Package: diffdriver (via r-universe)

October 9, 2026

Type Package

Title Identify differential selection

Version 0.1.8

Maintainer Siming Zhao <siming.zhao@dartmouth.edu>

Description This package implements a statistical method to detect differential selection of somatic mutations under different environments

License MIT

Encoding UTF-8

LazyData true

Imports Matrix, stats, data.table, plyr, brglm, fastTopics, matrixStats

Suggests knitr, rmarkdown, logging, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

Repository https://szhaolab.r-universe.dev

Date/Publication 2026-07-11 20:23:49 UTC

RemoteUrl https://github.com/szhaolab/diffdriver

RemoteRef HEAD

RemoteSha a47b7dd99b0f1c04e15b5eb2aadaa8d9feea90ad

Contents

ddmcode 2
ddmodel 2
ddmodel_binary 3

ddmodel_binary_simple . . . 3
diffdriver . . . 4
genebinom . . . 6
genefisher . . . 6
genelr . . . 6
mlr . . . 6
mlr.v2 . . . 7
optifix . . . 7
plot_mut . . . 8

Index **9**

`ddmcode` *Add intercept, if have functtypecode, then code and move to the front. different from driverMAPS! only allows functtypecode =7 ||8 when functypecode is included in selectvars.*

Description

Add intercept, if have functtypecode, then code and move to the front. different from driverMAPS! only allows functtypecode =7 ||8 when functypecode is included in selectvars.

Usage

```
ddmcode(matrixlist, selectvars, functypecodelevel = NULL)
```

`ddmodel` *diffDriver model*

Description

This model is applied on data of a single gene. It will infer effect size for both sample-level variable and positional level functional annotations. We used an EM algorithm to infer parameters.

Usage

```
ddmodel(mut, e, mr, fe, label, cov = NULL, ...)
```

Arguments

`mut`	a matrix of mutation status 0 or 1, rows positions, columns are samples.
`e`	a vector,phenotype of each sample, should match the columns of `mut` and `mr`
`mr`	a matrix, mutation rate of each sample at each mutation (log scale) that is not dependent on sample level factor
`fe`	a vector, increased mutation rate at each position, depending on e (log scale), should match the rows of `mut` and `mr`

cov	an optional numeric matrix of sample-level covariates (e.g. age, sex), with rows matching the columns (samples) of mut. When supplied, the covariates are added to the logistic regression of the EM responsibilities under both the null and alternative models, so the phenotype effect is tested conditional on the covariates. Default NULL (no covariates).

---

ddmodel_binary *diffDriver model only for binary phenotype.*

Description

This function uses the model as cmodel.frac, but generalizes to take more than 1 functional categories. This model is applied on data of a single gene

Usage

```
ddmodel_binary(mut, e, bmr, fe)
```

Arguments

mut	a matrix of mutation status 0 or 1
e	a vector,phenotype of each sample, should match the columns of mut and bmr
bmr	a matrix, background mutation rate of each sample at each mutation (log scale)
fe	a vector, increased mutation rate at each mutation, due to functional effect (log scale), should match the rows of mut and bmr

---

ddmodel_binary_simple *diffDriver model only for binary phenotype, assuming bmr the same across samples.*

Description

This function uses the model as cmodel.frac, but generalizes to take more than 1 functional categories. This model is applied on data of a single gene. This should give the same results as ddmodel_binary defined above.

Usage

```
ddmodel_binary_simple(mut, e, bmr, fe)
```

Arguments

mut	a matrix of mutation status 0 or 1
e	a vector,phenotype of each sample, should match the columns of mut and bmr
bmr	a matrix, background mutation rate of each sample at each mutation (log scale), as we assume bmr the same across samples, only the first column will be used.
fe	a vector, increased mutation rate at each mutation, due to functional effect (log scale), should match the rows of mut and bmr

---

diffdriver *Run diffDriver with Input Files*

Description

This function runs diffDriver.

Usage

```
diffdriver(
  gene,
  mut,
  pheno,
  covariates = NULL,
  anno_dir = ".",
  k = 6,
  totalnttype = 96,
  BMRmode = c("signature", "regular"),
  output_dir = ".",
  output_prefix = "diffdriver_results"
)
```

Arguments

gene	A vector of genes to be included in the analysis.

mut	A data frame containing all somatic mutations from the cohort. The format is:
	Chromosome <int>
	Position <int>
	Ref <chr>
	Alt <chr>
	SampleID <chr>
	Example:
	`Chromosome Position Ref Alt SampleID`
	`1 19 55653236 C T TCGA-N6-A4VE-01A-11D-A28R-08`

pheno	A data frame containing sample phenotypes. The format is: #'

SampleID <chr>
Phenotype1 <dbl>
Phenotype2 <dbl>
... ...

