Package ‘maxRgain’

May 8, 2026

Title Maximizing Polyclonal Selection Gains Using Integer Programming

Version 1.1.0

Description Implements an Integer Programming-based method for optimising genetic gain in polyclonal selection, where the goal is to select a group of genotypes that jointly meet multi-trait selection criteria. The method uses predictors of genotypic effects obtained from the fitting of mixed models. Its application is demonstrated with grapevine data, but is applicable to other species and breeding contexts. For more details see Surgy et al. (2025) <doi:10.1007/s00122-025-04885-0>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

BugReports https://github.com/soniasurgy/maxRgain/issues

Imports lpSolve, utils, stats

Suggests knitr, rmarkdown, roxygen2, devtools, usethis

VignetteBuilder knitr

Depends R (>= 3.5.0)

LazyData true

NeedsCompilation no

Author Sónia Surgy [aut, cre],
Jorge Cadima [aut],
Elsa Gonçalves [aut]

Maintainer Sónia Surgy <soniasurgy@isa.ulisboa.pt>

Repository CRAN

Date/Publication 2026-03-03 18:20:02 UTC

Contents

Gouveio 2
polyclonal 2
rmaxa 4
rmaxp 5
update.polyresult 7

Index 9

`Gouveio` *Gouveio variety dataset*

Description

A dataset containing the Empirical Best Linear Unbiased Predictors (EBLUP) of genotypic effects obtained by the fitting of linear mixed models with the observations from polyclonal selection field trials of variety Gouveio.

Usage

```
Gouveio
```

Format

An object of class `data.frame` with 150 rows and 6 columns.

Details

Clone the genotype label

yd EBLUP of the genotypic effect of yield

pa EBLUP of the genotypic effect of potential alcool

ta EBLUP of the genotypic effect of total acidity

bw EBLUP of the genotypic effect of berry weight

ph EBLUP of the genotypic effect of pH

`polyclonal` *Integer programming calculations*

Description

This function maximizes the predicted genetic gain in the selection of groups of genotypes based on the predictors of genotypic effects.

Usage

```
polyclonal(
  traits,
  ref = NULL,
  clmin = 2,
  clmax,
  dmg = NULL,
  meanvec = NULL,
  criteria = NULL,
  data
)
```

Arguments

`traits`	A vector with the names of the columns in the data corresponding to the target traits to be optimized, i.e., those included in the objective function.
`ref`	Name of the reference column (e.g., genotype ID). Defaults to the first column.
`clmin`	An integer specifying the minimum group size. If omitted, equal to 2.
`clmax`	An integer specifying the maximum group size. If omitted, equal to `clmin`.
`dmg`	A `data.frame` with three columns defining constraints: trait names; constraints signs (`">="`, `"<="` or `"=="`); and *right-hand side* values of the constraints.
`meanvec`	A named numeric vector of trait means; if omitted, data are assumed to be already normalized by the mean.
`criteria`	A named numeric vector indicating the selection criterion for each trait: 1 for traits to be increased, -1 for traits to be decreased. If omitted, all traits are assumed to be selected for increase.
`data`	A data frame comprising the input data consisting of the Predictors of genetic effects, which serve as the basis for the selection procedure.

Value

A list with the following components:

- `gain` with the gains of the several traits in each dimension
- `selected` with the reference os the clones selected in the group of each dimension

Note

The order of traits must be consistent across `traits`, `dmg`, `meanvec`, and `criteria`. Both `meanvec` and `criteria` must include values for all traits specified in `traits` and `dmg`.

References

Surgy, S., Cadima, J. & Gonçalves, E. Integer programming as a powerful tool for polyclonal selection in ancient grapevine varieties. Theor Appl Genet 138, 122 (2025). doi:10.1007/s00122-025048850

Examples

```
mymeanvec <- c(yd = 3.517, pa = 12.760, ta = 4.495, ph = 3.927, bw = 1.653)
mytraits <- c("yd", "pa",  "ta", "ph", "bw")
mydmg <- data.frame(
  lhs = c("yd", "pa", "ta", "ph", "bw"),
  rel = c(">=", ">=", ">=", ">=", ">="),
  rhs = c(20, 3, 3, 1, 2)
  )
mycriteria <- c(yd = 1, pa = 1, ta = 1, ph = -1, bw = -1)
selections <- polyclonal(
   traits = mytraits,
   clmin = 7,
   clmax = 20,
```

```
    dmg = mydmg,
    meanvec = mymeanvec,
    criteria = mycriteria,
    data = Gouveio
    )
selections
summary(selections)
```

---

rmaxa *Maximum admissible gain*

Description

The maximum admissible genetic gain in one trait that can be achieved without decreasing any of the other traits, as a percentage of the overall mean. This function calculates the maximum admissible gains achieved in the specified traits.

