

# Package ‘gpmmap’

May 8, 2026

**Type** Package

**Title** Analysing and Plotting Genotype-Phenotype Maps

**Version** 0.1.3

**Date** 2025-11-27

**Depends** isotone, plyr, ggplot2, foreach

**Description** Tools for studying genotype-phenotype maps for bi-allelic loci underlying quantitative phenotypes. The 0.1 version is released in connection with the publication of Gjuvsland et al (2013) and implements basic line plots and the monotonicity measures for GP maps presented in the paper. Reference: Gjuvsland AB, Wang Y, Plahte E and Omholt SW (2013) Monotonicity is a key feature of genotype-phenotype maps. *Frontier in Genetics* 4:216 <[doi:10.3389/fgene.2013.00216](https://doi.org/10.3389/fgene.2013.00216)>.

**License** GPL-3

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-11-27 19:00:06 UTC

**Author** Arne B. Gjuvsland [aut, cre],  
Yunpeng Wang [aut]

**Maintainer** Arne B. Gjuvsland <[arne.gjuvsland@nmbu.no](mailto:arne.gjuvsland@nmbu.no)>

## Contents

gpmmap-package . . . . .	2
decompose_monotone . . . . .	3
degree_of_monotonicity . . . . .	4
enumerate_genotypes . . . . .	5
generate_gpmmap . . . . .	6
GPmaps . . . . .	8
monotone_regression . . . . .	8
partial_genotype_order . . . . .	9
plot.gpmmap . . . . .	10
print.gpmmap . . . . .	11

---

gpmmap-package

*Tools for analysing and plotting genotype-phenotype maps*

---

## Description

The **gpmmap** package deals with genotype-phenotype maps for biallelic loci underlying quantitative phenotypes. The package provides a class `gpmmaps`, analysis functions and basic lineplots. The package is designed for studying the properties of GP maps without reference to any particular population, i.e. the physiological (Cheverud & Routman, 1995) or functional (Hansen, 2001) properties of the GP map. This is opposed to statistical effects underlying most of quantitative genetics, where the GP-map is analysed together with genotype frequencies in a given population (e.g. Lynch & Walsh, 1998).

In version 0.1 which is released as part of the publication of Gjuvsland *et al.* (2013) we have implemented functionality for studying monotonicity Gjuvsland *et al.* (2011) of GP maps. The package utilizes the **isotone** package for monotone regression, and the **foreach** package for parallel computation.

The package consists of the following high-level functions : `enumerate_genotypes`, `generate_gpmmap`, `degree_of_monotonicity`, `decompose_monotone` and `plot.gpmmap`

## Author(s)

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

## References

- Cheverud JM & Routman EJ (1995) Epistasis and Its Contribution to Genetic Variance Components. *Genetics* 139:1455-1461 doi:[10.1093/genetics/139.3.1455](https://doi.org/10.1093/genetics/139.3.1455)
- Gjuvsland AB, Vik JO, Woolliams JA, Omholt SW (2011) Order-preserving principles underlying genotype-phenotype maps ensure high additive proportions of genetic variance. *Journal of Evolutionary Biology* 24(10):2269-2279 doi:[10.1111/j.14209101.2011.02358.x](https://doi.org/10.1111/j.14209101.2011.02358.x)
- Gjuvsland AB, Wang Y, Plahte E and Omholt SW (2013) Monotonicity is a key feature of genotype-phenotype maps. *Front. Genet.* 4:216. doi:[10.3389/fgene.2013.00216](https://doi.org/10.3389/fgene.2013.00216)
- Hansen T & Wagner GP (2001) Modeling genetic Architecture: A Multilinear Theory of gene Interaction. *Theoretical Population Biology* 59:61-86 doi:[10.1006/tpbi.2000.1508](https://doi.org/10.1006/tpbi.2000.1508)
- Leeuw J, Hornik K and Mair P (2009) Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. *Journal of Statistical Software* 32(5) [[link](#)]
- Lynch M & Walsh B (1998) *Genetics and Analysis of Quantitative Traits*, Sunderland, MA: Sinauer Associates

---

decompose_monotone	<i>Decompose genotype-phenotype map(s) using monotone regression</i>
--------------------	--

---

## Description

Functions for decomposing genotype-phenotype (GP) maps with  $N$  biallelic loci using monotone regression from the `isotone` package.

## Usage

```
decompose_monotone(gmap)  
decompose_monotone_single(gmap)
```

## Arguments

`gmap`            A `gmap` object

## Details

`decompose_monotone` works for any `gmap` object (values is  $(3^N) \times K$  matrix of genotypic values) and calls the internal function `decompose_monotone_single` for each column. `decompose_monotone_single` takes a `gmap` object with a single set of genotypic values ( $K = 1$ ), loops through all  $2^N$  possible combinations of plusalleles, calls `monotone_regression` and identifies the best fit. The code uses the `foreach` package and will run in parallel if a *parallel backend* is registered (see `foreach` documentation).

