Package ‘genpwr’

April 7, 2020

Title  Power Calculations Under Genetic Model Misspecification

Version  1.0.2


Depends  R (>= 3.5.0)

License  GPL-3

Imports  ggplot2, nleqslv, MASS, stats, utils

Encoding  UTF-8

LazyData  true

RoxygenNote  7.1.0

Suggests  knitr, rmarkdown

VignetteBuilder  knitr

NeedsCompilation  no

Author  Camille Moore [aut, cre], Sean Jacobson [aut]

Maintainer  Camille Moore <moorec@njhealth.org>

Repository  CRAN

Date/Publication  2020-04-07 17:30:02 UTC
R topics documented:

add.fun.t .......................................................... 3
add.or.function ..................................................... 4
additive.ll .......................................................... 5
additive.ll.linear .................................................. 5
as.numeric2 .......................................................... 6
calc.like ............................................................ 7
calc.like.linear ...................................................... 7
calc.like.linear.log.envir.interaction ............................ 8
df2.ll ................................................................. 9
df2.ll.linear ........................................................ 10
dom.fun.t ............................................................ 11
dom.or.function ..................................................... 11
dominant.ll .......................................................... 12
dominant.ll.linear ................................................. 13
es.calc.linear ....................................................... 14
expected.linear.ll .................................................. 15
expected.linear.ll.lin.env ...................................... 15
find.prob.dom ....................................................... 16
find.prob.rec ....................................................... 17
genpwrcalc ......................................................... 17
integrand_funct_case ............................................... 20
integrand_funct_control ........................................... 20
linear.mles ........................................................ 21
linear.mles.lin.envir.interaction ............................... 22
linear.mles.lin.envir.interaction_reduced ....................... 23
linear.mles.log.envir.interaction ................................ 24
linear.outcome.lin.envir.interaction.sds ....................... 25
linear.outcome.lin.envir.interaction.sds_reduced ............... 26
linear.outcome.log.envir.interaction.sds ...................... 27
linear.sds ........................................................ 28
ll.ge.logistic ...................................................... 28
ll.ge.logistic.lin.env ........................................... 29
ll.linear.selector ................................................ 30
ll_zero_finder2 .................................................... 31
logistic.mles ...................................................... 32
logit ............................................................... 32
ncp.search ........................................................ 33
null.ll .............................................................. 34
null.ll.linear ...................................................... 34
odds_ratio_function ............................................... 35
or.function.2df .................................................... 36
or.plot ............................................................ 37
or_calc ............................................................. 38
power.calc ........................................................ 39
power.calc.linear ............................................... 40
power.plot ........................................................ 41
Function to Calculate t matrix for logistic outcome with binary environment interaction in additive model

Description

Calculates the t matrix for logistic outcome with binary environment interaction in additive model

Usage

add.fun.t(MAF, P_e, OR_E, OR_G, OR_GE, Case.Rate)

Arguments

MAF 
Vector of minor allele frequencies

P_e 
Vector of proportions of the population with exposure to the environmental effect

OR_E 
Vector of environmental odds ratios to detect

OR_G 
Vector of genetic odds ratios to detect

OR_GE 
Vector of genetic/environmental interaction odds ratios to detect

Case.Rate 
proportion of cases in the sample (cases/(cases + controls)).
Value

t matrix for all combinations of environment/outcome

Examples

```r
add.fun.t(MAF = 0.1, P_e = 0.2, Case.Rate = 0.5,
OR_G = 1.5, OR_E = 2, OR_GE = 1.8)
```

---

## Additive Model Function

**Description**

Operates within odds_ratio_function to calculate odds ratios for a Test.Model of "Additive"

**Usage**

```r
add.or.function(like, Case.Rate, P_AA, P_AB, P_BB, True.Model, risk_allele)
```

**Arguments**

- `like` Expected log likelihood
- `Case.Rate` proportion of cases in the sample (cases/(cases + controls)).
- `P_AA` Probability the allele is homozygous for the major allele
- `P_AB` Probability the allele is heterozygous
- `P_BB` Probability the allele is homozygous for the minor allele
- `True.Model` A vector object specifying the true underlying genetic model(s): 'Dominant', 'Additive', or 'Recessive'
- `risk_allele` Logical: If OR > 1, the allele is classified as a "risk allele"

**Value**

: The odds ratios and their corresponding genetic model(s)

**Examples**

```r
add.or.function(like=-0.57162, Case.Rate=0.3, P_AA=0.5625, P_AB=0.375,
P_BB=0.0625, True.Model="Additive", risk_allele=TRUE)
```
additive.ll  

*Function to Calculate Additive Log Likelihood for a Logistic Regression Model*

**Description**

Calculates the log likelihood for a given set of logistic regression coefficients under an additive genetic model.

**Usage**

```r
additive.ll(beta, t)
```

**Arguments**

- `beta` Vector of logistic regression coefficients.
- `t` A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns=genotype.

**Value**

The log likelihood.

**Examples**

```r
additive.ll(c(-0.3793525, -1.1395417),
            rbind(c(0.2339079, 0.05665039, 0.009441731),
                  c(0.3285921, 0.31834961, 0.053058269)))
```

---

additive.ll.linear  

*Function to Calculate Additive Log Likelihood for a Linear Regression Model*

**Description**

Calculates the log likelihood for a given set of linear regression coefficients under an additive genetic model.

**Usage**

```r
additive.ll.linear(beta, m, es, sd_y_x_model, sd_y_x_truth)
```
Arguments

- \texttt{beta} \hspace{1cm} \text{Vector of linear regression coefficients.}
- \texttt{m} \hspace{1cm} \text{Minor allele frequency.}
- \texttt{es} \hspace{1cm} \text{Vector of effect sizes with two elements, (mean AB - mean AA) and (mean BB - mean AA).}
- \texttt{sd_y_x_model} \hspace{1cm} \text{The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.}
- \texttt{sd_y_x_truth} \hspace{1cm} \text{The standard deviation of Y given X (predictors/genotype) given genotype under the true model.}

Value

The log likelihood.

Examples

\begin{verbatim}
additive.ll.linear(beta = c(-0.03, 0.3), m = 0.1, es = c(0,3),
                sd_y_x_model = 0.9918669, sd_y_x_truth = 0.9544108)
\end{verbatim}

\begin{verbatim}
as.numeric2(char)
\end{verbatim}

Description

convert to numeric with scientific notation containing the "." character

Usage

\texttt{as.numeric2(char)}

Arguments

- \texttt{char} \hspace{1cm} \text{string to be converted to numeric}

Value

a number

Examples

\begin{verbatim}
as.numeric2("2e.2")
\end{verbatim}
calc.like Function to Calculate Log Likelihood for a Logistic Regression Model

Description

Convenience function to calculate the log likelihood of a specified model.

Usage

calc.like(beta, t, model)

Arguments

- beta: Vector of logistic regression coefficients.
- t: A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns = genotype.
- model: The genetic model in the logistic regression: "Dominant", "Additive", "Recessive", "2df" or "null"

Value

The log likelihood.

Examples

t <- rbind(c(0.2967437, 0.1806723, 0.02258404), c(0.3432563, 0.1393277, 0.01741596))
calc.like(logistic.mles(t, "Dominant"), t, model="Dominant")

calc.like.linear Function to Calculate Log Likelihood for a Linear Regression Model

Description

Convenience function to calculate the log likelihood of a specified model.

Usage

calc.like.linear(beta, m, es_ab, es_bb, sd_y_x_model, sd_y_x_truth, model)
Arguments

- `beta` Vector of linear regression coefficients.
- `m` Minor allele frequency.
- `es_ab` effect size for mean AB - mean AA
- `es_bb` effect size for mean BB - mean AA
- `sd_y_x_model` The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.
- `sd_y_x_truth` The standard deviation of Y given X (predictors/genotype) given genotype under the true model.
- `model` The genetic model in the linear regression: "Dominant", "Additive", "Recessive", "2df" or "null"

Value

The log likelihood.

Examples

```r
calc.like.linear(beta = c(0.0000000, 0.1578947), m = 0.1, es_ab = 0, es_bb = 3,
                  sd_y_x_model = 0.9980797, sd_y_x_truth = 0.9544108, model = "Dominant")
```

Description

Returns the standard deviation of y given x for linear models with logistic environment interaction

Usage

```r
calc.like.linear.log.envir.interaction(
    beta_hat,
    MAF,
    P_e,
    ES_G,
    ES_E,
    ES_GE,
    sd_y_x_truth,
    sd_y_x_model,
    Test.Model,
    True.Model,
    reduced = F
)
```
**Arguments**

- **beta_hat**: Effect sizes from MLE
- **MAF**: Minor allele Frequency
- **P_e**: Population prevalence of logistic environmental factor
- **ES_G**: Genetic Effect size
- **ES_E**: Environment Effect size
- **ES_GE**: Environment x Genetic interaction Effect size
- **sd_y_x_truth**: Standard deviation of y for the true model
- **sd_y_x_model**: Standard deviation of y for the test model
- **Test.Model**: Test model
- **True.Model**: True model
- **reduced**: logical, indicates whether the X matrix will be used for a reduced model

**Value**

The standard deviation of y given x for linear models with logistic environment interaction

**Examples**

```r
beta_hat = linear.mles.log.envir.interaction(MAF = 0.1, P_e = 0.2, ES_G = 1.2, ES_E = 1.3, ES_GE = 2, Test.Model = "Dominant", True.Model = "Additive")
calc.like.linear.log.envir.interaction(beta_hat = beta_hat, MAF = 0.1, P_e = 0.2, ES_G = 1.2, ES_E = 1.3, ES_GE = 2, sd_y_x_truth = 9.947945, sd_y_x_model = 9.949468, True.Model = "Additive", Test.Model="Dominant")
```

**df2.ll**

*Function to Calculate 2df Log Likelihood for a Logistic Regression Model*

**Description**

Calculates the log likelihood for a given set of logistic regression coefficients under an unspeci-fied/2df genetic model.