Example:

```
SampleID SmokingCessation BMI
TCGA-N5-A4R8-01A-11D-A28R-08 0.5319630 20.0
TCGA-N5-A4RD-01A-11D-A28R-08 0.0448991 24.4
```

`covariates` Optional sample-level covariates (e.g. age, sex) to adjust for. Either a path to a delimited file or a data frame. It must contain a `SampleID` column plus one or more covariate columns, which may be numeric or categorical (categorical columns are expanded into dummy variables). Only samples with complete covariate information are kept, and these are intersected with the mutation and phenotype samples. Any covariate that is single-level (has no variance) among the retained samples is dropped with a warning, since it carries no information; if no covariate with variance remains, the analysis proceeds without covariate adjustment. When supplied, the covariates are added to the regression inside `ddmodel`/`ddmodel_nl` under both the null and alternative models, so the phenotype effect is tested conditional on the covariates. The format is:

SampleID <chr>
Covariate1 <dbl> or <chr>
Covariate2 <dbl> or <chr>
... ...

Example:

```
SampleID Age Sex
TCGA-N5-A4R8-01A-11D-A28R-08 56 M
TCGA-N5-A4RD-01A-11D-A28R-08 63 F
```

The default is `NULL` (no covariates).

`anno_dir` The path to the directory with all the annotation files. Please download from Zenodo.The default is current folder

`k` The number of topics used in modeling background mutation rate. The default is 6.

`totalnttype` either 9 or 96. Will look for annotation files anno9_ntypexxx_annodata.txt when totalnttype is 9 or anno96_ntypexxx_annodata when totalnttype is 96.

`BMRmode` There are two modes to run diffdriver. One is "signature", this will model individual level BMR, this is the default. The second one is "regular", this assumes BMR is the same across individuals, only models position-level difference.

`output_dir` The path to output directory

`output_prefix` The prefix being added to the output file names.

genebinom *gene level binomial test*

Description

gene level binomial test

Usage

```
genebinom(mut, e)
```

genefisher *gene level fisher's exact test*

Description

gene level fisher's exact test

Usage

```
genefisher(mut, e)
```

genelr *gene level logistic regression*

Description

gene level logistic regression

Usage

```
genelr(mut, e, covariates = rep(1, length(e)))
```

mlr *gene level multiple linear regression*

Description

gene level multiple linear regression

Usage

```
mlr(mut, e, covariates = rep(1, length(e)))
```

mlr.v2 *gene level multiple linear regression, correcting for total number of mutations*

Description

gene level multiple linear regression, correcting for total number of mutations

Usage

```
mlr.v2(mut, e, nmut, covariates = 1)
```

optifix *optifix. Optimise with fixed parameters*

Description

its like optim, but with fixed parameters.

Usage

```
optifix(
  par,
  fixed,
  fn,
  gr = NULL,
  ...,
  method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN"),
  lower = -Inf,
  upper = Inf,
  control = list(),
  hessian = FALSE
)
```

Details

specify a second argument 'fixed', a vector of TRUE/FALSE values. If TRUE, the corresponding parameter in fn() is fixed. Otherwise its variable and optimised over.

The return thing is the return thing from optim() but with a couple of extra bits - a vector of all the parameters and a vector copy of the 'fixed' argument.

Written by Barry Rowlingson <b.rowlingson@lancaster.ac.uk> October 2011

This file released under a CC By-SA license: http://creativecommons.org/licenses/by-sa/3.0/

and must retain the text: "Originally written by Barry Rowlingson" in comments.

plot_mut *plot phenotype, mutation and annotation for a gene across samples*

Description

plot phenotype, mutation and annotation for a gene across samples

Usage

```
plot_mut(
  gene_name,
  mut,
  pheno,
  totalnttype = 96,
  anno_dir = ".",
  output_prefix = "plot",
  output_dir = "."
)
```

Index

ddmcode, 2
ddmodel, 2
ddmodel_binary, 3
ddmodel_binary_simple, 3
diffdriver, 4

genebinom, 6
genefisher, 6
genelr, 6

mlr, 6
mlr.v2, 7

optifix, 7

plot_mut, 8