## Value

The input `gmap` is returned with two added elements

<code>monoR2</code>	The coefficient of determination of the monotone regression
<code>values.mono</code>	A matrix of genotypic values for the monotone component of genotype-phenotype map(s)

## Author(s)

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

## References

- Leeuw J, Hornik K and Mair P (2009) Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. *Journal of Statistical Software* 32(5) [[link](#)]
- Gjuvsland AB, Wang Y, Plahte E and Omholt SW (2013) Monotonicity is a key feature of genotype-phenotype maps. *Front. Genet.* 4:216. doi:10.3389/fgene.2013.00216

**Examples**

```

data(GPmaps)

#Additive GP map is monotone so monoR2=1 and values.mono=values
decompose_monotone(A)

#Pure AxA epistasis map
decompose_monotone(AA)

#two-locus example in Cheverud & Routman (1995)
decompose_monotone(mouseweight)

#decompose four random 3-locus GP maps
set.seed(0)
randomGP <- rnorm(3^2*4)
dim(randomGP) <- c(9,4)
decompose_monotone(generate_gpmap(randomGP))

```

---

degree\_of\_monotonicity

*Degree of monotonicity of GP map*

---

**Description**

Functions for computing degree of monotonicity  $m$  for [gpmap](#) objects.

**Usage**

```

degree_of_monotonicity(gpmap)
degree_of_monotonicity_single(gpmap)

```

**Arguments**

gpmap            A [gpmap](#) object

**Details**

degree\_of\_monotonicity works for any [gpmap](#) object (values is  $(3^N) \times K$  matrix of genotypic values) and calls the internal function degree\_of\_monotonicity\_single for each column. degree\_of\_monotonicity\_single computes substitution effect, locus weights and per-locus and overall degree of monotonicity as described in Gjuvsland *et al.* (2013).

**Value**

degree\_of\_monotonicity returns the input gpmap with the following added fields:

```

degree.monotonicity
  Overall degree of monotonicity for the  $K$  GP maps

```

```

degree.monotonicity.locus      Data frame with per locus degree of monotonicity for the  $K$  GP maps
locus.weight                   Data frame with locus weights

```

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**References**

Gjuvsland AB, Wang Y, Plahte E and Omholt SW (2013) Monotonicity is a key feature of genotype-phenotype maps. *Front. Genet.* 4:216. doi:10.3389/fgene.2013.00216

**Examples**

```

data(GPmaps)

#Additive GP map is monotone
degree_of_monotonicity(A)

#Pure AxA epistasis map
degree_of_monotonicity(AA)

#two-locus example in Cheverud & Routman (1995)
degree_of_monotonicity(mouseweight)

```

---

```

enumerate_genotypes      Function for enumerating genotypes for  $N$  biallelic loci

```

---

**Description**

Function for enumerating all  $3^N$  genotypes for  $N$  biallelic loci. Optional specification of names of loci and alleles. `genotypes`. Generates a data frame of multilocus genotypes in the sequence used for objects of class `gpmmap`.

**Usage**

```

enumerate_genotypes(nloci=1, locinames=NULL, allelenames=NULL)

```

**Arguments**

```

nloci           The number of loci  $N$ 
locinames       An optional character vector with  $N$  names of loci
allelenames     An optional character object specifying allele names

```

**Details**

Unless specified `locinames` default to "Locus 1", "Locus 2", ..., "Locus  $N$ ".  
 If `allelenames` is not specified then the alleles will be named "1" and "2".

**Value**

Returns a data frame with locinames as colnames, and with  $3^N$  rows specifying all possible genotypes in the sequence used for all GP maps in the package (the same sequence as used in Gjuvsland *et al.* (2011)), where the genotype at the first locus varies fastest, then the second locus, and so on:

	Locus_1	Locus_2
1	11	11
2	12	11
3	22	11
4	11	12
5	12	12
6	22	12
7	11	22
8	12	22
9	22	22

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**References**

Gjuvsland AB, Vik JO, Woolliams JA, Omholt SW (2011) Order-preserving principles underlying genotype-phenotype maps ensure high additive proportions of genetic variance. *Journal of Evolutionary Biology* 24(10):2269-2279 doi:10.1111/j.14209101.2011.02358.x

**Examples**

```
#genotypes for a single locus
enumerate_genotypes()

#genotypes for two loci "A" and "B", with alleles named "H" and "L"
enumerate_genotypes(2,c("A","B"),c("H","L"))

#genotypes for the two-locus example in Cheverud & Routman (1995)
enumerate_genotypes(2,c("D7Mit17","D1Mit7"),rbind(c('A1','A2'),c('B1','B2')))
```

---

generate_gpmap	<i>Function for creating genotype-phenotype (GP) maps</i>
----------------	---

---

**Description**

Function for creating a `gpmap` object representing a genotype-phenotype (GP) map for  $N$  biallelic loci or more generally  $K$  such maps, from a matrix of genotypic values.

