Package 'BayesQuantify'

November 17, 2023

Title An R package for quantifying the strength of evidence for ACMG/AMP criteria using a Bayesian framework

Version 1.0.0

Description The ACMG/AMP guidelines have undergone continuous review and refinement for different rules, genes, and diseases, driving optimization and enhancing variant interpretation standards in genetic testing. In 2018, the ClinGen Sequence Variant Interpretation Working Group has proposed a Bayesian Classification Framework to model the ACMG/AMP guidelines. This framework has successfully quantified the thresholds for applying PM5 and PP3/BP4. However, there are challenges for clinicians in utilising the Bayesian Classification Framework, as tools and software for convincingly calculating the positive likelihood ratio are lacking.
The BayesQuantify R Package provide a comprehensive functions to define the thresholds for each evidence strength level.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports bootLR,
ComplexHeatmap,
dplyr,
ggplot2,
gridExtra,
patchwork,
reshape2,
scales,
stringr,
ggpie,
stats,
utils,
circlize,
plyr

Depends R (>= 4.1.0)

LazyData true

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`ACMG_Classification` *Classifying variants into five distinct categories according to the 2015 ACMG/AMP guidelines*

Description

Classifying variants into five distinct categories according to the 2015 ACMG/AMP guidelines

Usage

```
ACMG_Classification(data, evidence_col)
```

Arguments

`data`	DataFrame comprising fundamental variant information, evidence labeling, and classification details
`evidence_col`	The column name for ACMG evidence(str)

Value

A new DataFrame that incorporates the input data and the results of variant classification

Examples

```
data("VCI_data")
ACMG_Classification(VCI_data, "Applied Evidence Codes (Met)")
```

add_info — *Count the number of "supportive", "moderate", "strong" and "very strong" strengths of evidence for pathogenicity*

Description

Count the number of "supportive", "moderate", "strong" and "very strong" strengths of evidence for pathogenicity

Usage

```
add_info(data, classification_col)
```

Arguments

`data`	DataFrame comprising fundamental variant information, evidence labeling, and classification details
`classification_col`	The column name for variant classification (str). Variants should be classified into five distinct categories: "P," "LP," "B," "LB," and "VUS."

Value

A new DataFrame that includes the input data and four new columns, these four columns count the number of different pathogenic evidence strengths for each variant, which can be used for further categorization

Examples

```
data("VCI_data")
VCI_data <- add_info(VCI_data, "Assertion")
```

auto_select_postp — *Automatic definition of posterior probability and positive likelihood ratio values for different strengths of evidence*

Description

Automatic definition of posterior probability and positive likelihood ratio values for different strengths of evidence

Usage

```
auto_select_postp(prior_probability)
```

Arguments

`prior_probability`	The prior probability of pathogenicity (proportion of P/LP variants in a set of variants)

Value

Selected odds path for each evidence level, combined odds path and posterior probability of 17 combination rules

Examples

```
auto_select_postp(0.1)
```

BCF *Classifying variants into five distinct categories according to the Bayesian classification framework*

Description

Classifying variants into five distinct categories according to the Bayesian classification framework

Usage

```
BCF(data, evidence_col, prior_p, op_vs)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
evidence_col	The column name for ACMG evidence(str)
prior_p	The prior probability of pathogenicity (proportion of P/LP variants in a set of variants)
op_vs	Odds path of "Very String"

Value

A new DataFrame that incorporates the input data and the results of variant classification

Examples

```
data("VCI_data")
BCF(VCI_data, "Applied Evidence Codes (Met)", 0.1, 350)
```

`ClinVar2020_AJHG_Pejaver_data` *Dataset in the paper "Calibration of computational tools for missense variant pathogenicity classification and ClinGen recommendations for PP3/BP4 criteria".*

Description

A dataset containing the ClinVar 2020 set to validate the calibration procedure proposed by Pejaver et al (2022).

Usage

