Package 'luca'

October 13, 2022

Title Likelihood Inference from Case-Control Data under Covariate Assumptions

Version 1.0-6

Author Ji-Hyung Shin <shin.jihyung@gmail.com>, Brad McNeney <mcneney@stat.sfu.ca>, Jinko Graham <jgraham@stat.sfu.ca>

Maintainer Ji-Hyung Shin <shin.jihyung@gmail.com>

Depends R (>= 2.0.0), survival, genetics

Description Likelihood inference under covariate assumptions (LUCA) in case-control studies of a rare disease assuming independence or simple dependence of genetic and non-genetic covariates.

License GPL-2

URL https://sfustatgen.github.io/research/luca.html

NeedsCompilation no

Repository CRAN

Date/Publication 2021-11-02 15:00:02 UTC

R topics documented:

luca 1
lucaDat 4
summary.luca 5

Index **7**

`luca` *Likelihood-based case-control inference Under Covariate Assumptions (LUCA)*

Description

In genetic association studies, there is increasing interest in understanding the joint effects of genetic and nongenetic factors. For rare diseases, the case-control study is the standard design and logistic regression is the standard method of inference. However, the power to detect statistical interaction is a concern, even with relatively large samples. LUCA implements maximum likelihood inference under

1. independence of the genetic factor and nongenetic attributes in the control population,
2. independence of the genetic factor and nongenetic attributes, plus Hardy-Weinberg proportions (HWP) in control genotype frequencies, or
3. simple dependence between the genetic and nongenetic covariates in the control population.

Maximum likelihood under covariate assumptions offers improved precision of interaction estimators compared to the standard logistic regression approach which makes no assumptions on the distribution of covariates.

Usage

```
luca(pen.model, gLabel, dat, HWP = FALSE, dep.model = NULL)
```

Arguments

`pen.model`	an R formula specifying the disease penetrance model relating a genetic factor and a number of nongenetic attributes (the predictors or transformations thereof) to disease status. A typical `pen.model` has the form `d ~ g + a + g:a` where `d` is a binary disease response, `g` is a genetic factor, `a` is a (possibly continuous) nongenetic factor and `g:a` is the interaction between the genetic and nongenetic factors.
`gLabel`	a character string specifying the name of the genetic factor in `pen.model`.
`dat`	a data frame containing the variables in `pen.model`, currently, with *no* default value. Each row of `dat` is considered as one multivariate observation for a subject. Note that the genetic term must be a `factor` object, and also needs to be a `genotype` object in some cases (as described in the following arguments). Currently, the disease response variable must be *numeric* with values 0 (unaffected) and 1 (affected). Also, note that missing values are not allowed in the data frame.
`HWP`	a logical value indicating whether the genotype frequencies in controls should be assumed to follow Hardy-Weinberg proportions. When `TRUE`, the genetic term must be a `genotype` object.
`dep.model`	an R formula specifying the dependence between the genetic factor and nongenetic attributes. (See the Details section below for more on the dependence model.) When `NULL` (default), it indicates independence between the genetic factor and nongenetic attributes in controls. The argument `HWP` is ignored for a *non-null* `dep.model`. The genetic factor must be a `genotype` object when `dep.model` is provided.

Details

Inference for association parameters is obtained by fitting a conditional logistic regression model with appropriate match-sets comprised of "pseudo-individuals" having all possible values of the genetic factor and disease status but common value of the nongenetic attribute. The function `coxph.fit` from the `survival` package is used to fit the conditional logistic regression.

A dependence model such as `g ~ a` specifies a polychotomous regression model for the genetic factor `g` as a function of the nongenetic attribute `a`. The polychotomous regression for `g` given `a` holds when the conditional distribution of `a` given `g` is from the exponential family of distributions, with a constant dispersion parameter across the levels of the genetic factor. Alternately, `g` and `a` may be conditionally independent given a third variable `a2`. Typically, `a2` is also a term in the penetrance model (`pen.model`). To model conditional independence of `g` and `a` given `a2`, specify the dependence model (`dep.model`) as `g ~ a2`. See Shin, McNeney and Graham (2007) for details. `luca` also allows dependence models of the form `g ~ a1 + a2 + ...` for multiple attributes `a1`, `a2`, ... However, there is no formal justification for the use of such a model to capture the dependence between `g` and multiple nongenetic attributes.

Value

An object of class `"luca"` with the following components:

`call`	the function call
`coefficients`	estimates of parameters in the covariate model (lebelled as `covmod.XX`) and the penetrance model (labelled as `penmod.YY` where `YY` denotes the name of a term in the model). The covariate model parameters depend on the covariate assumptions and are 1) control-population log-odds for each level of the genetic factor relative to a baseline level under independence, 2) control-population log-odds for each allele relative to a baseline allele under independence plus HWP, or 3) the parameters from the polychotomous regression model under dependence (see the Details section for a description of this model).
`var`	the variance-covariance matrix of the parameter estimates.
`iter`	number of iterations in the iterative search for parameter estimates

The function `summary.luca` (or `summary`) can be used to obtain a summary of the results in a similar style to the `lm` and `glm` summaries.