**Usage**

```r
df2.ll(beta, t)
```

**Arguments**

- **beta**: Vector of logistic regression coefficients.
- **t**: A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns=genotype.
df2.ll.linear

Function to Calculate 2 Degree of Freedom Log Likelihood for a Linear Regression Model

Description
Calculates the log likelihood for a given set of linear regression coefficients under a 2df model.

Usage
\[
df2.ll.linear(\beta, m, es, sd_y_x_model, sd_y_x_truth)
\]

Arguments
- **beta**: Vector of linear regression coefficients.
- **m**: Minor allele frequency.
- **es**: Vector of effect sizes with two elements, (mean AB - mean AA) and (mean BB - mean AA).
- **sd_y_x_model**: The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.
- **sd_y_x_truth**: The standard deviation of Y given X (predictors/genotype) given genotype under the true model.

Value
The log likelihood.

Examples
\[
df2.ll.linear(\beta = c(0, 0, 3), m = 0.1, es = c(0,3),
               sd_y_x_model = 0.9544108, sd_y_x_truth = 0.9544108)
\]
dom.fun.t  Function to Calculate t matrix for logistic outcome with binary environment interaction in dominant model

**Description**

Calculates the t matrix for logistic outcome with binary environment interaction in dominant model

**Usage**

```r
dom.fun.t(MAF, P_e, OR_E, OR_G, OR_GE, Case.Rate)
```

**Arguments**

- `MAF`: Vector of minor allele frequencies
- `P_e`: Vector of proportions of the population with exposure to the environmental effect
- `OR_E`: Vector of environmental odds ratios to detect
- `OR_G`: Vector of genetic odds ratios to detect
- `OR_GE`: Vector of genetic/environmental interaction odds ratios to detect
- `Case.Rate`: proportion of cases in the sample (cases/(cases + controls)).

**Value**

t matrix for all combinations of environment/outcome

**Examples**

```r
dom.fun.t(MAF = 0.1, P_e = 0.2, Case.Rate = 0.5, OR_G = 1.5, OR_E = 2, OR_GE = 1.8)
```

---

**dom.or.function**  Dominant Model Function

**Description**

Operates within odds_ratio_function to calculate odds ratios for a Test.Model of "Dominant"

**Usage**

```r
dom.or.function(like, Case.Rate, P_AA, P_AB, P_BB, True.Model, risk_allele)
```
Arguments

like Expected log likelihood
Case.Rate proportion of cases in the sample (cases/(cases + controls)).
P_AA Probability the allele is homozygous for the major allele
P_AB Probability the allele is heterozygous
P_BB Probability the allele is homozygous for the minor allele
True.Model A vector object specifying the true underlying genetic model(s): 'Dominant', 'Additive', or 'Recessive'
risk_allele Logical: If OR > 1, the allele is classified as a "risk allele"

Value

: The odds ratios and their corresponding genetic model(s)

Examples

dom.or.function(like=-0.57162, Case.Rate=0.3, P_AA=0.5625, P_AB=0.375, P_BB=0.0625, True.Model="Dominant", risk_allele=TRUE)

dominant.ll Function to Calculate Dominant Log Likelihood for a Logistic Regression Model

Description

Calculates the log likelihood for a given set of logistic regression coefficients under a dominant genetic model.

Usage

dominant.ll(beta, t)

Arguments

beta Vector of logistic regression coefficients.
t A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns=genotype.

Value

The log likelihood.
Examples

```r
dominant.ll(c(-0.3793525, -1.1395417),
    rbind(c(0.2339079, 0.05665039, 0.009441731),
        c(0.3285921, 0.31834961, 0.053058269)))
```

---

**dominant.ll.linear**  
*Function to Calculate Dominant Log Likelihood for a Linear Regression Model*

**Description**

Calculates the log likelihood for a given set of linear regression coefficients under a dominant genetic model.

**Usage**

```r
dominant.ll.linear(beta, m, es, sd_y_x_model, sd_y_x_truth)
```

**Arguments**

- `beta` Vector of linear regression coefficients.
- `m` Minor allele frequency.
- `es` Vector of effect sizes with two elements, (mean AB - mean AA) and (mean BB - mean AA).
- `sd_y_x_model` The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.
- `sd_y_x_truth` The standard deviation of Y given X (predictors/genotype) given genotype under the true model.

**Value**

The log likelihood.

**Examples**

```r
dominant.ll.linear(beta = c(0.0000000, 0.1578947), m = 0.1, es = c(0,3),
    sd_y_x_model = 0.9980797, sd_y_x_truth = 0.9544108)
```
es.calc.linear  

Function to Calculate Effect Size for Linear Models

Description
Calculates the detectable effect size/regression coefficient, at a given sample size, N, and power, with type 1 error rate, Alpha

Usage
es.calc.linear(
  power = NULL,
  N = NULL,
  MAF = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)

Arguments
- **power**: Vector of the desired power(s)
- **N**: Vector of the desired sample size(s)
- **MAF**: Vector of minor allele frequencies
- **sd_y**: Standard deviation of the outcome in the population (ignoring genotype). Either sd_y_x or sd_y must be specified.
- **Alpha**: the desired type 1 error rate(s)
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value
A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples
```r
es <- es.calc.linear(N=1000, power=0.8,
  MAF=0.1, sd_y = 1, Alpha=0.05,
  True.Model='All', Test.Model='All')
```
expected.linear.ll  

*Function to Calculate Expected Log Likelihood for a Single Genotype*

**Description**

Calculates the expected log likelihood for a single genotype given the true and estimated mean and standard deviation for the outcome.

**Usage**

```r
expected.linear.ll(mean_truth, mean_model, sd_y_x_truth, sd_y_x_model)
```

**Arguments**

- `mean_truth`: Mean of the outcome given X(predictors/genotype) under the true model.
- `mean_model`: Mean of the outcome given X(predictors/genotype) under the test model.
- `sd_y_x_truth`: The standard deviation of Y given X (predictors/genotype) given genotype under the true model.
- `sd_y_x_model`: The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.

**Value**

The log likelihood.

**Examples**

```r
expected.linear.ll(mean_truth = 0, mean_model = 0.03,
                   sd_y_x_model = 1, sd_y_x_truth = 0.9544108)
```

expected.linear.ll.lin.env

*Function to Calculate Expected Log Likelihood for a Single Genotype with linear environment interaction*

**Description**

Calculates the expected log likelihood for a single genotype with linear environment interaction given the true and estimated mean and standard deviation for the outcome.

**Usage**

```r
expected.linear.ll.lin.env(sd_y_x_model)
```
Arguments

sd_y_x_model  The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.

Value

The log likelihood.

Examples

expected.linear.ll.lin.env(4.309354)

find.prob.dom  Dominant probability finding function

Description

Operates within add.or.function to find probability of disease in a dominant truth given AB or BB, additive test model

Usage

find.prob.dom(x, P_AA, P_AB, P_BB, cr, like)

Arguments

x  Probability of disease given AB or BB
P_AA  Probability the allele is homozygous for the major allele
P_AB  Probability the allele is heterozygous
P_BB  Probability the allele is homozygous for the minor allele
cr  proportion of cases in the sample (cases/(cases + controls)).
like  Expected log likelihood

Value

: The "a" in the binomial function ax^2 + bx + c that arises in solution for the additive OR functions

Examples

find.prob.dom(0.1510677, 0.5625, 0.375, 0.0625, 0.3, -0.57162)
**find.prob.rec**

*Recessive probability finding function*

**Description**

Operates within add.or.function to find probability of disease in a recessive truth given AB or BB, additive test model.