```
ClinVar2020_AJHG_Pejaver_data
```

Format

A data frame with 9114 rows and 29 variables:

hg19_chr Chromosome

hg19_pos.1.based. Position

ref Reference allele

alt Alternative allele

rs_dbSNP151 rsID

genename Gene name

Ensembl_geneid GeneID

Ensembl_transcriptid TranscriptID

Ensembl_proteinid ProteinID

Uniprot_acc Uniprot Accession

Uniprot_entry UniProt entry name

aavar AA change

clnsig ClinVar Significance

MAF Minor allele frequency

SIFT_score SIFT score

FATHMM_score FATHMM score

VEST4_score VEST4 score

REVEL_score REVEL score

GERP.._RS GERP++ score

phyloP100way_vertebrate phyloP score

EA_1.0 EA score

BayesDel_nsfp33a_noAF BayesDel score

MutPred2.0_score MutPred score

CADDv1.6_PHRED CADD score

pph2_prob pph2 score

MPC_score MPC score

PrimateAI_score PrimateAI score ...

Source

https://zenodo.org/records/8347415

`discrete_cutoff` *Introducing columns to assess if the observed value is above (1) or below (0) a tested cutoff. A value of 1 indicates being above the tested cutoff, while 0 indicates being below the tested cutoff*

Description

Introducing columns to assess if the observed value is above (1) or below (0) a tested cutoff. A value of 1 indicates being above the tested cutoff, while 0 indicates being below the tested cutoff

Usage

```
discrete_cutoff(data, feature, range = NULL, criteria = NULL)
```

Arguments

`data`	DataFrame comprising fundamental variant information, evidence labeling, and classification details
`feature`	The column name that requires testing for optimizing the thresholds
`range`	Evaluated intervals
`criteria`	ACMG/AMP guidelines criteria (str)

Value

A fresh DataFrame incorporating the input data with additional column

Examples

```
data("VCI_data")
discrete_cutoff(VCI_data, "Applied Evidence Codes (Met)", criteria = "PM2")
```

`get_lr_threshold` *Establish the thresholds for each level of evidence strength*

Description

Establish the thresholds for each level of evidence strength

Usage

```
get_lr_threshold(postp_list, discountonesided, bootstrap, dir)
```

Arguments

postp_list	A list of posterior probability corresponding to each level of evidence strength
discountonesided	The one-sided confidence intervals
bootstrap	The number of bootstrapping iterations
dir	The directory containing the results of bootstrapping

Value

A list of optimized thresholds

Examples

```
data("ClinVar2020_AJHG_Pejaver_data")
ClinVar2020_AJHG_Pejaver_data <- add_info(ClinVar2020_AJHG_Pejaver_data, "clnsig")
local_bootstrapped_lr(ClinVar2020_AJHG_Pejaver_data, "PrimateAI_score", 0.1, 10, 100, 0.1, "test_dir")
postp_list <- c(0.1778350, 0.3129676, 0.6689245, 0.9754584)
get_lr_threshold(postp_list, 0.05, 10, "test_dir")
```

---

heatmap_LR *Visualize the results of LR+ for each evaluated cutoff*

Description

Visualize the results of LR+ for each evaluated cutoff

Usage

```
heatmap_LR(data, op_list)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
op_list	A list of odds path corresponding to each level of evidence strength

Value

Figures

Examples

```
data("LR_result")
op_list <- c(2.11, 4.46, 19.90, 397)
heatmap_LR(LR_result, op_list)
```

`local_bootstrapped_lr` *The one-sided 95% confidence bound for each estimated lr+ was determined through bootstrapping iterations, enabling the assessment of evidence strength.*

Description

The one-sided 95% confidence bound for each estimated lr+ was determined through bootstrapping iterations, enabling the assessment of evidence strength.

Usage

```
local_bootstrapped_lr(
  input_data,
  feature,
  alpha,
  bootstrap,
  minpoints,
  increment,
  output_dir
)
```

Arguments

`input_data`	DataFrame comprising fundamental variant information, evidence labeling, and classification details
`feature`	The column name that requires testing for optimizing the thresholds
`alpha`	Prior probability
`bootstrap`	The number of bootstrapping iterations
`minpoints`	The number of at least pathogenic and non-pathogenic variants
`increment`	Sliding window
`output_dir`	Output directory