Usage

```
rmaxa(
  traits,
  ref = NULL,
  clmin = 2,
  clmax,
  constraints = NULL,
  meanvec = NULL,
  criteria = NULL,
  data
)
```

Arguments

`traits`	A vector with the names of the columns in the data corresponding to the target traits to be optimized, i.e., those included in the objective function.
`ref`	Name of the reference column (e.g., genotype ID). Defaults to the first column.
`clmin`	An integer specifying the minimum group size. If omitted, equal to 2.
`clmax`	An integer specifying the maximum group size. If omitted, equal to `clmin`.
`constraints`	Vector with traits to which constraints apply. If omitted, all except `ref` are used.
`meanvec`	A named numeric vector of trait means; if omitted, data are assumed to be already normalized by the mean.
`criteria`	A named numeric vector indicating the selection criterion for each trait: 1 for traits to be increased, -1 for traits to be decreased. If omitted, all traits are assumed to be selected for increase.
`data`	A data frame comprising the input data consisting of the Predictors of genetic effects, which serve as the basis for the selection procedure.

Value

A list with the following components:

- gain with the gains of the several traits in each dimension
- selected_<trait> with the reference of the clones selected in the group of each dimension in each trait

Note

The order of traits must be consistent across traits, constraints, meanvec, and criteria. Both meanvec and criteria must include values for all traits specified in traits and constraints. If constraints is omitted, all traits in the dataset are considered; in that case, meanvec and criteria must provide values for all of them.

References

Surgy, S., Cadima, J. & Gonçalves, E. Integer programming as a powerful tool for polyclonal selection in ancient grapevine varieties. Theor Appl Genet 138, 122 (2025). doi:10.1007/s00122-025048850

Examples

```
mymeanvec <- c(yd = 3.517, pa = 12.760, ta = 4.495, ph = 3.927, bw = 1.653)
mytraits <- c("yd", "pa")
mycriteria <- c(yd = 1, pa = 1, ta = 1, ph = -1, bw = -1)
maxadm <- rmaxa(
   traits = mytraits,
   clmin = 7,
   clmax = 20,
   meanvec = mymeanvec,
   criteria = mycriteria,
   data = Gouveio
   )
maxadm
```

---

rmaxp *Maximum possible gain*

Description

The maximum possible is the mean of the EBLUPs of the genotypic effects of the best n clones in a given trait, as a percentage of the overall mean. This function calculates the maximum possible gain achieved in the specified traits.

Usage

```
rmaxp(
  traits,
  ref = NULL,
  clmin = 2,
  clmax,
  meanvec = NULL,
  criteria = NULL,
  data
)
```

Arguments

`traits`	A vector with the names of the columns in the data corresponding to the target traits to be optimized, i.e., those included in the objective function.
`ref`	Name of the reference column (e.g., genotype ID). Defaults to the first column.
`clmin`	An integer specifying the minimum group size. If omitted, equal to 2.
`clmax`	An integer specifying the maximum group size. If omitted, equal to `clmin`.
`meanvec`	A named numeric vector of trait means; if omitted, data are assumed to be already normalized by the mean.
`criteria`	A named numeric vector indicating the selection criterion for each trait: 1 for traits to be increased, -1 for traits to be decreased. If omitted, all traits are assumed to be selected for increase.
`data`	A data frame comprising the input data consisting of the Predictors of genetic effects, which serve as the basis for the selection procedure.

Value

A list with the following components:

- `gain` with the gains of the several traits in each dimension
- `selected` with the reference of the clones selected in the group of each dimension

Note

The order of the traits must be consistent across `traits`, `meanvec`, and `criteria`.

References

Surgy, S., Cadima, J. & Gonçalves, E. Integer programming as a powerful tool for polyclonal selection in ancient grapevine varieties. Theor Appl Genet 138, 122 (2025). doi:10.1007/s00122-025048850

Examples

```
mymeanvec <- c(pa = 12.760)
mytraits <- c("pa")
maxpos <- rmaxp(
   traits = mytraits,
   clmin = 7,
   clmax = 20,
   meanvec = mymeanvec,
   data = Gouveio
   )
maxpos
```

---

update.polyresult *Update method for polyclonal results*

Description

Allows updating any argument of a previous `polyclonal()` call without rewriting the full call. Defaults are preserved for arguments not provided.

Usage

```
## S3 method for class 'polyresult'
update(object, ...)
```

Arguments

`object`	A `polyresult` object returned by `polyclonal()`.
`...`	Arguments to update in the original `polyclonal()` call.

Value

A new `polyresult` object with updated arguments applied.

Examples

```
#' Original call

selections <- polyclonal(traits = c("yd", "pa", "ta", "ph", "bw"),
                  clmin = 7, clmax = 15,
                  dmg = "base",
                meanvec = c(yd = 3.517, pa = 12.760, ta = 4.495, ph = 3.927, bw = 1.653),
                  criteria = c(yd = 1, pa = 1, ta = 1, ph = -1, bw = -1),
                  data = Gouveio)

selections
# Update clmax
selupdate1 <- update(selections, clmax = 10)
```

```
selupdate1

# Update dmg
selupdate2 <- update(selupdate1, dmg = data.frame(
                     lhs = c("yd", "pa", "ta", "ph", "bw"),
                     rel = c(">=", ">=", ">=", ">=", ">="),
                     rhs = c(25, 0, 0, 0, 0)
                     )
                   )

selupdate2
```

Index

∗ **datasets**
    Gouveio, 2

Gouveio, 2

polyclonal, 2

rmaxa, 4
rmaxp, 5

update.polyresult, 7