# Package 'gcatest'

December 18, 2024

**Title** Genotype Conditional Association TEST

**Version** 2.7.0

**Encoding** UTF-8

**LazyData** true

**Description**
GCAT is an association test for genome wide association studies that controls for population structure under a general class of trait models. This test conditions on the trait, which makes it immune to confounding by unmodeled environmental factors. Population structure is modeled via logistic factors, which are estimated using the `lfa` package.

**Imports** methods, lfa

**Depends** R (>= 4.0)

**Suggests** knitr, ggplot2, testthat, BEDMatrix, genio

**VignetteBuilder** knitr

**License** GPL (>= 3)

**biocViews** SNP, DimensionReduction, PrincipalComponent, GenomeWideAssociation

**BugReports** https://github.com/StoreyLab/gcatest/issues

**URL** https://github.com/StoreyLab/gcatest

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**git_url** https://git.bioconductor.org/packages/gcatest

**git_branch** devel

**git_last_commit** 98df64e

**git_last_commit_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-18

**Author** Wei Hao [aut],
Minsun Song [aut],
Alejandro Ochoa [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4928-3403>),
John D. Storey [aut] (ORCID: <https://orcid.org/0000-0001-5992-402X>)

**Maintainer** Alejandro Ochoa <alejandro.ochoa@duke.edu>

# Contents

delta_deviance_lf . . . . . 2
gcat . . . . . 3
sim_geno . . . . . 4
sim_trait . . . . . 5

**Index** **6**

---

`delta_deviance_lf` *Calculate delta deviance of logistic null/alternative models*

---

## Description

This function fits, at each locus of a given genotype matrix, two logistic models, and under the assumption that the models are nested, calculates the delta deviance between the two. This general function is intended for testing models in a broad setting; for the specific problem of genetic association, the interface in `gcat()` and `gcat.stat()` are more user-friendly.

## Usage

```
delta_deviance_lf(X, LF0, LF1)
```

## Arguments

| | |
|---|---|
| `X` | A matrix of SNP genotypes, i.e. an integer matrix of 0's, 1's, 2's and `NA`s. BEDMatrix is supported. Sparse matrices of class Matrix are not supported (yet). |
| `LF0` | Logistic factors for null model. |
| `LF1` | Logistic factors for alternative model. |

## Value

The vector of delta deviance values, one per locus of `X`.

## Examples

```
library(lfa)

# make example data smaller so example is fast
# goes from 1000 to 100 individuals
indexes <- sample.int( ncol(sim_geno), 100 )
sim_geno <- sim_geno[ , indexes ]
sim_trait <- sim_trait[ indexes ]

# now run LFA and get delta deviances for trait assoc
# (recapitulating `gcat.stat` in this case)
LF <- lfa(sim_geno, 3)
LF0 <- LF # structure is null
LF1 <- cbind(LF, sim_trait) # trait is alt
devdiff_assoc <- delta_deviance_lf(sim_geno, LF0, LF1)

# can instead do delta deviances for structure only
LF0 <- cbind(rep.int(1, ncol(sim_geno))) # intercept only is null
LF1 <- LF # structure is alt, no trait
devdiff_struc <- delta_deviance_lf(sim_geno, LF0, LF1)
```

---

# gcat *Genotype Conditional Association TEST*

---

## Description

Performs the GCAT association test between SNPs and trait, returning p-values.

## Usage

```
gcat(X, LF, trait, adjustment = NULL)

gcatest(X, LF, trait, adjustment = NULL)

gcat.stat(X, LF, trait, adjustment = NULL)
```

## Arguments

| | |
|---|---|
| X | A matrix of SNP genotypes, i.e. an integer matrix of 0's, 1's, 2's and NAs. BEDMatrix is supported. Sparse matrices of class Matrix are not supported (yet). |
| LF | matrix of logistic factors from `lfa::lfa()` |
| trait | vector |
| adjustment | matrix of adjustment variables |

## Value

vector of p-values

## Functions

- `gcatest()`: Alias of gcat
- `gcat.stat()`: returns the association statistics instead of the p-value.

## References

Song, M, Hao, W, Storey, JD (2015). Testing for genetic associations in arbitrarily structured populations. Nat. Genet., 47, 5:550-4.

## Examples

```
library(lfa)

# make example data smaller so example is fast
# goes from 1000 to 100 individuals
indexes <- sample.int( ncol(sim_geno), 100 )
sim_geno <- sim_geno[ , indexes ]
sim_trait <- sim_trait[ indexes ]

# now run LFA and GCATest
LF <- lfa(sim_geno, 3)
gcat_p <- gcat(sim_geno, LF, sim_trait)
gcat_stat <- gcat.stat(sim_geno, LF, sim_trait)
```

---

# sim_geno *Simulated data from PSD model*

---

## Description

10,000 SNPs, 1,000 individuals, first five SNPs are associated.

## Usage

```
sim_geno
```

## Format

a matrix of 0's, 1's and 2's for the genotypes

## Value

simulated genotype matrix

`sim_trait` *Simulated data from PSD model*

## Description

10,000 SNPs, 1,000 individuals, first five SNPs are associated.

## Usage

```
sim_trait
```

## Format

a vector of traits

## Value

simulated traits

# Index

delta_deviance_lf, 2

gcat, 3
gcat(), *2*
gcat.stat(), *2*
gcatest *(*gcat*)*, 3

lfa::lfa(), *3*

sim_geno, 4
sim_trait, 5