Value

The posterior probability values for each bootstrap iteration

Examples

```
data("ClinVar2020_AJHG_Pejaver_data")
ClinVar2020_AJHG_Pejaver_data <- add_info(ClinVar2020_AJHG_Pejaver_data, "clnsig")
local_bootstrapped_lr(ClinVar2020_AJHG_Pejaver_data, "PrimateAI_score", 0.1, 10, 100, 0.1, "test_dir")
```

local_lr	*Calculating the local positive likelihood ratio (lr+) value, which is applicable to continuous evidence proposed by Pejaver et al. First, all unique tested cutoff values were sorted, then each value was positioned at the center of a sliding window. The posterior probability was calculated for each tested cutoff value within the interval, considering a minimum of selected pathogenic and non-pathogenic variants.*

Description

Calculating the local positive likelihood ratio (lr+) value, which is applicable to continuous evidence proposed by Pejaver et al. First, all unique tested cutoff values were sorted, then each value was positioned at the center of a sliding window. The posterior probability was calculated for each tested cutoff value within the interval, considering a minimum of selected pathogenic and non-pathogenic variants.

Usage

```
local_lr(input_data, feature, alpha, minpoints, increment)
```

Arguments

input_data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
feature	The column name that requires testing for optimizing the thresholds
alpha	Prior probability
minpoints	The number of at least pathogenic and non-pathogenic variants
increment	Sliding window

Value

The posterior probability value for each tested cutoff

Examples

```
data("ClinVar2020_AJHG_Pejaver_data")
ClinVar2020_AJHG_Pejaver_data <- add_info(ClinVar2020_AJHG_Pejaver_data, "clnsig")
local_lr(ClinVar2020_AJHG_Pejaver_data, "PrimateAI_score", 0.1, 100, 0.1)
```

LR *Calculating positive likelihood ratio (LR) for each tested cutoff (for discrete cutoffs) For each cutoff, true positive (TP, the number of P/LP variants above a tested cutoff), false positive (FP, the number of BL-VUS/B/LB variants above a tested cutoff), true negative (TN, the number of BL-VUS/B/LB variants below a tested cutoff), and false negative (FN, the number of P/LP variants below a tested cutoff) were estimated. Subsequently, LR+, overall accuracy, true positive rate (sensitivity), true negative rate (specificity), positive predictive value (PPV), negative predictive value (NPV), and F1 score were calculated. Estimates of 95% CI of LR+ were generated using bootstrapping in the R package, bootLR.*

Description

Calculating positive likelihood ratio (LR) for each tested cutoff (for discrete cutoffs) For each cutoff, true positive (TP, the number of P/LP variants above a tested cutoff), false positive (FP, the number of BL-VUS/B/LB variants above a tested cutoff), true negative (TN, the number of BL-VUS/B/LB variants below a tested cutoff), and false negative (FN, the number of P/LP variants below a tested cutoff) were estimated. Subsequently, LR+, overall accuracy, true positive rate (sensitivity), true negative rate (specificity), positive predictive value (PPV), negative predictive value (NPV), and F1 score were calculated. Estimates of 95% CI of LR+ were generated using bootstrapping in the R package, bootLR.

Usage

```
LR(data, start, end)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
start	The beginning column index of the evaluated cutoffs
end	The concluding column index of evaluated cutoffs

Value

A DataFrame comprising the evaluation metrics for each assessed cutoff

Examples

```
data("VCI_data")
VCI_data <- add_info(VCI_data, "Assertion")
VCI_data <- VUS_classify(VCI_data, "Assertion", "Applied Evidence Codes (Met)")
truth_set <- VCI_data[VCI_data$VUS_class != "Hot" & VCI_data$VUS_class != "Warm" & VCI_data$VUS_class != "Tepid
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PM2")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PP3")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PP1")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PVS1")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PS1")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PS2")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PM3")
```

```
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PM4")
truth_set <- discrete_cutoff(truth_set, "Applied Evidence Codes (Met)", criteria = "PM5")
LR(truth_set, 28, 36)
```

`lr_CI` *Merging the results from bootstrap*

Description

Merging the results from bootstrap

Usage

```
lr_CI(bootstrap, dir)
```

Arguments

`bootstrap`	The number of bootstrapping iterations
`dir`	The directory containing the results of bootstrapping

Value

A DataFrame containing posterior probabilities and the 95% confidence interval lower bounds of posterior probabilities for each cutoff

Examples