**Usage**

```r
find.prob.rec(x, P_AA, P_AB, P_BB, cr, like)
```

**Arguments**

- `x`: Probability of disease given AB or BB
- `P_AA`: Probability the allele is homozygous for the major allele
- `P_AB`: Probability the allele is heterozygous
- `P_BB`: Probability the allele is homozygous for the minor allele
- `cr`: proportion of cases in the sample (cases/(cases + controls)).
- `like`: Expected log likelihood

**Value**

: The "a" in the binomial function ax^2 + bx + c that arises in solution for the additive OR functions

**Examples**

```r
find.prob.rec(0.7072381, 0.5625, 0.375, 0.0625, 0.3, -0.6005743)
```

**genpwr.calc**

*Function to Calculate Power for Linear Models with logistic environment interaction*

**Description**

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha.
Usage

genpwr.calc(
  calc,  # What kind of calculation to perform? sample size ("ss"), power ("power"), or effect size ("es")
  model,  # Distribution of the outcome variable? ("logistic" or "linear")
  ge.interaction = NULL,  # If no environment interaction, should be NULL, otherwise should be "logistic" or "linear"
  N = NULL,  # Vector of the desired sample size(s)
  Power = NULL,  # Vector of the desired power(s)
  MAF = NULL,  # Vector of minor allele frequencies
  Alpha = 0.05,  # the desired type 1 error rate(s)
  P_e = NULL,  # Vector of proportions of the population with exposure to the environmental effect
  sd_e = NULL,  # Standard deviation of the environmental variable
  sd_y = NULL,
  Case.Rate = NULL,
  k = NULL,
  OR = NULL,
  OR_G = NULL,
  OR_E = NULL,
  OR_GE = NULL,
  risk_allele = TRUE,
  ES = NULL,
  ES_G = NULL,
  ES_E = NULL,
  ES_GE = NULL,
  R2 = NULL,
  R2_G = NULL,
  R2_E = NULL,
  R2_GE = NULL,
  True.Model = "All",
  Test.Model = "All"
)

Arguments

calc  # What kind of calculation to perform? sample size ("ss"), power ("power"), or effect size ("es")
model  # Distribution of the outcome variable? ("logistic" or "linear")
ge.interaction  # If no environment interaction, should be NULL, otherwise should be "logistic" or "linear"
N  # Vector of the desired sample size(s)
Power  # Vector of the desired power(s)
MAF  # Vector of minor allele frequencies
Alpha  # the desired type 1 error rate(s)
P_e  # Vector of proportions of the population with exposure to the environmental effect
sd_e  # Standard deviation of the environmental variable
sd_y  Standard deviation of the outcome in the population (ignoring genotype). Either sd_y_x or sd_y must be specified.
Case.Rate  Case Rate of the outcome in the population (ignoring genotype). Either Case.Rate_x or Case.Rate must be specified.
k  Vector of the number of controls per case. Either k or Case.Rate must be specified.
OR  Vector of genetic odds ratios to detect in absence of environmental odds ratios
OR_G  Vector of genetic odds ratios to detect
OR_E  Vector of environmental odds ratios to detect
OR_GE  Vector of genetic/environmental interaction odds ratios to detect
risk_allele Logical: If OR > 1, the allele is classified as a "risk allele"
ES  Vector of effect sizes (difference in means) to detect. Either ES or R2 must be specified.
ES_G  Vector of genetic effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_E  Vector of environmental effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_GE  Vector of genetic/environment interaction effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
R2  Vector of R-squared values to detect. Either ES or R2 must be specified.
R2_G  Vector of genetic R-squared values to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
R2_E  Vector of environmental R-squared values to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
R2_GE  Vector of genetic/environment interaction R-squared values. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
True.Model  A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
Test.Model  A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value
A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples
pw <- genpwr.calc(calc = "power", model = "logistic", ge.interaction = "continuous", N=100, OR_G=2, OR_E=1.4, OR_GE=c(1.5, 2), sd_e = 1.1, MAF=0.1, Case.Rate = 0.3, Alpha=0.05, True.Model="All", Test.Model=c("Dominant", "Recessive"))
integrand_funct_case  
*Function to generate integrand for mle for cases*

**Description**

Returns the standard deviation of y given x for linear models with linear environment interaction

**Usage**

`integrand_funct_case(x1, x2)`

**Arguments**

- `x1`  : "true" part of model
- `x2`  : "test" part of model

**Value**

a function to be used as the integrand for the mle

**Examples**

```r
integrand_funct_case(-1.462531 + 1*0.1823216,
-1.462531 + 1*0.1823216)
```

integrand_funct_control  
*Function to generate integrand for mle for controls*

**Description**

Returns the standard deviation of y given x for linear models with linear environment interaction

**Usage**

`integrand_funct_control(x1, x2)`

**Arguments**

- `x1`  : "true" part of model
- `x2`  : "test" part of model

**Value**

a function to be used as the integrand for the mle
**linear.mles**  

*Function to calculate MLE’s for linear models*

**Examples**

```r
integrand_funct_control(-1.462531 + 1*0.1823216,  
-1.462531 + 1*0.1823216)
```

---

**Description**

Finds the maximum likelihood estimates for a given MAF under the specified genetic model and effect size.

**Usage**

```r
linear.mles(m, es_ab, es_bb, model)
```

**Arguments**

- `m`: minor allele frequency
- `es_ab`: effect size for mean AB - mean AA
- `es_bb`: effect size for mean BB - mean AA
- `model`: The assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive', '2df'

**Value**

A vector of linear regression model coefficients.

**Examples**

```r
linear.mles(m = 0.1, es_ab = 0, es_bb = 3, model = "Dominant")
```
linear.mles.lin.envir.interaction

Function to calculate the standard deviation of y given x for linear models with linear environment interaction

Description

Returns the standard deviation of y given x for linear models with linear environment interaction

Usage

linear.mles.lin.envir.interaction(
  MAF,
  beta0,
  ES_G,
  ES_E,
  ES_GE,
  Test.Model,
  True.Model
)

Arguments

MAF Minor allele Frequency
beta0 baseline value for the outcome
ES_G Genetic Effect size
ES_E Environment Effect size
ES_GE Environment x Genetic interaction Effect size
Test.Model Test Model
True.Model True Model

Value

The standard deviation of y given x for linear models with linear environment interaction

Examples

linear.mles.lin.envir.interaction(MAF = 0.28, ES_G = 0.5, beta0 = -0.28,
ES_E = 1.6, ES_GE = 1.4, Test.Model = "Dominant", True.Model = "Additive")
linear.mles.lin.envir.interaction_reduced

Function to calculate the standard deviation of y given x for linear models with linear environment interaction for the reduced model without GxE interaction

---

**Description**

Returns the standard deviation of y given x for linear models with linear environment interaction

**Usage**

```r
linear.mles.lin.envir.interaction_reduced(
  MAF,
  beta0,
  ES_G,
  ES_E,
  ES_GE,
  Test.Model,
  True.Model
)
```

**Arguments**

- **MAF** Minor allele Frequency
- **beta0** baseline value for the outcome
- **ES_G** Genetic Effect size
- **ES_E** Environment Effect size
- **ES_GE** Environment x Genetic interaction Effect size
- **Test.Model** Test Model
- **True.Model** True Model

**Value**

The standard deviation of y given x for linear models with linear environment interaction

**Examples**

```r
linear.mles.lin.envir.interaction_reduced(MAF = 0.28, ES_G = 0.5, beta0 = -0.28, ES_E = 1.6, ES_GE = 1.4, Test.Model = "Dominant", True.Model = "Additive")
```
Function to calculate the standard deviation of y given x for linear models with logistic environment interaction

## Description

Returns the standard deviation of y given x for linear models with logistic environment interaction

## Usage

```r
linear.mles.log.envir.interaction(
  MAF,
  P_e,
  ES_G,
  ES_E,
  ES_GE,
  Test.Model,
  True.Model,
  reduced = F
)
```

## Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAF</td>
<td>Minor allele Frequency</td>
</tr>
<tr>
<td>P_e</td>
<td>Population prevalence of logistic environmental factor</td>
</tr>
<tr>
<td>ES_G</td>
<td>Genetic Effect size</td>
</tr>
<tr>
<td>ES_E</td>
<td>Environment Effect size</td>
</tr>
<tr>
<td>ES_GE</td>
<td>Environment x Genetic interaction Effect size</td>
</tr>
<tr>
<td>Test.Model</td>
<td>Test model</td>
</tr>
<tr>
<td>True.Model</td>
<td>True model</td>
</tr>
<tr>
<td>reduced</td>
<td>logical, indicates whether the X matrix will be used for a reduced model</td>
</tr>
</tbody>
</table>

## Value

The standard deviation of y given x for linear models with logistic environment interaction

## Examples

```r
linear.mles.log.envir.interaction(MAF = 0.1, P_e = 0.2,
ES_G = 1.2, ES_E = 1.3, ES_GE = 2,
Test.Model = "Dominant", True.Model = "Additive")
```
linear.outcome.lin.envir.interaction.sds

Function to calculate the standard deviation of y given x for linear models with linear environment interaction

Description

Returns the standard deviation of y given x for linear models with linear environment interaction

Usage

linear.outcome.lin.envir.interaction.sds(
   MAF,
   sd_e,
   beta0,
   ES_G,
   ES_E,
   ES_GE,
   mod,
   True.Model,
   sd_y
)

Arguments

MAF Minor allele Frequency
sd_e Standard deviation of linear environmental factor
beta0 baseline value for the outcome
ES_G Genetic Effect size
ES_E Environment Effect size
ES_GE Environment x Genetic interaction Effect size
mod Test model
True.Model True model
sd_y Standard deviation of y

Value

The standard deviation of y given x for linear models with linear environment interaction