**Usage**

```
generate_gpmap(y, locinames = NULL, allelenames = NULL, mapnames = NULL)
```

**Arguments**

y	A ( $3^N \times K$ ) matrix or numeric with each column specifying the $3^N$ genotypic values for $K$ GP maps
locinames	An optional character vector with $N$ names of loci
allelenames	An optional character object specifying allele names
mapnames	An optional character vector with $K$ names of GP maps / phenotypes

**Details**

Arguments `locinames` and `allelenames` are passed on to `enumerate_genotypes`, and the genotypic values in `y` should be given in the same sequence as the sequence of genotypes returned by `enumerate_genotypes`. If `mapnames` is not specified then the GP maps will be named "GPmap\_1", "GPmap\_2", ..., "GPmap\_K".

**Value**

The function returns an object of class `gpmap` containing the following components

values	The vector or matrix of genotypic values
nloci	The number of loci in the map
genotypes	Data frame with enumeration of genotypes
locinames	Character vector with names for all loci
mapname	The name(s) of the GP map

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**References**

Gjuvsland AB, Vik JO, Woolliams JA, Omholt SW (2011) Order-preserving principles underlying genotype-phenotype maps ensure high additive proportions of genetic variance. *Journal of Evolutionary Biology* 24(10):2269-2279 doi:10.1111/j.14209101.2011.02358.x

**Examples**

```
#inter- and intra-locus additive GPmap with two loci
generate_gpmap(c(-2,1,0,-1,0,1,0,1,2),mapnames="Additive")

#two-locus GP map with AxA epistasis for loci named A and B
generate_gpmap(c(-1,0,1,0,0,0,1,0,-1),locinames=c("A","B"),mapnames="AxA")

#random GP map with 3 loci
set.seed(0)
generate_gpmap(rnorm(27))
```

---

GPmaps

*Dataset containing example GP maps*

---

### Description

Example GP maps with two loci including: The orthogonal GP maps A, D, AA, AD, DA and DD used in decomposition of genetic variance (Zeng *et al.* 2005). The GP map mouseweight for body weight studied by Cheverud *et al.* (1995).

### Usage

```
data(GPmaps)
```

### Format

Objects of class `gpmmap`

### References

Cheverud JM & Routman EJ (1995) Epistasis and Its Contribution to Genetic Variance Components. *Genetics* 139:1455-1461 [doi:10.1093/genetics/139.3.1455](https://doi.org/10.1093/genetics/139.3.1455)

Zeng ZB, Wang T, Zou W. (2005). Modelling quantitative trait loci and interpretation of models. *Genetics* 169: 1711-1725. [doi:10.1534/genetics.104.035857](https://doi.org/10.1534/genetics.104.035857)

---

monotone\_regression

*Perform monotone regression on a genotype-phenotype (GP) map*

---

### Description

The function uses `partial_genotype_order` and `activeSet` from the `isotone` package to do monotone regression (Leeuw *et al.*, 2009) on a GP map.

### Usage

```
monotone_regression(gpmmap, plusallele)
```

### Arguments

`gpmmap` An object of class `gpmmap`  
`plusallele` An  $N$  vector of allele indexes (1 or 2)

**Details**

Element  $i$  in `plusallele` specifies the ordering of the genotypes at locus  $i$ , if the element is 1 then  $11 < 12 < 22$  and conversely if it is 2 then  $22 < 12 < 11$ . `monotone_regression` calls `partial_genotype_order` to obtain the partial ordering of genotypic values for the given plusalleles. This partial ordering is then used together with the GP map itself as input to the `activeSet` function from the package `isotone`.

**Value**

`monotone_regression` returns the output from `activeSet` directly.

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**References**

Leeuw J, Hornik K and Mair P (2009) Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. *Journal of Statistical Software* 32(5) [[link](#)]

Gjuvsland AB, Wang Y, Plahte E and Omholt SW (2013) Monotonicity is a key feature of genotype-phenotype maps. *Front. Genet.* 4:216. doi:10.3389/fgene.2013.00216

**Examples**

```
data(GPmaps)

#Additive GP map is monotone
monotone_regression(A,c(2,2))

#Pure AxA epistasis map
monotone_regression(AA,c(2,2))

#two-locus example in Cheverud & Routman (1995)
monotone_regression(mouseweight,c(1,1))
```

---

partial\_genotype\_order

*Generate partial ordering of genotype space based from allele content of genotypes*

---

**Description**

Generate the strict partial order on genotype space specified in eq. (13) in Gjuvsland *et al.* (2011). For a genotype space with  $N$  biallelic loci a minimal description of this partial order is given by  $2N * 3^{(N-1)}$  inequalities.