```
data("ClinVar2020_AJHG_Pejaver_data")
ClinVar2020_AJHG_Pejaver_data <- add_info(ClinVar2020_AJHG_Pejaver_data, "clnsig")
local_bootstrapped_lr(ClinVar2020_AJHG_Pejaver_data, "PrimateAI_score", 0.1, 30, 100, 0.1, "test_dir")
lr_CI_result <- lr_CI(30, "test_dir")
```

`lr_CI_result` *locallr results for PrimateAI_score in ClinVar2020_AJHG_Pejaver_data*

Description

locallr results for PrimateAI_score in ClinVar2020_AJHG_Pejaver_data

Usage

```
lr_CI_result
```

Format

A data frame with 8586 rows and 3 variables:

test_cutoff Each PrimateAI score

Posterior Posterior probability

Posterior1 The 95% CI lower boundry of posterior probability ...

Source

ClinVar2020_AJHG_Pejaver_data

`LR_result` *LR results for VCI_data*

Description

LR results for VCI_data

Usage

```
LR_result
```

Format

A data frame with 8 rows and 20 variables:

TP True positive

FN False negative

FP False positive

TN True negative

Accuracy (TP+TN)/Total

PPV Positive predictive values

NPV Negative predictive values

FNR False negative rate

FPR False positive rate

FOR False omission rate

FDR False discovery rate

F1 F1 score

Sensitivity True positive rate

Specificity True negative rate

posLR Positive likelihood ratio

posLR_LB The 95% CI lower boundry of posLR

posLR_UB The 95% CI upper boundry of posLR

negLR Negative likelihood ratio

negLR_LB The 95% CI lower boundry of negLR

negLR_UB The 95% CI upper boundry of negLR ...

Source

VCI_data

multi_plot *Visualize the distribution of variants*

Description

Visualize the distribution of variants

Usage

```
multi_plot(data, classification_col, gene, consequence = NULL)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
classification_col	The column name for variant classification (str)
gene	The column name for the gene where the variant is located(str)
consequence	The column name for the annotation results of variant consequences(str)

Value

Figures

Examples

```
data("VCI_data")
VCI_data <- add_info(VCI_data, "Assertion")
VCI_data <- VUS_classify(VCI_data, "Assertion", "Applied Evidence Codes (Met)")
multi_plot(VCI_data, "Assertion", "HGNC Gene Symbol")
```

op_postp *Calculate the corresponding combined odds_path and posterior probability of 17 combination rules for a given prior_probability and odds_path of pathogenicity*

Description

Calculate the corresponding combined odds_path and posterior probability of 17 combination rules for a given prior_probability and odds_path of pathogenicity

Usage

```
op_postp(prior_probability, op_vs)
```

Arguments

prior_probability	The prior probability of pathogenicity (proportion of P/LP variants in a set of variants)
op_vs	Odds path of "Very String"

Value

Combined odds_path and posterior probability of 17 combination rules outlined by avtigian et al.(2018)

Examples

```
op_postp(0.1, 350)
```

---

plot_lr *Generate plots depicting the results of lr+ for each tested cutoff*

Description

Generate plots depicting the results of lr+ for each tested cutoff

Usage

```
plot_lr(data, postp_list)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
postp_list	A list of posterior probability corresponding to each level of evidence strength

Value

Figures

Examples

```
data("lr_CI_result")
# ClinVar2020_AJHG_Pejaver_data <- add_info(ClinVar2020_AJHG_Pejaver_data, "clnsig")
# local_bootstrapped_lr(ClinVar2020_AJHG_Pejaver_data, "PrimateAI_score", 0.1, 30, 100, 0.1, "test_dir")
postp_list <- c(0.1778350, 0.3129676, 0.6689245, 0.9754584)
# lr_CI_result <- lr_CI(30, "test_dir")
plot_lr(lr_CI_result, postp_list)
```

Point_Classification *Classifying variants into five distinct categories according to the scaled point system.*

Description

Classifying variants into five distinct categories according to the scaled point system.

Usage