Examples

linear.outcome.lin.envir.interaction.sds(MAF = 0.28, beta0 = -0.28,
   sd_y = 5, sd_e = 1, ES_G = 0.5, ES_E = 1.6, ES_GE = 1.4,
   mod = "Dominant", True.Model = "Additive")
Function to calculate the standard deviation of y given x for linear models with linear environment interaction

Description

Returns the standard deviation of y given x for linear models with linear environment interaction

Usage

linear.outcome.lin.envir.interaction.sds_reduced(
    MAF,
    sd_e,
    beta0,
    ES_G,
    ES_E,
    ES_GE,
    mod,  
    True.Model,  
    sd_y
)

Arguments

MAF Minor allele Frequency
sd_e Standard deviation of linear environmental factor
beta0 baseline value for the outcome
ES_G Genetic Effect size
ES_E Environment Effect size
ES_GE Environment x Genetic interaction Effect size
mod Test model
True.Model True model
sd_y Standard deviation of y

Value

The standard deviation of y given x for linear models with linear environment interaction

Examples

linear.outcome.lin.envir.interaction.sds_reduced(MAF = 0.28, beta0 = -0.28, sd_y = 5, sd_e = 1, ES_G = 0.5, ES_E = 1.6, ES_GE = 1.4, mod = "Dominant", True.Model = "Additive")
linear.outcome.log.envir.interaction.sds

Function to calculate the standard deviation of y given x for linear models with logistic environment interaction

Description

Returns the standard deviation of y given x for linear models with logistic environment interaction

Usage

linear.outcome.log.envir.interaction.sds(
  MAF,
  P_e,
  ES_G,
  ES_E,
  ES_GE,
  mod,
  True.Model,
  sd_y,
  reduced = F
)

Arguments

MAF  Minor allele Frequency
P_e  Population prevalence of logistic environmental factor
ES_G Genetic Effect size
ES_E Environment Effect size
ES_GE Environment x Genetic interaction Effect size
mod  Test model
True.Model  True model
sd_y  Standard deviation of y
reduced  logical, indicates whether the X matrix will be used for a reduced model

Value

The standard deviation of y given x for linear models with logistic environment interaction

Examples

linear.outcome.log.envir.interaction.sds(MAF = 0.1, P_e = 0.2, sd_y = 10, ES_G = 1.2, ES_E = 1.3, ES_GE = 2, mod = "Dominant", True.Model = "Additive")
linear.sds

*Functions to Calculate Residual SD for Normal/Continuous Outcomes*

*Function to calculate the standard deviation of y given x for linear models*

### Description

Functions to Calculate Residual SD for Normal/Continuous Outcomes Function to calculate the standard deviation of y given x for linear models

### Usage

```r
linear.sds(m, es_ab, es_bb, sd_y, model)
```

### Arguments

- `m`: minor allele frequency
- `es_ab`: effect size for mean AB - mean AA
- `es_bb`: effect size for mean BB - mean AA
- `sd_y`: the standard deviation of y in the overall population.
- `model`: The assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive', '2df'

### Value

A vector of linear regression model coefficients.

### Examples

```r
linear.sds(m = 0.1, es_ab = 0, es_bb = 3, sd_y = 1, model = "Dominant")
```

---

ll.ge.logistic

*Function to calculate MLE’s for logistic models with logistic environment interaction*

### Description

Finds the maximum likelihood estimates for a given 2x3 table under the specified genetic model.

### Usage

```r
ll.ge.logistic(t, N = NULL, power = NULL, Alpha, mod)
```
Arguments

t A 2x6 table of the joint probabilities of disease, genotype, and environment. Rows are case vs. control and columns are genotypes.
N Sample size
power Power
Alpha Alpha
mod Test model

Value
A vector of logistic regression model coefficients.

Examples

t <- rbind(c(0.2870353, 0.07833006, 0.00435167, 0.09946088, 0.029199878, 0.0016222154),
c(0.3609647, 0.06566994, 0.00364833, 0.06253912, 0.006800122, 0.0003777846))
ll.ge.logistic(t, N = 200, Alpha = 0.05, mod = "Dominant")

Description
Returns the standard deviation of y given x for linear models with linear environment interaction

Usage

ll.ge.logistic.lin.envir(
    sd_e,
    N = NULL,
    MAF,
    power = NULL,
    beta0,
    OR_G,
    OR_E,
    OR_GE,
    Alpha,
    True.Model,
    Test.Model
)
ll.linear.selector

Function to return log likelihood function for specified model type

**Arguments**

- `sd_e`: Standard deviation of the environmental variable
- `N`: desired sample size
- `MAF`: Vector of minor allele frequencies
- `power`: desired power
- `beta0`: the beta0 coefficient in the logistic model
- `OR_G`: Vector of genetic odds ratios to detect
- `OR_E`: Vector of environmental odds ratios to detect
- `OR_GE`: Vector of genetic/environmental interaction odds ratios to detect
- `Alpha`: the desired type 1 error rate(s)
- `True.Model`: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- `Test.Model`: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

**Value**

A function to be used as the integrand for the mle

**Examples**

```r
ll.ge.logistic.lin.envir(sd_e = 1, MAF = 0.2, N = 30, beta0 = -1.462531, OR_G = 1.1, OR_E = 1.2, OR_GE = 1.5, Alpha = 0.05, True.Model = "Dominant", Test.Model = "Dominant")
```

**Description**

Convenience function to return log likelihood function for specified model type

**Usage**

```r
ll.linear.selector(model)
```

**Arguments**

- `model`: The genetic model in the linear regression: "Dominant", "Additive", "Recessive", "2df" or "null"

**Value**

Log likelihood function for specified model type
**ll_zero_finder2**

**Examples**

```r
dl.linear.selector("Dominant")
```

---

**ll_zero_finder2**  
**Zero finding function**

**Description**

Finds the zeros of a function af. Alternative to uniroot, designed specifically to work with the genpwr package. Finds multiple zeros if a function has more than one in the given range.

**Usage**

```r
ll_zero_finder2(af, ii = 6, lower = 0, upper = 1, qdelta = 27)
```

**Arguments**

- `af` The function for which to find the zero(s)
- `ii` Number of iterations. The more iterations, the more accuracy. It is recommended that ii be at least 4.
- `lower` Lower limit of region in which to find the zero
- `upper` Upper limit of region in which to find the zero
- `qdelta` Factor for finding intervals over which the function is close to zero

**Value**

Points over the given interval at which the given function is approximately equal to zero

**Examples**

```r
ll_zero_finder2(function(x) (x-0.5)^2 - 0.1)  
ll_zero_finder2(function(x) 8*x^3 - 11.2*x^2 + 4.56*x - 0.476)
```
logistic.mles  
*Function to calculate MLE's for logistic models*

**Description**
Finds the maximum likelihood estimates for a given 2x3 table under the specified genetic model.

**Usage**
```r
logistic.mles(t, model)
```

**Arguments**
- **t**
  A 2x3 table of the joint probabilities of disease and genotype. Rows are case vs. control and columns are genotypes.
- **model**
  The assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive'

**Value**
A vector of logistic regression model coefficients.

**Examples**
```r
logistic.mles(rbind(c(0.2967437, 0.1806723, 0.02258404),
c(0.3432563, 0.1393277, 0.01741596)), "Dominant")
```

logit  
*Logit Function*

**Description**
Calculates the logit of a specified value.

**Usage**
```r
logit(x, min = 0, max = 1)
```

**Arguments**
- **x**
  a number between 0 and 1.
- **min**
  minimum
- **max**
  maximum
ncp.search

Value

The logit of x.

Examples

logit(0.5)

ncp.search Function to Determine Non-Centrality Parameter of the Chi-squared distribution

Description

This function is set to 0 and solved for x, the non-centrality parameter to determine the sample size in ss.calc

Usage

ncp.search(x, power, Alpha, df)

Arguments

x the non-centrality parameter
power the desired power
Alpha the desired type 1 error rate
df the degrees of freedom for the likelihood ratio test

Value

numeric value of the function

Examples

ncp.search(x = 7.848861, pow = 0.8, Alpha = 0.05, df=1)
null.ll Function to Calculate Null Log Likelihood for a Logistic Regression Model

Description

Calculates the log likelihood for a given set of logistic regression coefficients under the null.

Usage

null.ll(t)

Arguments

   t   A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns=genotype.

Value

The log likelihood.

Examples

null.ll(rbind(c(0.2339079, 0.05665039, 0.009441731),
               c(0.3285921, 0.31834961, 0.053058269)))

null.ll.linear Function to Calculate Expected Null Log Likelihood for a Linear Regression Model

Description

Calculates the expected log likelihood for a given set of linear regression coefficients under the null.

Usage

null.ll.linear(beta, m, es, sd_y_x_model, sd_y_x_truth)
odds_ratio_function

Arguments

- **beta**: Vector of linear regression coefficients.
- **m**: Minor allele frequency.
- **es**: Vector of effect sizes with two elements, (mean AB - mean AA) and (mean BB - mean AA).
- **sd_y_x_model**: The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.
- **sd_y_x_truth**: The standard deviation of Y given X (predictors/genotype) given genotype under the true model.

Value

The log likelihood.