**Usage**

```
partial_genotype_order(plusallele)
```

**Arguments**

plusallele      A  $N$ -vector with indexes (1 or 2) for the +allele for each locus

**Details**

In short a partial order on a set is a binary relation defining a pairwise ordering of some pairs of elements in the set, for a formal definition see [https://en.wikipedia.org/wiki/Partial\\_order](https://en.wikipedia.org/wiki/Partial_order). In the partial order on the set of genotypes defined in Gjuvsland et al. (2011) the comparable pairs of genotypes are equal at every locus except one, while all other pairs of genotypes are incomparable. This partial ordering of genotype space is implicit in the regression on gene content (the number of alleles with a given index in each genotype) used for decomposition of the genotypic value in quantitative genetics (see e.g. Lynch and Walsh page 65).

**Value**

Returns a  $(2N * 3^{(N-1)}) \times 2$  matrix of genotype indexes. The genotype indexes refer to row number in the genotype sequence set up in [enumerate\\_genotypes](#). Each row vector in the matrix contains the genotype indexes of one comparable pair, and if the first index is  $k$  and the second is  $l$  then  $genotype[k] < genotype[l]$ .

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**References**

Gjuvsland AB, Vik JO, Woolliams JA, Omholt SW (2011) Order-preserving principles underlying genotype-phenotype maps ensure high additive proportions of genetic variance. *Journal of Evolutionary Biology* 24(10):2269-2279 doi:10.1111/j.14209101.2011.02358.x

Lynch M & Walsh B (1998) *Genetics and Analysis of Quantitative Traits*, Sunderland, MA: Sinauer Associates

---

plot.gpmap

*Functions for creating lineplots of genotype-phenotype (GP) maps*

---

**Description**

Function for creating lineplots for genotype-phenotype (GP) map (an object of class [gpmap](#)) with 1-3 biallelic loci.

**Usage**

```
## S3 method for class 'gpmap'
plot(x, show=1, decomposed=FALSE, ...)
```

**Arguments**

x	A gpmap object
show	Which map (only used if >1 map in gpmap object) to plot
decomposed	Decomposition into monotone and non-monotone component plotted if TRUE
...	ignored

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**Examples**

```
data(GPmaps)

#plot additive gpmap
plot(A)

#plot decomposition of GPmap from Cheverud & Routman (1995)
decomp <- decompose_monotone(mouseweight)
plot(decomp, decomposed=TRUE)
```

---

print.gpmap

*Print function for [gpmap](#) objects*

---

**Description**

Print a summary of a genotype-phenotype (GP) map (an object of class [gpmap](#)) with 1-3  $N$  biallelic loci.

**Usage**

```
## S3 method for class 'gpmap'
print(x, ...)
```

**Arguments**

x	A <a href="#">gpmap</a> object
...	ignored

**Details**

Prints name(s) of GP map(s) and loci, a summary of genotypic values. Monotonicity measures are printed if available.

**Author(s)**

Arne B. Gjuvsland <arne.gjuvsland@nmbu.no> and Yunpeng Wang <yunpeng.wng@gmail.com>

**Examples**

```
data(GPmaps)  
print(A)
```

# Index

- \* **datasets**
  - GPmaps, 8
- \* **functional epistasis**
  - gmap-package, 2
- \* **genotype-phenotype map**
  - gmap-package, 2
- \* **monotonicity**
  - gmap-package, 2
- \* **package**
  - gmap-package, 2

A (GPmaps), 8  
AA (GPmaps), 8  
activeSet, 8, 9  
AD (GPmaps), 8

D (GPmaps), 8  
DA (GPmaps), 8  
DD (GPmaps), 8  
decompose\_monotone, 2, 3  
decompose\_monotone\_single  
    (decompose\_monotone), 3  
degree\_of\_monotonicity, 2, 4  
degree\_of\_monotonicity\_single  
    (degree\_of\_monotonicity), 4

enumerate\_genotypes, 2, 5, 7, 10

generate\_gmap, 2, 6  
gmap, 3–8, 10, 11  
gmap (generate\_gmap), 6  
gmap-package, 2  
GPmaps, 8

monotone\_regression, 3, 8  
mouseweight (GPmaps), 8

partial\_genotype\_order, 8, 9, 9  
plot.gmap, 2, 10  
print.gmap, 11