Package 'CCAFE'

November 11, 2026

Type Package

Title Case Control Allele Frequency Estimation

Version 1.4.0

Description Functions to reconstruct case and control AFs from summary statistics.
One function uses OR, NCase, NControl, and SE(log(OR)).
The second function uses OR, NCase, NControl, and AF for the whole sample.

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Imports dplyr, VariantAnnotation

Depends R (>= 4.4.0)

Suggests testthat (>= 3.0.0), rmarkdown, markdown, knitr, tidyverse,
DescTools, cowplot, BiocStyle, GenomicRanges,
SummarizedExperiment, S4Vectors, IRanges

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/wolffha/CCAFE/

BugReports https://github.com/wolffha/CCAFE/issues

biocViews GenomeWideAssociation, ComparativeGenomics, Genetics,
Preprocessing, SNP, Software, WholeGenome

git_url https://git.bioconductor.org/packages/CCAFE

git_branch RELEASE_3_23

git_last_commit 91b20f3

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Hayley Wolff [cre, aut]

Maintainer Hayley Wolff <hayley.wolff@cuanschutz.edu>

Contents

CCAFE-package 2
CaseControl_AF 2
CaseControl_SE 4
CCAFE_convertVCF 6
sampleDat 7
vcf_sample 8

Index **9**

`CCAFE-package` *CCAFE: Case Control Allele Frequency Estimation*

Description

Functions to reconstruct case and control AFs from summary statistics. One function uses OR, NCase, NControl, and SE(log(OR)). The second function uses OR, NCase, NControl, and AF for the whole sample.

Author(s)

Maintainer: Hayley Wolff <hayley.wolff@cuanschutz.edu>

See Also

Useful links:

- https://github.com/wolffha/CCAFE/
- Report bugs at https://github.com/wolffha/CCAFE/issues

`CaseControl_AF` *CaseControl_AF*

Description

This is a function to derive the case and control AFs from GWAS summary statistics when the user has access to the whole sample AF, the sample sizes, and the OR (or beta). If user has SE instead of sample AF use `CaseControl_SE()`

Usage

```
CaseControl_AF(
  data,
  N_case = 0,
  N_control = 0,
  OR_colname = "OR",
  AF_total_colname = "AF"
)
```

Arguments

data	dataframe with each row being a variant and columns for AF_total and OR
N_case	the number of cases in the sample
N_control	the number of controls in the sample
OR_colname	a string containing the exact column name in 'data' with the OR
AF_total_colname	a string containing the exact column name in 'data' with the whole sample AF

Value

returns a dataframe with two columns (AF_case, AF_control) and rows equal to the number of variants

Author(s)

Hayley Wolff (Stoneman), <hayley.wolff@cuanschutz.edu>

References

https://github.com/wolffha/CCAFE

See Also

https://github.com/wolffha/CCAFE for further documentation

Examples

```
library(CCAFE)

data("sampleDat")
sampleDat <- as.data.frame(sampleDat)

nCase_sample = 16550
nControl_sample = 403923

# get the estimated case and control AFs
af_method_results <- CaseControl_AF(data = sampleDat,
                                    N_case = nCase_sample,
                                    N_control = nControl_sample,
                                    OR_colname = "OR",
                                    AF_total_colname = "true_maf_pop")

head(af_method_results)
```

CaseControl_SE *CaseControl_SE*

Description

This is a function to derive the case, control, and total MAFs from GWAS summary statistics when the user has access to the sample sizes, and the OR (or beta), and SE for the log(OR) for each variant. If user has total AF instead of SE use `CaseControl_AF()` This code uses the GroupFreq function adapted from C from https://github.com/Paschou-Lab/ReAct/blob/main/GrpPRS_src/CountConstruct.c

Usage

```
CaseControl_SE(
  data,
  N_case = 0,
  N_control = 0,
  OR_colname = "OR",
  SE_colname = "SE",
  chromosome_colname = "chr",
  sex_chromosomes = FALSE,
  position_colname = "pos",
  N_XX_case = NA,
  N_XX_control = NA,
  N_XY_case = NA,
  N_XY_control = NA,
  do_correction = FALSE,
  correction_data = NA,
  remove_sex_chromosomes = TRUE,
  verbose = FALSE
)
```

Arguments

data	dataframe where each row is a variant and columns contain the OR, SE, chromosome and positions
N_case	an integer of the number of Case individuals
N_control	an integer of the number of Control individuals
OR_colname	a string containing the exact column name in 'data' with the OR
SE_colname	a string containing the exact column name in 'data' with the SE
chromosome_colname	a string containing the exact column name in 'data' with the chromosomes, default "chr"
sex_chromosomes	boolean, TRUE if variants from sex chromosomes are included in the dataset. Sex chromosomes can be numeric (23, 24) or character (X, Y). If numeric, assumes X=23 and Y=24.

`position_colname`
: a string containing the exact column name in 'data' with the position, default "pos"

`N_XX_case`
: the number of XX chromosome case individuals (REQUIRED if sex_chromosomes == TRUE)

`N_XX_control`
: the number of XX chromosome control individuals (REQUIRED if sex_chromosomes == TRUE)

`N_XY_case`
: the number of XY chromosome case individuals (REQUIRED if sex_chromosomes == TRUE)

`N_XY_control`
: the number of XY chromosome control individuals (REQUIRED if sex_chromosomes == TRUE)

`do_correction`
: boolean, TRUE if data is provided to perform correction

`correction_data`
: a dataframe with the following exact columns: CHR, POS, proxy_MAF with data that is harmonized between the proxy true datasets and the observed dataset

`remove_sex_chromosomes`
: boolean, TRUE if should keep autosomes only. This is needed when the number of biological sex males/females per case and control group is not known.

