Package 'cwad'

September 26, 2026

Type Package

Title Connectivity-Weighted Allocation and Comparison of Field-Plot Designs

Version 0.2.0

Date 2026-09-15

Description A reproducible mixed-model toolkit for plant-breeding trial design. It evaluates any replication allocation under a known genetic relationship (kinship) matrix using one common linear-mixed-model engine on genotype means. Crucially, allocation and analysis model are crossed rather than confounded: every allocation can be scored both with and without kinship, so the precision gain attributable to a design can be separated from the gain attributable to the kinship-based analysis adopted alongside it. It computes A-optimal, connectivity-aware allocations via rank-1 Sherman-Morrison updates, and provides Monte-Carlo stress tests for outlier shrinkage and for an incorrectly specified kinship matrix, each with a matched control arm.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.1

URL https://github.com/bkpraveenars-del/cwad

BugReports https://github.com/bkpraveenars-del/cwad/issues

NeedsCompilation no

Author Praveen Kumar [aut, cre]

Maintainer Praveen Kumar <bkpraveenars@gmail.com>

Repository CRAN

Date/Publication 2026-09-26 16:10:07 UTC

Contents

`balanced_control` *Neutral integer-feasible control allocation*

Description

Balanced allocation summing to `B`: every genotype gets `floor(B/N)` plots and the surplus is placed at random. Averaging the criterion over several draws gives the control against which an optimised allocation should be judged when `B/N` is not an integer; comparing an integer allocation against a fractional "uniform r = B/N" is not a feasible comparison and inflates the apparent gain.

Usage

```
balanced_control(B, N, reps = 25L, seed = 11L)
```

Arguments

`B`	total plot budget.
`N`	number of genotypes.
`reps`	number of random draws to return.
`seed`	RNG seed.

Value

list of `reps` replication vectors.

blup — *BLUP of genotype effects from the reduced (means) model*

Description

BLUP of genotype effects from the reduced (means) model

Usage

```
blup(ybar, r, Ginv, alpha)
```

Arguments

ybar	genotype phenotypic means.
r	numeric vector of replications per genotype.
Ginv	inverse relationship matrix (use diag(N) for independence).
alpha	variance ratio σ_e^2/σ_g^2.

Value

numeric vector of predicted genotype effects.

build_mme — *Reduced mixed-model coefficient matrix on genotype means*

Description

$M = [[\sum r, r'], [r, diag(r) + \alpha G^{-1}]]$ where $\alpha = \sigma_e^2/\sigma_g^2$. The genotype-mean reduction is exact for the model $\bar{y}_i = \mu + u_i + \bar{e}_i$, $Var(\bar{e}_i) = \sigma_e^2/r_i$.

Usage

```
build_mme(r, Ginv, alpha)
```

Arguments

r	numeric vector of replications per genotype.
Ginv	inverse relationship matrix (use diag(N) for independence).
alpha	variance ratio σ_e^2/σ_g^2.

Value

the $(N+1) \times (N+1)$ coefficient matrix.

compare_designs *Crossed comparison of allocations and analysis models*

Description

Replaces the v0.1.0 `compare_designs()`. Every allocation is evaluated under BOTH analysis models – kinship-free ($G = I$) and kinship (G) – so that the allocation contrast and the analysis-model contrast are crossed rather than confounded. In v0.1.0 the caller supplied one analysis matrix per design, which made it easy (and, in the shipped example, actual) to change allocation and analysis model in the same step and then attribute the whole difference to the allocation.

Usage

```
compare_designs(G, alpha, sig2e, r_list, idx = NULL)
```

Arguments

`G`	relationship (kinship) matrix, $N \times N$.
`alpha`	variance ratio σ_e^2/σ_g^2.
`sig2e`	residual variance.
`r_list`	named list of replication vectors, each of length N (or length $N + n_{check}$ when `idx` restricts evaluation to the first N entries).
`idx`	optional 1-based indices of the evaluation set.

Value

a data.frame with one row per allocation and columns `PEV_noG`, `rel_noG`, `PEV_kin`, `rel_kin`, plus `gain_analysis_pp` and `gain_allocation_pp`, the decomposition of the naive gain relative to the first allocation in `r_list` analysed without kinship.

Examples

```
g <- make_fs_G(c(12, 12, 4, 4, 1, 1))
Ginv <- solve(g$G); N <- nrow(g$G)
r_unif <- rep(2, N)
r_cwad <- greedy_alloc(Ginv, B = 2 * N, alpha = 2)
compare_designs(g$G, alpha = 2, sig2e = 2,
                list(Uniform = r_unif, CWAD = r_cwad))
```

`greedy_alloc` *Greedy A-optimal, connectivity-aware plot allocation*

Description

Allocates a plot budget one plot at a time to the genotype whose extra replicate most reduces the mean pairwise PEV, using rank-1 Sherman-Morrison updates of the inverse MME (adding a plot to genotype i is the rank-1 update vv', $v = e_\mu + e_i$). When the relationship matrix carries genetic connectivity, the optimiser spontaneously concentrates replication on genetically isolated genotypes.

Usage

```
greedy_alloc(
  Ginv,
  B,
  alpha,
  sig2e = 1,
  floor = 1L,
  r_start = NULL,
  idx_eval = NULL,
  check = TRUE
)
```

Arguments

`Ginv`	inverse relationship matrix.
`B`	total plot budget (>= floor * N).
`alpha`	variance ratio.
`sig2e`	residual variance.
`floor`	minimum replication per genotype (default 1).
`r_start`	optional starting allocation (overrides `floor`).
`idx_eval`	evaluation set for the criterion (default all).
`check`	logical; if TRUE, validate the first Sherman-Morrison score against a brute-force re-inversion (correctness self-test).