Examples

```r
null.ll.linear(beta = 0.03, m = 0.1, es = c(0,3),
               sd_y_x_model = 1, sd_y_x_truth = 0.9544108)
```

Description

Calculates the odds ratio for a given power, at a given sample size, N, with type 1 error rate, Alpha

Usage

```r
odds_ratio_function(
    N = NULL,
    Case.Rate = NULL,
    k = NULL,
    MAF = NULL,
    power = NULL,
    risk_allele = TRUE,
    Alpha = 0.05,
    True.Model = "All",
    Test.Model = "All"
)
```
or.function.2df

**Arguments**

- **N**: Vector of the desired sample size(s)
- **Case.Rate**: Vector of the proportion(s) of cases in the sample (cases/(cases + controls)). Either k or Case.Rate must be specified.
- **k**: Vector of the number of controls per case. Either k or Case.Rate must be specified.
- **MAF**: Vector of minor allele frequencies
- **power**: Vector of powers to detect
- **risk_allele**: Logical: If OR > 1, the allele is classified as a "risk allele"
- **Alpha**: the desired type 1 error rate(s)
- **True.Model**: A vector vector the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

**Value**

A data frame including the odds ratios for all combinations of the specified parameters

**Examples**

```r
or <- odds_ratio_function(N=c(100), Case.Rate=0.3, k=0, MAF=0.25, power=0.8, Alpha=0.05, risk_allele = TRUE, True.Model = 'All', Test.Model = 'All')
```

---

**or.function.2df**

**2df Model Function**

**Description**

Operates within odds_ratio_function to calculate odds ratios for a Test.Model of "2df"

**Usage**

```r
or.function.2df(like, Case.Rate, P_AA, P_AB, P_BB, True.Model, risk_allele)
```

**Arguments**

- **like**: Expected log likelihood
- **Case.Rate**: proportion of cases in the sample (cases/(cases + controls)).
- **P_AA**: Probability the allele is homozygous for the major allele
- **P_AB**: Probability the allele is heterozygous
- **P_BB**: Probability the allele is homozygous for the minor allele
True.Model  A vector object specifying the true underlying genetic model(s): 'Dominant', 'Additive', or 'Recessive'

risk_allele  Logical: If OR > 1, the allele is classified as a "risk allele"

Value

: The odds ratios and their corresponding genetic model(s)

Examples

or.function.2df(like=-0.5626909, Case.Rate=0.3, P_AA=0.5625, P_AB=0.375, P_BB=0.0625, True.Model="Recessive", risk_allele=TRUE)

or.plot  *Function to Plot Odds Ratio Results*

Description

Plot the power results by MAF, Power, Alpha or N

Usage

or.plot(
    data = NULL,
    x = "MAF",
    panel.by = "True.Model",
    y_limit = NULL,
    y_log = F,
    return_gg = F,
    linear.effect.measure = "ES",
    select.Alpha = NULL,
    select.power = NULL,
    select.ES = NULL,
    select.N = NULL,
    select.MAF = NULL,
    select.Case.Rate = NULL,
    select.SD = NULL,
    select.True.Model = NULL,
    select.Test.Model = NULL
)

Arguments

data  The data frame result from power.calc

x  The desired variable on the y axis: "MAF", "OR", "Alpha", or "N_total"
or_calc

A grouping variable to panel the graphs by: "True.Model", "MAF", "Power", "Alpha", or "N_total".

y_limit
An object specifying the minimum and maximum of the y-axis (eg c(0,4)) default is NULL, which allows the limits to be picked automatically.

y_log
Logical, specifying whether the y axis should be logarithmic. Default is F.

return_gg
Logical, specifying whether to return the ggplot object instead of printing out the plot.

linear.effect.measure
Should the graphs indicate ES values, or R2 values? (default ES).

select.Alpha
Only produce graphs for the specified Alpha level(s).

select.power
Only produce graphs for the specified Power(s).

select.ES
Only produce graphs for the specified effect sizes(s).

select.N
Only produce graphs for the specified sample size(s).

select.MAF
Only produce graphs for the specified minor allele frequency(ies).

select.Case.Rate
Only produce graphs for the specified case rate(s).

select.SD
Only produce graphs for the specified standard deviation(s).

select.True.Model
Only produce graphs for the specified true genetic model(s): "Additive", "Dominant", "Recessive".

select.Test.Model
Only produce graphs for the specified testing model(s): "Additive", "Dominant", "Recessive", "2df".

Value
A series of plots with power on the Y axis.

Examples

```r
or <- odds_ratio_function(N=1000, Case.Rate=0.5, k=NULL, MAF=seq(0.3, 0.32, 0.01), power=0.8, Alpha=0.05, True.Model=c("Dominant", "Recessive"), Test.Model=c("Dominant", "Recessive"))
or.plot(data=or, x='MAF')
```

or_calc

Odds ratio calculation

Description

Calculates odds ratio for given parameters. Used by the function odds_ratio_function.
Usage

\texttt{or\_calc(a, b, c, d, e, f, mod, risk\_allele)}

Arguments

\begin{itemize}
  \item \texttt{a} The probability of a case given homogeneity for the major allele
  \item \texttt{b} The probability of a case given heterozygosity
  \item \texttt{c} The probability of a case given homogeneity for the minor allele
  \item \texttt{d} The probability of a control given homogeneity for the major allele
  \item \texttt{e} The probability of a control given heterozygosity
  \item \texttt{f} The probability of a control given homogeneity for the minor allele
  \item \texttt{mod} The model to be used (eg "Dominant", "Recessive", etc)
  \item \texttt{risk\_allele} Is this allele a risk allele? use T or F
\end{itemize}

Value

Odds ratio

Examples

\texttt{or\_calc(a = 0.3649185, b = 0.12797197, c = 0.007109554,}
\texttt{d= 0.4450815, e= 0.05202803, f = 0.002890446,}
\texttt{mod = "Dominant", risk\_allele = TRUE)}

---

\texttt{power\_calc} \hspace{1cm} \textit{Function to Calculate Power}

Description

Calculates the power to detect an odds ratio, OR, at a given sample size, N, with type 1 error rate, Alpha

Usage

\texttt{power\_calc(}
\texttt{  N = NULL,}
\texttt{  Case\_Rate = NULL,}
\texttt{  k = NULL,}
\texttt{  MAF = NULL,}
\texttt{  OR = NULL,}
\texttt{  Alpha = 0.05,}
\texttt{  True\_Model = "All",}
\texttt{  Test\_Model = "All"}
\texttt{)}
Arguments

- **N**: Vector of the desired sample size(s).
- **Case.Rate**: Vector of the proportion(s) of cases in the sample (cases/(cases + controls)). Either k or Case.Rate must be specified.
- **k**: Vector of the number of controls per case. Either k or Case.Rate must be specified.
- **MAF**: Vector of minor allele frequencies.
- **OR**: Vector of odds ratios to detect.
- **Alpha**: The desired type 1 error rate(s).
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'.
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'.

Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, OR, Power, etc).

Examples

```r
pw <- power.calc(N=2000, Case.Rate=0.5, k=NULL, MAF=0.2, OR=1.5, Alpha=0.05, True.Model='All', Test.Model='All')
```

---

**power.calc.linear Function to Calculate Power for Linear Models**

Description

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha.

Usage

```r
power.calc.linear(
  N = NULL,
  MAF = NULL,
  ES = NULL,
  R2 = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)
```
power.plot

Arguments

N Vector of the desired sample size(s)
MAF Vector of minor allele frequencies
ES Vector of effect sizes (difference in means) to detect. Either ES or R2 must be specified.
R2 Vector of R-squared values to detect. Either ES or R2 must be specified.
sd_y Standard deviation of the outcome in the population (ignoring genotype). Either sd_y_x or sd_y must be specified.
Alpha the desired type I error rate(s)
True.Model A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
Test.Model A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples

pw <- power.calc.linear(N=1000,
MAF=0.1, ES=3, sd_y = 1, Alpha=0.05,
True.Model="All", Test.Model="All")

power.plot Function to Plot Power Results

Description

Plot the power results by MAF, OR, Alpha or N

Usage

power.plot(
  data = NULL,
  x = "MAF",
  panel.by = "True.Model",
  y_limit = NULL,
  y_log = F,
  return_gg = F,
  linear.effect.measure = "ES",
  select.Alpha = NULL,
  select.OR = NULL,
Arguments

data The data frame result from `power.calc`
x The desired variable on the y axis: "MAF", "OR", "Alpha", or "N_total"
panel.by A grouping variable to panel the graphs by: "True.Model", "MAF", "OR", "Alpha", or "N_total"
y_limit An object specifying the minimum and maximum of the y-axis (eg c(0,4)) default is NULL, which allows the limits to be picked automatically
y_log Logical, specifying whether the y axis should be logarithmic. Default is F
return_gg Logical, specifying whether to return the ggplot object instead of printing out the plot
linear.effect.measure Should the graphs indicate ES values, or R2 values? (default ES)
select.Alpha Only produce graphs for the specified Alpha level(s).
select.OR Only produce graphs for the specified odds ratio(s).
select.ES Only produce graphs for the specified effect sizes(s).
select.N Only produce graphs for the specified sample size(s).
select.MAF Only produce graphs for the specified minor allele frequency(ies).
select.Case.Rate Only produce graphs for the specified case rate(s).
select.SD Only produce graphs for the specified standard deviation(s).
select.True.Model Only produce graphs for the specified true genetic model(s): "Additive", "Dominant", "Recessive".
select.Test.Model Only produce graphs for the specified testing model(s): "Additive", "Dominant", "Recessive", "2df".