Warning

Inference is not robust to misspecification of the covariate assumptions. There should be strong *a priori* evidence to support any assumptions that are made. Alternately, `luca` may be used to screen for "interesting" interactions that are followed up with logistic regression using data from a larger study.

Author(s)

Ji-Hyung Shin, Brad McNeney, Jinko Graham

References

Shin J-H, McNeney B, Graham J (2007). Case-Control Inference of Interaction between Genetic and Nongenetic Risk Factors under Assumptions on Their Distribution. Statistical Applications in Genetics and Molecular Biology 6(1), Article 13. Available at: http://www.bepress.com/sagmb/vol6/iss1/art13.

See Also

summary.luca, glm, coxph, clogit

Examples

```
data(lucaDat)
# typical penetrance model:
pen.model<-formula(d~I(allele.count(g,"C"))+a+a2+I(allele.count(g,"C")):a)

#1. Assuming independence and HWP
fitHWP<-luca(pen.model=pen.model, gLabel="g", dat=lucaDat, HWP=TRUE)
fitHWP$coef
fitHWP$var
summary.luca(fitHWP) # OR 'summary(fitHWP)'

#2. Assuming independence only
fitDefault<-luca(pen.model=pen.model, gLabel="g", dat=lucaDat)
fitDefault$coef
fitDefault$var

#3. Allowing for dependence between genetic and nongenetic factors

# General dependence model
fitDep1<-luca(pen.model=pen.model, gLabel="g", dat=lucaDat,
 dep.model=formula(g~a))
fitDep1$coef
fitDep1$var

# When 'g' and 'a' are conditioanally independent given the third variable 'a2':
fitDep2<-luca(pen.model=pen.model, gLabel="g", dat=lucaDat,
 dep.model=formula(g~a2))
fitDep2$coef
fitDep2$var
```

---

lucaDat *Simulated data for a hypothetical binary trait*

Description

Simulated data used to illustrate the luca package.

Usage

```
data(lucaDat)
```

Format

A data frame with 1000 observations on the following 4 variables:

[,1]	d	numeric	disease status (1=yes, 0=no)
[,2]	g	factor	genetic factor with levels `A/A`, `A/C`, `C/C`
[,3]	a	numeric	first non-genetic attribute
[,4]	a2	numeric	second non-genetic attribute

Examples

```
data(lucaDat)
```

`summary.luca` *Summarize results of the luca function*

Description

Summary function for reporting the results of the luca function in a similar style to the lm and glm summaries.

Usage

```
## S3 method for class 'luca'
summary(object, ...)
```

Arguments

`object`	a list of class `luca` output by the `luca` function
`...`	additional arguments to the summary function (currently unused)

Value

`call`	function call
`coefficients`	Table of estimated coefficients, standard errors and Wald tests for each variable

References

Shin J-H, McNeney B, Graham J (2007). Case-Control Inference of Interaction between Genetic and Nongenetic Risk Factors under Assumptions on Their Distribution. Statistical Applications in Genetics and Molecular Biology 6(1), Article 13. Available at: http://www.bepress.com/sagmb/vol6/iss1/art13.

See Also

`luca`

Examples

```
data(lucaDat)
pen.model <- formula(d ~ I(allele.count(g, "C")) +
    a + a2 + I(allele.count(g, "C")):a2)
fitDep <- luca(pen.model = pen.model, gLabel = "g",
    dat = lucaDat, dep.model = formula(g ~ a))
# Summarize the results:
summary.luca(fitDep) # or just summary(fitDep)
#Returns:
#Call:
#luca(dat = lucaDat, pen.model = pen.model, gLabel = "g", dep.model =
#formula(g ~ a))
#
#Coefficients:
#                              Estimate Std. Error     zscore     Pr(>|z|)
#I(allele.count(g, "C"))     0.61738385 0.10820323  5.7057800 1.158115e-08
#a                           0.11629696 0.07815014  1.4881222 1.367187e-01
#a2                         -0.03087368 0.10787965 -0.2861863 7.747354e-01
#I(allele.count(g, "C")):a2  0.31879401 0.08236130  3.8706772 1.085334e-04
```

Index

∗ **datasets**
lucaDat, 4
∗ **methods**
luca, 1
summary.luca, 5

clogit, *4*
coxph, *4*
coxph.fit, *3*

factor, *2*

genotype, *2*
glm, *4*

luca, 1, *5*
lucaDat, 4

summary.luca, *3, 4*, 5