

# Package ‘gemma2’

October 1, 2019

**Title** GEMMA Multivariate Linear Mixed Model

**Version** 0.1.1

**Description** Fits a multivariate linear mixed effects model that uses a polygenic term, after Zhou & Stephens (2014) (<<https://www.nature.com/articles/nmeth.2848>>). Of particular interest is the estimation of variance components with restricted maximum likelihood (REML) methods. Genome-wide efficient mixed-model association (GEMMA), as implemented in the package 'gemma2', uses an expectation-maximization algorithm for variance components inference for use in quantitative trait locus studies.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/fboehm/gemma2>

**BugReports** <https://github.com/fboehm/gemma2/issues>

**Suggests** covr, testthat, knitr, rmarkdown

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**Imports** readr, Matrix

**Language** en-US

**NeedsCompilation** no

**Author** Frederick Boehm [aut, cre] (<<https://orcid.org/0000-0002-1644-5931>>)

**Maintainer** Frederick Boehm <[frederick.boehm@gmail.com](mailto:frederick.boehm@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-10-01 12:00:11 UTC

## R topics documented:

calc_omega . . . . .	2
calc_qi . . . . .	3
calc_sigma . . . . .	4

calc_XHiY . . . . .	4
center_kinship . . . . .	5
eigen2 . . . . .	6
eigen_proc . . . . .	6
gemma2 . . . . .	7
MphCalcLogL . . . . .	7
MphEM . . . . .	8
stagger_mats . . . . .	8
UpdateRL_B . . . . .	9
update_e . . . . .	9
update_u . . . . .	10
update_v . . . . .	11
<b>Index</b>	<b>12</b>

---

calc_omega	<i>Calculate Omega matrices</i>
------------	---------------------------------

---

### Description

Calculate Omega matrices

### Usage

```
calc_omega(eval, D_1)
```

### Arguments

eval	vector of eigenvalues from decomposition of relatedness matrix
D_1	vector of length d_size

### Value

list of length 2. First entry in the list is the symmetric matrix OmegaU. Second entry in the list is the symmetric matrix OmegaE.

### Examples

```
calc_omega(eval = 50:1, D_1 = runif(2))
```

---

calc_qi	<i>Calculate Qi (inverse of Q) and log determinant of Q</i>
---------	---

---

## Description

Calculate Qi (inverse of Q) and log determinant of Q

## Usage

```
calc_qi(eval, D_l, X)
```

## Arguments

eval	vector of eigenvalues from decomposition of relatedness matrix
D_l	vector of length d_size
X	design matrix

## Value

a list of length two. First entry in the list is a symmetric numeric matrix, Qi, the inverse of the Q matrix. The second entry in the outputted list is the log determinant of the matrix Q for use in likelihood calculations.

## Examples

```
as.matrix(readr::read_tsv(system.file("extdata",  
"mouse100.cXX.txt",  
package = "gemma2"),  
col_names = FALSE)[, 1:100]) -> kinship  
eigen2(kinship) -> e2_out  
e2_out$values -> eval  
e2_out$vectors -> U  
eigen_proc(V_g = diag(c(1.91352, 0.530827)),  
V_e = diag(c(0.320028, 0.561589))) -> ep_out  
  
calc_qi(eval = eval,  
D_l = ep_out[[4]],  
X = t(rep(1, 100)) %*% U)
```

---

calc_sigma	<i>Calculate Sigma_ee and Sigma_uu matrices</i>
------------	---

---

**Description**

Calculate Sigma\_ee and Sigma\_uu matrices

**Usage**

```
calc_sigma(eval, D_l, X, OmegaU, OmegaE, UltVeh, Qi)
```

**Arguments**

eval	eigenvalues vector from decomposition of relatedness matrix
D_l	vector
X	design matrix
OmegaU	matrix
OmegaE	matrix
UltVeh	matrix
Qi	inverse of Q matrix

---

calc_XHiY	<i>Calculate XHiY</i>
-----------	-----------------------

---

**Description**

Calculate XHiY

**Usage**

```
calc_XHiY(eval, D_l, X, UltVehiY)
```

**Arguments**

eval	vector of eigenvalues from the decomposition of the relatedness matrix
D_l	vector of length d_size
X	design matrix
UltVehiY	a matrix