```
Point_Classification(data, evidence_col)
```

Arguments

data	DataFrame comprising fundamental variant information, evidence labeling, and classification details
evidence_col	The column name for ACMG evidence(str)

Value

A new DataFrame that incorporates the input data and the results of variant classification

Examples

```
data("VCI_data")
Point_Classification(VCI_data, "Applied Evidence Codes (Met)")
```

VCI_data *Clinical variant classification in ClinGen Variant Curation Interface*

Description

A dataset containing the curated 5724 variants by ClinGen.

Usage

```
VCI_data
```

Format

A data frame with 5724 rows and 20 variables:

#Variation Variation, in HGVSc

ClinVar Variation Id ClinVar Variation ID

Allele Registry Id ClinGen Allele Registry ID

HGVS Expressions HGVS Expressions in ClinVar

HGNC Gene Symbol Gene Symbol

Disease Variant related disease

Mondo Id Mondo Disease Ontology ID

Mode of Inheritance Genetic Inheritance pattern

Assertion Variant Classification

Applied Evidence Codes (Met) Criteria, represent following the SVI's recommendations

Applied Evidence Codes (Not Met) Criteria not used

Summary of interpretation Detailed information for each applied criteria

PubMed Articles PubMed ID

Expert Panel The name of variant curation expert panel

Guideline Links of specific guidelines

Approval Date Approval Date

Published Date Published Date

Retracted Retracted, in logical

Evidence Repo Link Evidence Repo Link

Uuid ID ...

Source

https://erepo.clinicalgenome.org/evrepo/

`VUS_classify`

Variants of uncertain significance (VUS) were categorized into six levels (hot, warm, tepid, cool, cold, and ice cold), according to the Association for Clinical Genomic Science (ACGS) Best Practice Guidelines

1. *hot: 1 very strong or 1 strong + 1 supporting or 2 moderate + 1 supporting or 1 moderate + 3 supporting evidence;*
2. *warm: 1 strong or 2 moderate or 1 moderate + 2 supporting or 4 supporting evidence;*
3. *tepid: 1 moderate + 1 supporting or 3 supporting evidence;*
4. *cool: 1 moderate or 2 supporting evidence;*
5. *cold: 1 supporting evidence;*
6. *ice cold: no supporting evidence (https://www.acgs.uk.com/quality/best-practice-guidelines/#VariantGuidelines). Variants classified as cool, cold, or ice cold were considered as benign-leaning VUS, unlikely to be disease-causing. Variants classified as hot, warm, or tepid were considered to be pathogenic-leaning VUS.*

Description

Variants of uncertain significance (VUS) were categorized into six levels (hot, warm, tepid, cool, cold, and ice cold), according to the Association for Clinical Genomic Science (ACGS) Best Practice Guidelines

1. hot: 1 very strong or 1 strong + 1 supporting or 2 moderate + 1 supporting or 1 moderate + 3 supporting evidence;

2. warm: 1 strong or 2 moderate or 1 moderate + 2 supporting or 4 supporting evidence;
3. tepid: 1 moderate + 1 supporting or 3 supporting evidence;
4. cool: 1 moderate or 2 supporting evidence;
5. cold: 1 supporting evidence;
6. ice cold: no supporting evidence (https://www.acgs.uk.com/quality/best-practice-guidelines/#VariantGuidelines). Variants classified as cool, cold, or ice cold were considered as benign-leaning VUS, unlikely to be disease-causing. Variants classified as hot, warm, or tepid were considered to be pathogenic-leaning VUS.

Usage

```
VUS_classify(data, classification_col, evidence_col)
```

Arguments

`data`	DataFrame comprising fundamental variant information, evidence labeling, and classification details
`classification_col`	The column name for variant classification (str). Variants should be classified into five distinct categories: "P," "LP," "B," "LB," and "VUS."
`evidence_col`	The column name for ACMG evidence(str). The content of this column should be composed of evidence names and their strengths, connected by semicolons or comma, such as "PM2_Supporting;PM5;BP4"

Value

A new DataFrame that includes the input data and VUS classification

Examples

```
data("VCI_data")
VCI_data <- VUS_classify(VCI_data, "Assertion", "Applied Evidence Codes (Met)")
```

Index