Value

A series of plots with power on the Y axis.

Examples

```r
pw <- power.calc(N=1000, Case.Rate=c(0.5), k=NULL,
                  MAF=seq(0.15, 0.2, 0.01), OR=1.5, Alpha=c(0.05),
                  True.Model='All', Test.Model='All')
```
Function to Calculate Power for Logistic Models with Environment Interaction

Description

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type I error rate, Alpha

Usage

```r
power_envir.calc(  
  N = NULL,  
  Case.Rate = NULL,  
  k = NULL,  
  MAF = NULL,  
  OR_G = NULL,  
  OR_E = NULL,  
  OR_GE = NULL,  
  P_e = NULL,  
  Alpha = 0.05,  
  True.Model = "All",  
  Test.Model = "All"
)
```

Arguments

- **N**: Vector of the desired sample size(s)
- **Case.Rate**: proportion of cases in the sample (cases/(cases + controls)).
- **k**: Vector of the number of controls per case. Either k or Case.Rate must be specified.
- **MAF**: Vector of minor allele frequencies
- **OR_G**: Vector of genetic odds ratios to detect
- **OR_E**: Vector of environmental odds ratios to detect
- **OR_GE**: Vector of genetic/environmental interaction odds ratios to detect
- **P_e**: Vector of proportions of the population with exposure to the environmental effect
- **Alpha**: the desired type I error rate(s)
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive1', 'Additive2', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'
Value
A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples
pw <- power_envir.calc(P_e = 0.2, MAF = 0.1, N = 200, Case.Rate = 0.5, Alpha = 0.05, OR_G = 1.5, OR_E = 2, OR_GE = 1.8, Test.Model = "All", True.Model = "All")

power_envir.calc.linear_outcome

Function to Calculate Power for Linear Models with logistic environment interaction

Description
Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

Usage
power_envir.calc.linear_outcome(
  N = NULL,
  MAF = NULL,
  ES_G = NULL,
  ES_E = NULL,
  ES_GE = NULL,
  P_e = NULL,
  R2_G = NULL,
  R2_E = NULL,
  R2_GE = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)

Arguments
N Vector of the desired sample size(s)
MAF Vector of minor allele frequencies
ES_G Vector of genetic effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_E Vector of environmental effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_GE  Vector of genetic/environment interaction effect sizes (difference in means) to detect. Either \( ES_G \), \( ES_E \), and \( ES_{EG} \) or \( R^2_G \), \( R^2_E \), and \( R^2_{EG} \) must be specified.

\( P_e \)  Vector of proportions of the population with exposure to the environmental effect

\( R^2_G \)  Vector of genetic R-squared values to detect. Either \( ES_G \), \( ES_E \), and \( ES_{EG} \) or \( R^2_G \), \( R^2_E \), and \( R^2_{EG} \) must be specified.

\( R^2_E \)  Vector of environmental R-squared values to detect. Either \( ES_G \), \( ES_E \), and \( ES_{EG} \) or \( R^2_G \), \( R^2_E \), and \( R^2_{EG} \) must be specified.

\( R^2_{GE} \)  Vector of genetic/environment interaction R-squared values Either \( ES_G \), \( ES_E \), and \( ES_{EG} \) or \( R^2_G \), \( R^2_E \), and \( R^2_{EG} \) must be specified.

\( sd_y \)  Standard deviation of the outcome in the population (ignoring genotype). Either \( sd_y \) or \( sd_y \) must be specified.

\( Alpha \)  the desired type 1 error rate(s)

\( True.Model \)  A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'

\( Test.Model \)  A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

**Value**

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

**Examples**

```r
pw <- power_envir.calc.linear_outcome(N=100, ES_G = 1.2, ES_E = 1.3, ES_GE = 2, Alpha = 0.05, MAF = 0.2, P_e = 0.2, sd_y = 10, True.Model = "All", Test.Model = "All")
```

---

**power_linear_envir.calc.linear_outcome**

Function to Calculate Power for Linear Models with linear environment interaction

**Description**

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, \( N \), with type 1 error rate, \( Alpha \)
Usage

```r
power_linear_envir.calc.linear_outcome(
  N = NULL,
  MAF = NULL,
  ES_G = NULL,
  ES_E = NULL,
  ES_GE = NULL,
  sd_e = NULL,
  R2_G = NULL,
  R2_E = NULL,
  R2_GE = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)
```

Arguments

- **N**: Vector of the desired sample size(s)
- **MAF**: Vector of minor allele frequencies
- **ES_G**: Vector of genetic effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_GE or R2_G, R2_E, and R2_EG must be specified.
- **ES_E**: Vector of environmental effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_GE or R2_G, R2_E, and R2_EG must be specified.
- **ES_GE**: Vector of genetic/environment interaction effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_GE or R2_G, R2_E, and R2_EG must be specified.
- **sd_e**: Standard deviation of the environmental variable
- **R2_E**: Vector of environmental R-squared values to detect. Either ES_G, ES_E, and ES_GE or R2_G, R2_E, and R2_EG must be specified.
- **R2_GE**: Vector of genetic/environment interaction R-squared values to detect. Either ES_G, ES_E, and ES_GE or R2_G, R2_E, and R2_EG must be specified.
- **sd_y**: Standard deviation of the outcome in the population (ignoring genotype). Either sd_y or sd_y must be specified.
- **Alpha**: the desired type 1 error rate(s)
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'
Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples

```r
pw <- power_linear_envir.calc.linear_outcome(N=1000, ES_G=0.5, ES_E=1.6, ES_GE=1.4, sd_e = 1, MAF=0.28, sd_y = 5, Alpha=0.05, True.Model='All', Test.Model='All')
```

Description

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

Usage

```r
power_linear_envir.calc.logistic_outcome(
  N = NULL,
  MAF = NULL,
  OR_G = NULL,
  OR_E = NULL,
  OR_GE = NULL,
  sd_e = NULL,
  Case.Rate = NULL,
  k = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)
```

Arguments

- `N`: Vector of the desired sample size(s)
- `MAF`: Vector of minor allele frequencies
- `OR_G`: Vector of genetic odds ratios to detect
- `OR_E`: Vector of environmental odds ratios to detect
- `OR_GE`: Vector of genetic/environmental interaction odds ratios to detect
sd_e  Standard deviation of the environmental variable
Case.Rate  Standard deviation of the outcome in the population (ignoring genotype). Either Case.Rate_x or Case.Rate must be specified.
k  Vector of the number of controls per case. Either k or Case.Rate must be specified.
Alpha  the desired type 1 error rate(s)
True.Model  A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
Test.Model  A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value
A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples
pw <- power_linear_envir.calc.logistic_outcome(N=30, OR_G=1.1, OR_E=1.2, OR_GE=1.5, sd_e = 1, MAF=0.2, Case.Rate = 0.2, Alpha=0.05, True.Model="All", Test.Model="All")

p_vec_returner
Function to output probability vector used in calculation of MLE’s for linear outcome with logistic environment interaction

Description
Returns probability vector used in calculation of MLE’s for linear outcome with logistic environment interaction

Usage
p_vec_returner(MAF, P_e)

Arguments
MAF  Minor allele frequency
P_e  Population prevalence of logistic environmental factor

Value
A probability vector to be used in MLE calculation for linear outcome with logistic environment interaction
p_vec_returner_lin_env

Examples

```r
p_vec_returner(MAF = 0.1, P_e = 0.2)
```

Description

Returns probability vector used in calculation of MLE’s for linear outcome with linear environment interaction

Usage

```r
p_vec_returner_lin_env(MAF)
```

Arguments

- **MAF**: Minor Allele Frequency

Value

A probability vector to be used in MLE calculation for linear outcome with linear environment interaction

Examples

```r
p_vec_returner_lin_env(0.1)
```

quad_roots

*Function to Solve Quadratic Equations*

Description

Finds the positive root of a quadratic equation \(ax^2 + bx + c\).

Usage

```r
quad_roots(a, b, c)
```
Arguments

a  the coefficient for $x^2$

b  the coefficient for $x$

c  the constant

Value

The positive root of the quadratic equation $ax^2 + bx + c$

Examples

pw<-power.calc(N=c(1000,2000), Case.Rate=c(0.5),
k=NULL, MAF=seq(0.05, 0.1, 0.01), OR=c(3,4),
Alpha=c(0.05), True.Model='All', Test.Model='All')

rec.fun.t  Function to Calculate t matrix for logistic outcome with binary environment interaction in recessive model

Description

Calculates the t matrix for logistic outcome with binary environment interaction in recessive model

Usage

rec.fun.t(MAF, P_e, OR_E, OR_G, OR_GE, Case.Rate)

Arguments

MAF  Vector of minor allele frequencies

P_e  Vector of proportions of the population with exposure to the environmental effect

OR_E  Vector of environmental odds ratios to detect

OR_G  Vector of genetic odds ratios to detect

OR_GE  Vector of genetic/environmental interaction odds ratios to detect

Case.Rate  proportion of cases in the sample (cases/(cases + controls)).