**Value**

numeric vector

**Examples**

```

readr::read_tsv(system.file("extdata",
"mouse100.pheno.txt",
package = "gemma2"),
col_names = FALSE) -> pheno
phe16 <- as.matrix(pheno[, c(1, 6)])
as.matrix(readr::read_tsv(system.file("extdata",
"mouse100.cXX.txt",
package = "gemma2"),
col_names = FALSE)[, 1:100]) -> kinship
eigen2(kinship) -> eout
eout$values -> eval
eout$vectors -> U
UltVehi <- matrix(c(0, -1.76769, -1.334414, 0),
nrow = 2,
byrow = FALSE) # from output of eigen_proc()
calc_XHiY(eval = eval,
UltVehiY = UltVehi %*% t(phe16) %*% U
)

```

---

center_kinship	<i>Center a relatedness matrix, after Zhou's GEMMA function CenterMatrix</i>
----------------	--

---

**Description**

Center a relatedness matrix, after Zhou's GEMMA function CenterMatrix

**Usage**

```
center_kinship(mat)
```

**Arguments**

mat                    a relatedness matrix

**Value**

a centered relatedness matrix

**Examples**

```

readr::read_tsv(system.file("extdata",
"mouse100.cXX.txt",
package = "gemma2"),
col_names = FALSE)[, 1:100] -> kinship
e_out <- eigen2(as.matrix(kinship))
center_kinship(as.matrix(kinship)) -> kinship_centered

```

---

eigen2	<i>Calculate eigendecomposition and return ordered eigenvalues and eigenvectors</i>
--------	---

---

**Description**

Calculate eigendecomposition and return ordered eigenvalues and eigenvectors

**Usage**

```
eigen2(spd, decreasing = FALSE)
```

**Arguments**

spd	a semi-positive definite matrix
decreasing	argument passed to order()

**Value**

a list with 2 components, the eigenvalues and the eigenvectors

**Examples**

```
readr::read_tsv(system.file("extdata",
"mouse100.cXX.txt",
package = "gemma2"),
col_names = FALSE)[, 1:100] -> kinship
e_out <- eigen2(as.matrix(kinship))
```

---

eigen_proc	<i>Eigendecomposition procedure for Vg and Ve</i>
------------	---

---

**Description**

Eigendecomposition procedure for Vg and Ve

**Usage**

```
eigen_proc(V_g, V_e, tol = 1/10000)
```

**Arguments**

V_g	a d_size by d_size covariance matrix
V_e	a d_size by d_size covariance matrix
tol	a positive number indicating the tolerance for isSymmetric

**Value**

a named list of length 4 containing the outputs of eigendecomposition procedure

**Examples**

```
eigen_proc(diag(2), diag(2))
```

---

<code>gemma2</code>	<i>gemma2.</i>
---------------------	----------------

---

**Description**

`gemma2.`

---

<code>MphCalcLogL</code>	<i>Calculate log likelihood</i>
--------------------------	---------------------------------

---

**Description**

Calculate log likelihood

**Usage**

```
MphCalcLogL(eval, D_l, Qi, UltVehiY, xHiy)
```

**Arguments**

- `eval` eigenvalues vector from decomposition of relatedness matrix
- `D_l` vector of eigenvalues from decomposition of  $V_e$  matrix
- `Qi` inverse of Q matrix
- `UltVehiY` matrix of (transformed) Y values
- `xHiy` vector

---

MphEM	<i>Perform expectation-maximization algorithm to infer <math>V_g</math> and <math>V_e</math> values for a pair of traits.</i>
-------	---

---

### Description

Perform expectation-maximization algorithm to infer  $V_g$  and  $V_e$  values for a pair of traits.

### Usage

```
MphEM(max_iter = 10000, max_prec = 1/1e+06, eval, X, Y, V_g, V_e,
       verbose_output = FALSE)
```

### Arguments

max_iter	maximum number of iterations for EM algorithm
max_prec	maximum precision for EM algorithm
eval	vector of eigenvalues from relatedness matrix decomposition
X	design matrix. Typically contains founder allele dosages.
Y	matrix of phenotype values
V_g	genetic covariance matrix
V_e	error covariance matrix
verbose_output	logical indicating whether to output entire collection of intermediate values for all iterations. Default is FALSE.