`verbose`
: boolean, determine whether warnings should be displayed (default FALSE)

Value

returns data as a dataframe with three additional columns: MAF_case, MAF_control, MAF_total for the estimated MAFs for each variant. If do_correction = TRUE, then will output 3 additional columns (MAF_case_adj, MAF_control_adj, MAF_total_adj) with the adjusted estimates.

Author(s)

Hayley Wolff (Stoneman), <hayley.wolff@cuanschutz.edu>

References

https://github.com/wolffha/CCAFE

See Also

https://github.com/wolffha/CCAFE for further documentation

Examples

```
library(CCAFE)

data("sampleDat")
sampleDat <- as.data.frame(sampleDat)

nCase_sample = 16550
nControl_sample = 403923
```

```
# get the estimated case and control MAFs
se_method_results <- CaseControl_SE(data = sampleDat,
                                    N_case = nCase_sample,
                                    N_control = nControl_sample,
                                    OR_colname = "OR",
                                    SE_colname = "SE",
                                    chromosome_colname = "CHR",
                                    position_colname = "POS")

head(se_method_results)
```

CCAFE_convertVCF *CCAFE_convertVCF*

Description

Formats information from a VCF object for use in CCAFE methods as follows: From the rowRanges object: seqnames (chromosome), ranges (position), From the geno object: ES (effect size of ALT), SE, AF (allele frequency of ALT)

Usage

```
CCAFE_convertVCF(vcf)
```

Arguments

vcf	a Variant Call Format (VCF) file read in using VariantAnnotation BioConductor package

Value

a dataframe object with columns Position, RSID, Chromosome, REF, ALT, beta, SE, AF, OR

Author(s)

Hayley Wolff (Stoneman), <hayley.wolff@cuanschutz.edu>

Examples

```
library(VariantAnnotation)
library(CCAFE)

# load the data
data("vcf_sample")

# run the method
df_sample <- CCAFE_convertVCF(vcf_sample)
print(head(df_sample))
```

```
# can then use in CCAFE methods
# since we have total AF, will use CaseControl_AF
df_sample <- CaseControl_AF(data = df_sample,
                            N_case = 48286,
                            N_control = 250671,
                            OR_colname = "OR",
                            AF_total_colname = "AF")
head(df_sample)
```

`sampleDat` *PanUKBB and gnomAD Diabetes Data*

Description

This is a subset of 500 variants on chromosome 1 from the Pan-UKBB diabetes GWAS with the whole sample (pop), case, and control minor allele frequency (MAF) for those classified in Pan-UKBB as European (EUR). These variants (which are mapped to GRCh37) have been harmonized with gnomAD non-Finnish European (NFE) MAFs.

Usage

```
data("sampleDat")
```

Format

'sampleDat' A data frame with 500 rows and 11 columns:

CHR chromosome number

POS base-pair position of variant (GRCh37 coordinates)

REF Reference allele

ALT Alternate allele

true_maf_case MAF in EUR cases in Pan-UKBB Diabetes

true_maf_control MAF in EUR controls in Pan-UKBB Diabetes

true_maf_pop MAF in whole EUR sample in Pan-UKBB Diabetes

beta beta from EUR GWAS in Pan-UKBB Diabetes

SE SE of beta from EUR GWAS in Pan-UKBB Diabetes

OR OR from EUR GWAS in Pan-UKBB Diabetes

gnomad_maf MAF in gnomAD NFE

Source

<https://pan.ukbb.broadinstitute.org/docs/per-phenotype-files>

<https://gnomad.broadinstitute.org/downloads>

vcf_sample	*A VCF from this GWAS of Type 2 Diabetes https://doi.org/10.1038/s41588-018-0084-1. containing a subset of 10,000 variants*

Description

A VCF from this GWAS of Type 2 Diabetes https://doi.org/10.1038/s41588-018-0084-1. containing a subset of 10,000 variants

Usage

```
data("vcf_sample")
```

Format

'vcf_sample' A CollapsedVCF

dim: 10000 1 rowRanges(vcf): GRanges with 5 metadata columns: paramRangeID, REF, ALT, QUAL, FILTER info(vcf): DataFrame with 1 column: AF info(header(vcf)): Number Type Description AF A Float Allele Frequency geno(vcf): List of length 9: ES, SE, LP, AF, SS, EZ, SI, NC, ID geno(header(vcf)): Number Type Description ES A Float Effect size estimate relative to the alternative allele SE A Float Standard error of effect size estimate LP A Float -log10 p-value for effect estimate AF A Float Alternate allele frequency in the association study SS A Integer Sample size used to estimate genetic effect EZ A Float Z-score provided if it was used to derive the EFFECT and SE fields SI A Float Accuracy score of summary data imputation NC A Integer Number of cases used to estimate genetic effect ID 1 String Study variant identifier

Index

∗ **datasets**
sampleDat, 7
vcf_sample, 8
∗ **internal**
CCAFE-package, 2

CaseControl_AF, 2
CaseControl_AF(), *4*
CaseControl_SE, 4
CaseControl_SE(), *2*
CCAFE-package, 2
CCAFE_convertVCF, 6

sampleDat, 7

vcf_sample, 8