Value

t matrix for all combinations of environment/outcome

Examples

rec.fun.t(MAF = 0.1, P_e = 0.2, Case.Rate = 0.5,
OR_G = 1.5, OR_E = 2, OR_GE = 1.8)
**rec.or.function**  
*Recessive Model Function*

**Description**

Operates within odds_ratio_function to calculate odds ratios for a Test.Model of "Recessive"

**Usage**

`rec.or.function(like, Case.Rate, P_AA, P_AB, P_BB, True.Model, risk_allele)`

**Arguments**

- **like**: Expected log likelihood
- **Case.Rate**: proportion of cases in the sample (cases/(cases + controls)).
- **P_AA**: Probability the allele is homozygous for the major allele
- **P_AB**: Probability the allele is heterozygous
- **P_BB**: Probability the allele is homozygous for the minor allele
- **True.Model**: A vector object specifying the true underlying genetic model(s): 'Dominant', 'Additive', or 'Recessive'
- **risk_allele**: Logical: If OR > 1, the allele is classified as a "risk allele"

**Value**

: The odds ratios and their corresponding genetic model(s)

**Examples**

`rec.or.function(like=-0.57162, Case.Rate=0.3, P_AA=0.5625, P_AB=0.375, P_BB=0.0625, True.Model="Recessive", risk_allele=TRUE)`

---

**recessive.ll**  
*Function to Calculate Recessive Log Likelihood for a Logistic Regression Model*

**Description**

Calculates the log likelihood for a given set of logistic regression coefficients under a recessive genetic model.

**Usage**

`recessive.ll(beta, t)`
Arguments

- **beta**: Vector of logistic regression coefficients.
- **t**: A 2x3 table of joint probabilities of disease and genotype. Rows = case vs. control, columns=genotype.

Value

The log likelihood.

Examples

```r
test <- cbind(c(0.2339079, 0.05665039, 0.009441731),
             c(0.3285921, 0.31834961, 0.053058269))
recessive.ll(beta = c(-0.3793525, -1.1395417),
             t = test)
```

---

**recessive.ll.linear**

*Function to Calculate Recessive Log Likelihood for a Linear Regression Model*

Description

Calculates the log likelihood for a given set of linear regression coefficients under a recessive genetic model.

Usage

```r
recessive.ll.linear(beta, m, es, sd_y_x_model, sd_y_x_truth)
```

Arguments

- **beta**: Vector of linear regression coefficients.
- **m**: Minor allele frequency.
- **es**: Vector of effect sizes with two elements, (mean AB - mean AA) and (mean BB - mean AA).
- **sd_y_x_model**: The standard deviation of Y (the outcome) given X (predictors/genotype) under the test model.
- **sd_y_x_truth**: The standard deviation of Y given X (predictors/genotype) given genotype under the true model.

Value

The log likelihood.

Examples

```r
recessive.ll.linear(beta = c(0, 3), m = 0.1, es = c(0, 3),
                    sd_y_x_model = 0.9544108, sd_y_x_truth = 0.9544108)
```
solve_a  Binomial coefficient calculation

Description

Operates within add.or.function to solve for 'a' when 'b' is known in an additive model

Usage

solve_a(b, cr, P_AA, P_AB, P_BB)

Arguments

b  The "b" in the binomial function ax^2 + bx + c that arises in solution for the additive OR functions
 cr  proportion of cases in the sample (cases/(cases + controls)).
 P_AA  Probability the allele is homozygous for the major allele
 P_AB  Probability the allele is heterozygous
 P_BB  Probability the allele is homozygous for the minor allele

Value

: The "a" in the binomial function ax^2 + bx + c that arises in solution for the additive OR functions

Examples

solve_a(0.1493558, 0.3, 0.5625, 0.375, 0.062)

ss.calc  Function to Calculate Sample Size

Description

Calculates the necessary sample size to achieve the specified level of power to detect an odds ratio, OR, with type 1 error rate, Alpha
Usage

```r
ss.calc(
  power = 0.8,
  Case.Rate = NULL,
  k = NULL,
  MAF = NULL,
  OR = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)
```

Arguments

- **power**: Vector of the desired power(s)
- **Case.Rate**: Vector of the proportion(s) of cases in the sample (cases/(cases + controls)). Either k or Case.Rate must be specified.
- **k**: Vector of the number of controls per case. Either k or Case.Rate must be specified.
- **MAF**: Vector of minor allele frequencies
- **OR**: Vector of odds ratios to detect
- **Alpha**: the desired type 1 error rate(s)
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value

A data frame including the total number of subjects required for all combinations of the specified parameters (Case.Rate, OR, Power, etc)

Examples

```r
ss <- ss.calc(power=0.8, Case.Rate=0.5, k=NULL,
               MAF=0.1, OR=3, Alpha=0.05,
               True.Model='All', Test.Model='All')
```
Function to Calculate Sample Size in Linear Models

Description
Calculates the necessary sample size to achieve the specified level of power to detect an effect size, ES or R2 value, with type 1 error rate, Alpha.

Usage
ss.calc.linear(
  power = 0.8,
  MAF = NULL,
  ES = NULL,
  R2 = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)

Arguments
- power: Vector of the desired power(s)
- MAF: Vector of minor allele frequencies
- ES: Vector of effect sizes (difference in means) to detect. Either ES or R2 must be specified.
- R2: Vector of R-squared values to detect. Either ES or R2 must be specified.
- sd_y: Standard deviation of the outcome in the population (ignoring genotype). Either sd_y_x or sd_y must be specified.
- Alpha: the desired type 1 error rate(s)
- True.Model: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
- Test.Model: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value
A data frame including the total number of subjects required for all combinations of the specified parameters

Examples
ss <- ss.calc.linear(power=0.8, MAF=0.1, ES=3, R2=NULL, sd_y = 1, Alpha=0.05, True.Model='All', Test.Model='All')
ss.plot

Function to Plot Sample Size Results

Description

Plot the sample size results by MAF, OR, Alpha or Power

Usage

ss.plot(
  data = NULL,
  x = "MAF",
  panel.by = "True.Model",
  y_limit = NULL,
  y_log = F,
  return_gg = F,
  linear.effect.measure = "ES",
  select.Alpha = NULL,
  select.OR = NULL,
  select.ES = NULL,
  select.Power = NULL,
  select.MAF = NULL,
  select.Case.Rate = NULL,
  select.SD = NULL,
  select.True.Model = NULL,
  select.Test.Model = NULL
)

Arguments

data The data frame result from ss.calc
x The desired variable on the y axis: "MAF", "OR", "ES","Alpha", or "Power"
panel.by A grouping variable to panel the graphs by: "True.Model", "MAF", "OR", "Alpha", or "Power"
y_limit An object specifying the minimum and maximum of the y-axis (eg c(0,4)) default is NULL, which allows the limits to be picked automatically
y_log Logical, specifying whether the y axis should be logarithmic. Default is F
return_gg Logical, specifying whether to return the ggplot object instead of printing out the plot
linear.effect.measure Should the graphs indicate ES values, or R2 values? (default ES)
select.Alpha Only produce graphs for the specified Alpha level(s).
select.OR Only produce graphs for the specified odds ratio(s).
select.ES Only produce graphs for the specified effect size(s).
select.Power  Only produce graphs for the specified power(s).
select.MAF    Only produce graphs for the specified minor allele frequency(ies).
select.Case.Rate
              Only produce graphs for the specified case rate(s).
select.SD     Only produce graphs for the specified standard deviation(s).
select.True.Model
               Only produce graphs for the specified true genetic model(s): "Additive", "Dominant", "Recessive".
select.Test.Model
               Only produce graphs for the specified testing model(s): "Additive", "Dominant", "Recessive", "2df".

Value
A series of plots with sample size on the Y axis.