### Value

a list of lists. Length of list corresponds to number of EM iterations

---

stagger_mats	<i>Stagger matrices within a larger, block-diagonal matrix</i>
--------------	--

---

### Description

Stagger matrices within a larger, block-diagonal matrix

### Usage

```
stagger_mats(...)
```

### Arguments

...	one or more matrices, separated by commas
-----	---



**Value**

a block-diagonal matrix, with the inputted matrices as blocks on the diagonal.

**Examples**

```
foo <- matrix(rnorm(40000), ncol = 8)
block_diag <- stagger_mats(foo, foo)
dim(foo)
dim(block_diag)
```

---

 UpdateRL\_B

*Update B for restricted log likelihood*


---

**Description**

Update B for restricted log likelihood

**Usage**

```
UpdateRL_B(xHiy, Qi, d_size)
```

**Arguments**

xHiy	vector
Qi	Q inverse matrix
d_size	number of traits

**See Also**

Other expectation-maximization functions: [update\\_e](#), [update\\_u](#), [update\\_v](#)

---

 update\_e

*Update E*


---

**Description**

Update E

**Usage**

```
update_e(UltVehiY, UltVehiBX, UltVehiU)
```

**Arguments**

UltVehiY	matrix of transformed Y values
UltVehiBX	matrix of transformed BX values
UltVehiU	matrix of transformed U values

**See Also**

Other expectation-maximization functions: [UpdateRL\\_B](#), [update\\_u](#), [update\\_v](#)

---

update_u	<i>Update U matrix</i>
----------	------------------------

---

**Description**

Update U matrix

**Usage**

```
update_u(OmegaE, UltVehiY, UltVehiBX)
```

**Arguments**

OmegaE	the OmegaE matrix, calculated in <code>calc_omega</code>
UltVehiY	matrix
UltVehiBX	matrix

**See Also**

Other expectation-maximization functions: [UpdateRL\\_B](#), [update\\_e](#), [update\\_v](#)

**Examples**

```
readr::read_tsv(system.file("extdata",
"mouse100.pheno.txt",
package = "gemma2"),
col_names = FALSE) -> pheno
phe16 <- as.matrix(pheno[, c(1, 6)])
as.matrix(readr::read_tsv(system.file("extdata",
"mouse100.cXX.txt",
package = "gemma2"),
col_names = FALSE)[, 1:100]) -> kinship
eigen2(kinship) -> e2_out
e2_out$values -> eval
e2_out$vectors -> U
eigen_proc(V_g = diag(c(1.91352, 0.530827)),
V_e = diag(c(0.320028, 0.561589))) -> ep_out
UltVehi <- ep_out[[3]]
```

```
calc_omega(eval, ep_out$D_1) -> co_out
update_u(OmegaE = co_out[[2]],
         UltVehiY = UltVehi %*% t(phe16),
         UltVehiBX = matrix(c(-0.71342, -0.824482),
                             ncol = 1) %*% t(rep(1, 100))
        )
```

---

update\_v

*Update V\_e and V\_g*

---

### Description

Update V\_e and V\_g

### Usage

```
update_v(eval, U, E, Sigma_uu, Sigma_ee, tol = 1/10000)
```

### Arguments

eval	vector of eigenvalues from eigendecomposition of relatedness matrix
U	matrix
E	matrix
Sigma_uu	matrix
Sigma_ee	matrix
tol	a positive number indicating tolerance to be passed to isSymmetric()

### See Also

Other expectation-maximization functions: [UpdateRL\\_B](#), [update\\_e](#), [update\\_u](#)

# Index

calc\_omega, 2  
calc\_qi, 3  
calc\_sigma, 4  
calc\_XHiY, 4  
center\_kinship, 5  
  
eigen2, 6  
eigen\_proc, 6  
  
gemma2, 7  
gemma2-package (gemma2), 7  
  
MphCalcLogL, 7  
MphEM, 8  
  
stagger\_mats, 8  
  
update\_e, 9, 9, 10, 11  
update\_u, 9, 10, 10, 11  
update\_v, 9, 10, 11  
UpdateRL\_B, 9, 10, 11