Value

numeric vector of replications summing to `B`.

Examples

```
g <- make_fs_G(c(12, 12, 4, 4, 1, 1)); Ginv <- solve(g$G)
N <- nrow(Ginv)                   # 34 genotypes
r <- greedy_alloc(Ginv, B = 2 * N, alpha = 2)
tapply(r, g$connectivity, mean)   # isolated genotypes get more plots
```

make_fs_G *Full-sib additive relationship matrix*

Description

Builds a block-diagonal numerator relationship matrix for a set of full-sib families (diagonal 1, within-family off-diagonal `within`).

Usage

```
make_fs_G(fam_sizes, within = 0.5)
```

Arguments

`fam_sizes`	integer vector of family sizes.
`within`	within-family additive relationship (default 0.5 for full sibs).

Value

list with `G` (relationship matrix), `fam_id` (0-based family index per genotype) and `connectivity` (row-sum of off-diagonals).

Examples

```
g <- make_fs_G(c(12, 12, 4, 4, 1, 1))
range(g$connectivity)
```

Muu_inv *u-block of the inverse MME (unit-variance scale)*

Description

u-block of the inverse MME (unit-variance scale)

Usage

```
Muu_inv(r, Ginv, alpha)
```

Arguments

`r`	numeric vector of replications per genotype.
`Ginv`	inverse relationship matrix (use `diag(N)` for independence).
`alpha`	variance ratio σ_e^2/σ_g^2.

Value

the $N \times N$ genotype block of M^{-1}.

pev_pairwise *Mean pairwise prediction-error variance (A-optimality criterion)*

Description

Mean pairwise prediction-error variance (A-optimality criterion)

Usage

```
pev_pairwise(r, Ginv, alpha, sig2e, idx = NULL)
```

Arguments

r	replications.
Ginv	inverse relationship.
alpha	ratio.
sig2e	residual variance.
idx	optional 1-based indices of the evaluation set (default: all genotypes).

Value

mean variance of a difference between two genotype predictions.

reliability *Mean genotype reliability*

Description

Mean genotype reliability

Usage

```
reliability(r, Ginv, alpha, idx = NULL)
```

Arguments

r	replications.
Ginv	inverse relationship.
alpha	ratio.
idx	optional 1-based indices of the evaluation set (default: all genotypes).

Value

mean of $1 - \alpha\,(M^{-1})_{ii}$ over the evaluation set.

`sim_transgressive` *Transgressive-segregant shrinkage experiment, with matched controls*

Description

Plants one genotype with a genetic value `spike` genetic standard deviations above its family mean and measures, by Monte Carlo over residual noise, how far each arm's BLUP is pulled toward the family mean.

Usage

```
sim_transgressive(
  G,
  alpha,
  sig2e,
  sig2g,
  fam_id,
  t,
  r_list,
  spike = 3,
  reps = 3000,
  seed = 7
)
```

Arguments

`G`	relationship matrix.
`alpha`	variance ratio.
`sig2e`	residual variance.
`sig2g`	genetic variance.
`fam_id`	0-based family index per genotype.
`t`	1-based index of the transgressive genotype.
`r_list`	named list of replication vectors.
`spike`	deviation added to genotype `t`, in genetic sd units.
`reps`	Monte Carlo replications.
`seed`	RNG seed.

Details

Unlike v0.1.0, every allocation in `r_list` is run under BOTH analysis models, so the shrinkage attributable to borrowing from relatives is separated from the shrinkage attributable to reduced replication. In v0.1.0 the uniform arm was analysed without kinship and the CWAD arm with it, and the whole difference was reported as a property of CWAD; on the shipped example 58 percent of it is the analysis model.

Value

list with `true_val`, `fam_mean`, the raw `estimates` matrix and a data.frame `summary` carrying one row per (allocation, analysis model) cell.

`sim_wrongG` *Wrong-kinship breaking-point experiment, with matched controls*

Description

Generates truth under $G_{true} = (1-\delta)G + \delta I$ but analyses with the assumed kinship `G`, and reports RMSE of the BLUP for every (allocation, analysis model) cell across δ.

Usage

```
sim_wrongG(
  G,
  alpha,
  sig2e,
  sig2g,
  r_list,
  split_by_r = TRUE,
  deltas = seq(0, 1, by = 0.1),
  reps = 500,
  seed = 7
)
```

Arguments

`G`	assumed relationship matrix.
`alpha`	variance ratio.
`sig2e`	residual variance.
`sig2g`	genetic variance.
`r_list`	named list of replication vectors; the first is the reference.
`split_by_r`	logical; if TRUE, additionally report each allocation's $r = 1$ and $r \geq 2$ genotypes separately.
`deltas`	sequence of misspecification levels.
`reps`	Monte Carlo replications per delta.
`seed`	RNG seed.

Details

The matched control is the reference allocation analysed with the SAME wrong kinship. Comparing an aggressive allocation analysed with kinship against a uniform allocation analysed WITHOUT kinship – as v0.1.0 did – measures the cost of using kinship at all, because a kinship-free analysis cannot be harmed by a wrong kinship matrix; its curve is flat by construction and any crossing with it is an artefact of the mismatch, not a breaking point.

Value

long-format data.frame: delta, allocation, analysis, subset, RMSE.

Index