Examples
ss <- ss.calc(power=0.8, Case.Rate=c(0.5), k=NULL,
               MAF=seq(0.01, 0.05, 0.01), OR=c(4),Alpha=c(0.05),
               True.Model='All', Test.Model='All')
ss.plot(data=ss, x='MAF',panel.by='OR')

ss_envir.calc  Function to Calculate Power for Logistic Models with Environment Interaction

Description
Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

Usage
ss_envir.calc(
    power = 0.8,
    Case.Rate = NULL,
    k = NULL,
    MAF = NULL,
    OR_G = NULL,
    OR_E = NULL,
    OR_GE = NULL,
    P_e = NULL,
    Alpha = 0.05,
    True.Model = "All",
    Test.Model = "All"
)
ss_envir.calc.linear_outcome

Function to Calculate Power for Linear Models with logistic environment interaction

Description

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

Arguments

- **power**: Vector of the desired power(s)
- **Case.Rate**: proportion of cases in the sample (cases/(cases + controls)).
- **k**: Vector of the number of controls per case. Either k or Case.Rate must be specified.
- **MAF**: Vector of minor allele frequencies
- **OR_G**: Vector of genetic odds ratios to detect
- **OR_E**: Vector of environmental odds ratios to detect
- **OR_GE**: Vector of genetic/environmental interaction odds ratios to detect
- **P_e**: Vector of proportions of the population with exposure to the environmental effect
- **Alpha**: the desired type 1 error rate(s)
- **True.Model**: A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive1', 'Additive2', 'Recessive' or 'All'
- **Test.Model**: A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'

Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples

```r
ssc <- ss_envir.calc(P_e = 0.2, MAF = 0.1, power = 0.6, Case.Rate = 0.5, Alpha = 0.05, OR_G = 1.5, OR_E = 2, OR_GE = 1.8, Test.Model = "All", True.Model = "All")
```
ss_envir.calc.linear_outcome

Usage

ss_envir.calc.linear_outcome(
  pow = NULL,
  MAF = NULL,
  ES_G = NULL,
  ES_E = NULL,
  ES_GE = NULL,
  P_e = NULL,
  R2_G = NULL,
  R2_E = NULL,
  R2_GE = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)

Arguments

pow Vector of the desired power(s)
MAF Vector of minor allele frequencies
ES_G Vector of genetic effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_E Vector of environmental effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
ES_GE Vector of genetic/environment interaction effect sizes (difference in means) to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
P_e Vector of proportions of the population with exposure to the environmental effect
R2_G Vector of genetic R-squared values to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
R2_E Vector of environmental R-squared values to detect. Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
R2_GE Vector of genetic/environment interaction R-squared values Either ES_G, ES_E, and ES_EG or R2_G, R2_E, and R2_EG must be specified.
sd_y Standard deviation of the outcome in the population (ignoring genotype). Either sd_y_x or sd_y must be specified.
Alpha the desired type 1 error rate(s)
True.Model A vector specifying the true underlying genetic model(s): 'Dominant', 'Additive', 'Recessive' or 'All'
Test.Model A vector specifying the assumed genetic model(s) used in testing: 'Dominant', 'Additive', 'Recessive' or 'All'
Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES, Power, etc)

Examples

```r
ss_envir.calc.linear_outcome(pow=0.8, ES_G = 1.2, ES_E = 1.3, ES_GE = 2, Alpha = 0.05, MAF = 0.1, P_e = 0.2, sd_y = 10, True.Model = "All", Test.Model = "All")
```

Description

Calculates the power to detect a difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

Usage

```r
ss_linear_envir.calc.linear_outcome(
  pow = NULL,
  MAF = NULL,
  ES_G = NULL,
  ES_E = NULL,
  ES_GE = NULL,
  sd_e = NULL,
  R2_G = NULL,
  R2_E = NULL,
  R2_GE = NULL,
  sd_y = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)
```

Arguments

- `pow`: Vector of the desired power(s)
- `MAF`: Vector of minor allele frequencies
- `ES_G`: Vector of genetic effect sizes (difference in means) to detect. Either `ES_G`, `ES_E`, and `ES_EG` or `R2_G`, `R2_E`, and `R2_EG` must be specified.
- `ES_E`: Vector of environmental effect sizes (difference in means) to detect. Either `ES_G`, `ES_E`, and `ES_EG` or `R2_G`, `R2_E`, and `R2_EG` must be specified.
**ss_linear_envir.calc.logistic_outcome**

Function to Calculate Sample Size for Linear Models with logistic environment interaction

**Description**

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha

---

**ss_linear_envir.calc.logistic_outcome**

**Description**

Calculates the power to detect an difference in means/effect size/regression coefficient, at a given sample size, N, with type 1 error rate, Alpha
ss_linear_envir.calc.logistic_outcome

Usage

ss_linear_envir.calc.logistic_outcome(
  power = NULL,
  MAF = NULL,
  OR_G = NULL,
  OR_E = NULL,
  OR_GE = NULL,
  sd_e = NULL,
  Case.Rate = NULL,
  k = NULL,
  Alpha = 0.05,
  True.Model = "All",
  Test.Model = "All"
)

Arguments

  power  Vector of the desired power(s)
  MAF    Vector of minor allele frequencies
  OR_G   Vector of genetic odds ratios to detect
  OR_E   Vector of environmental odds ratios to detect
  OR_GE  Vector of genetic/environmental interaction odds ratios to detect
  sd_e   Standard deviation of the environmental variable
  Case.Rate Standard deviation of the outcome in the population (ignoring genotype). Either
               Case.Rate_x or Case.Rate must be specified.
  k      Vector of the number of controls per case. Either k or Case.Rate must be speci-
         fied.
  Alpha  the desired type 1 error rate(s)
  True.Model A vector specifying the true underlying genetic model(s): 'Dominant', 'Addi-
               tive', 'Recessive' or 'All'
  Test.Model A vector specifying the assumed genetic model(s) used in testing: 'Dominant',
               'Additive', 'Recessive' or 'All'

Value

A data frame including the power for all combinations of the specified parameters (Case.Rate, ES,
Power, etc)

Examples

ss <- ss_linear_envir.calc.logistic_outcome(power=0.8,
  OR_G=1.1, OR_E=1.2, OR_GE=1.5,
  sd_e = 1, MAF=0.2, Case.Rate = 0.2,
  Alpha=0.05, True.Model="All", Test.Model="All")
X_mat_returner

Function to output X matrices used in calculation of MLE’s for linear outcome with logistic environment interaction

Description

Returns X matrices used in calculation of MLE’s for linear outcome with logistic environment interaction

Usage

X_mat_returner(mod, reduced = F)

Arguments

mod type of model
reduced logical, indicates whether the X matrix will be used for a reduced model

Value

A matrix to be used in MLE calculation for linear outcome with logistic environment interaction

Examples

X_mat_returner(mod = "Dominant")

X_mat_returner_lle

Function to output X matrices used in calculation of MLE’s for linear outcome with linear environment interaction

Description

Function to output X matrices used in calculation of MLE’s for linear outcome with linear environment interaction

Usage

X_mat_returner_lle(mod)

Arguments

mod type of model
Value

A probability vector to be used in MLE calculation for linear outcome with linear environment interaction

Examples

X_mat_returner_lle("Dominant")

zero_finder_nleqslv Zero finder

Description

Finds zeros of multinomial functions using the nleqslv package

Usage

zero_finder_nleqslv(
  afun,
  veclength,
  tol = 0.4,
  x.start.vals = NULL,
  upper.lim = Inf
)

Arguments

afun The function to find zeros
veclength The dimension of the system of equations
tol The range within which to set start values for the function to use to find zeros
x.start.vals Optional user defined start values
upper.lim to be used if there is to be an upper limit to the solution

Value

Predicted zeros of the given equation

Examples

afun <- function(x) {
  y <- numeric(2)
  y[2] <- exp(x[1]-1) + x[2]^3 - 1.1
  y
}

zero_finder_nleqslv(afun, veclength = 2)
Index

add.fun.t, 3
add.or.function, 4
additive.ll, 5
additive.ll.linear, 5
as.numeric2, 6
calc.like, 7
calc.like.linear, 7
calc.like.linear.log.envir.interaction, 8
df2.ll, 9
df2.ll.linear, 10
dom.fun.t, 11
dom.or.function, 11
dominant.ll, 12
dominant.ll.linear, 13
es.calc.linear, 14
expected_linear.ll, 15
expected.linear.ll.lin.env, 15
find.prob.dom, 16
find.prob.rec, 17
genpwr.calc, 17
integrand_funct_case, 20
integrand_funct_control, 20
linear.mles, 21
linear.mles.lin.envir.interaction, 22
linear.mles.lin.envir.interaction_reduced, 23
linear.mles.log.envir.interaction, 24
linear.outcome.lin.envir.interaction.sds, 25
linear.outcome.lin.envir.interaction.sds_reduced, 26
linear.outcome.log.envir.interaction.sds, 27
linear.sds, 28
ll.ge.logistic, 28
ll.ge.logistic.lin.envir, 29
ll.linear.selector, 30
ll_zero_finder2, 31
logistic.mles, 32
logit, 32
ncp.search, 33
null.ll, 34
null.ll.linear, 34
odds_ratio_function, 35
or.function.2df, 36
or.plot, 37
or_calc, 38
p_vec_returner, 48
p_vec_returner_lin_env, 49
power.calc, 37, 39, 42
power.calc.linear, 40
power.plot, 41
power_envir.calc, 43
power_envir.calc.linear_outcome, 44
power_linear_envir.calc.linear_outcome, 45
power_linear_envir.calc.logistic_outcome, 47
quad_roots, 49
rec.fun.t, 50
rec.or.function, 51
recessive.ll, 51
recessive.ll.linear, 52
solve_a, 53
ss.calc.linear, 55
ss.plot, 56
ss_envir.calc, 57
ss_envir.calc.linear_outcome, 58
ss_linear_envir.calc.linear_outcome,
   60
ss_linear_envir.calc.logistic_outcome,
   61

X_mat_returner, 63
X_mat_returner_lle, 63

zero_finder_nleqslv, 64