

# Package: diffdriver (via r-universe)

June 27, 2026

**Type** Package

**Title** Identify differential selection

**Version** 0.1.7

**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** 7.2.3

**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** 2026-06-23 05:07:44 UTC

**RemoteUrl** <https://github.com/szhaolab/diffdriver>

**RemoteRef** HEAD

**RemoteSha** 8eb3b0f8206ebcad602cd89453eaa5221202701

## Contents

ddmcode . . . . .	2
ddmodel . . . . .	2
ddmodel_binary . . . . .	3

ddmodel_binary_simple . . . . .	3
diffdriver . . . . .	4
genebinom . . . . .	5
genefisher . . . . .	5
genelr . . . . .	5
mlr . . . . .	6
mlr.v2 . . . . .	6
optifix . . . . .	6
plot_mut . . . . .	7

## Index 8

---

ddmcode	<i>Add intercept, if have functypecode, then code and move to the front. different from driverMAPS! only allows functypecode =7   8 when functypecode is included in selectvars.</i>
---------	--

---

### Description

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

### Usage

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

---

ddmodel	<i>diffDriver model</i>
---------	-------------------------

---

### 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, ...)
```

### 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

---

ddmodel_binary	<i>diffDriver model only for binary phenotype.</i>
----------------	--

---

**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	<i>diffDriver model only for binary phenotype, assuming bmr the same across samples.</i>
-----------------------	--

---

**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,
  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: <b>Chromosome</b> <int> <b>Position</b> <int> <b>Ref</b> <chr> <b>Alt</b> <chr> <b>SampleID</b> <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: #' <b>SampleID</b> <chr> <b>Phenotype1</b> <dbl> <b>Phenotype2</b> <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

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	<i>gene level multiple linear regression</i>
-----	--

---

**Description**

gene level multiple linear regression

**Usage**

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

---

mlr.v2	<i>gene level multiple linear regression, correcting for total number of mutations</i>
--------	--

---

**Description**

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

**Usage**

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

---

optifix	<i>optifix. Optimise with fixed parameters</i>
---------	--

---

**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`, 5  
`genefisher`, 5  
`genelr`, 5  
  
`mlr`, 6  
`mlr.v2`, 6  
  
`optifix`, 6  
  
`plot_mut`, 7