results_Summary 18

Index **21**

`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 = NULL
)
```

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 names of annotation scores (default = NULL).

Value

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

`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` variants include in the conditional analysis. Choice includes "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 names of annotation scores used in variant-set test (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-B.

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-O.

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.

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"),
  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 = NULL
)
```

Arguments

- `category` — the coding functional category of rare variants to be functionally annotated. Choices include `plof`, `plof_ds`, `missense`, `disruptive_missense`, `synonymous` (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` — 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 names of annotation scores (default = NULL).

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 rare variants of the input coding mask.

`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 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,
  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 = 2.5e-06,
  manhattan_plot = FALSE,
  QQ_plot = FALSE
)
```

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).

`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` variants include in the 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` annotations used in STAAR.

`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).

Value

The function returns the following analysis results:

`coding_sig.csv`: a matrix that summarizes the unconditional significant coding masks detected by STAAR-O (STAAR-O 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").

`coding_sig_cond.csv`: a matrix that summarized the conditional analysis results of unconditional significant coding masks detected by STAAR-O (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").

`results_plof_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O) 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) of the unconditional significant plof masks.

`plof_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant plof_ds masks.

`plof_ds_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant disruptive_missense masks.

`disruptive_missense_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant missense masks.

`missense_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant synonymous masks.

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

`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 = NULL
)
```

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` 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 names of annotation scores (default = NULL).

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 rare variants of the input noncoding mask.

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 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,
  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 = 2.5e-06,
  manhattan_plot = FALSE,
  QQ_plot = FALSE
)
```

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).

`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` variants include in the 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` annotations used in STAAR.

`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).

Value

The function returns the following analysis results:

`noncoding_sig.csv`: a matrix that summarized the unconditional significant region detected by STAAR-O (STAAR-O 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).

`noncoding_sig_cond.csv`: a matrix that summarized the conditional analysis results of the unconditional significant region detected by STAAR-O (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).

`results_UTR_genome`: a matrix contains the STAAR p-values (including STAAR-O) 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) of the unconditional significant UTR masks.

`UTR_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant upstream masks.

`upstream_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant downstream masks.

`downstream_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant promoter_CAGE masks.

`promoter_CAGE_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant promoter_DHS masks.

`promoter_DHS_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant enhancer_CAGE masks.

`enhancer_CAGE_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant enhancer_DHS masks.

`enhancer_DHS_sig_cond.csv`: a matrix contains the conditional STAAR p-values (including STAAR-O) 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) 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) of the unconditional significant ncRNA masks.

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

`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
)
```

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).

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).

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

Usage

```
Single_Variants_List_Analysis(
  agds_dir,
  single_variants_list,
  obj_nullmodel,
  QC_label = "annotation/filter",
  geno_missing_imputation = c("mean", "minor")
)
```

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").

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.

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 = NULL
)
```

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 names of annotation scores (default = NULL).

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.

`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 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,
  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,
```

```
  manhattan_plot = FALSE,
  QQ_plot = FALSE
)
```

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).
- `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` 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 names of annotation scores used in variant-set test (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).

Value

The function returns the following analysis results:

`results_sliding_window_genome.Rdata`: a matrix contains the STAAR p-values (including STAAR-O) 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) 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) of the significant sliding windows (available if known_loci is not a NULL).

Index

Annotate_Single_Variants, 2

Dynamic_Window_Results_Summary, 3

Gene_Centric_Coding_Info, 5
Gene_Centric_Coding_Results_Summary, 7
Gene_Centric_Noncoding_Info, 9
Gene_Centric_Noncoding_Results_Summary, 11

Individual_Analysis_Results_Summary, 14

Single_Variants_List_Analysis, 16
Sliding_Window_Info, 17
Sliding_Window_Results_Summary, 18