

Package ‘GreenManureR’

October 13, 2026

Type Package

Title Green Manure Decomposition, Nutrient Release and Crop Response

Version 0.1.0

Description Tools for analysing green manure and cover crop experiments, for example with Mucuna, sunhemp and dhaincha. Fits single-pool, double-pool and asymptotic litterbag decomposition models and ranks them by the corrected Akaike information criterion, estimates the time to 50 percent mass loss with Monte-Carlo confidence intervals, and quantifies nitrogen, phosphorus, potassium and other nutrient release over time. Further functions compute nutrient inputs, crop uptake, apparent nitrogen recovery and fertilizer nitrogen replacement value, estimate residual effects of green manures on succeeding kharif and rabi crops, link decomposition, nutrient release and crop response in a joint analysis, and generate standardized tables, diagnostic plots and a Word report.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports stats, utils, withr

Suggests ggplot2, officer, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

LazyData true

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation no

Author Anbukkani Perumal [aut, cre, cph],
Ramachandran S [aut, cph]

Maintainer Anbukkani Perumal <anbueconomic@gmail.com>

Repository CRAN

Date/Publication 2026-10-13 15:00:18 UTC

Contents

carryover_effect	2
decomp_fit	3
decomp_t50	5
gm_agronomy	6
gm_joint_analysis	7
gm_litterbag	8
gm_methods	9
greenmanure_n_balance	10
greenmanure_report	12
nutrient_release	13
simulate_greenmanure	14
Index	16

carryover_effect	<i>Residual (carry-over) effects of green manure on succeeding crops</i>
------------------	--

Description

Quantifies the effect of each green manure treatment relative to a control on the yield (or any response) of successive crops, for example a kharif crop followed by a rabi crop.

Usage

```
carryover_effect(  
  data,  
  response = "grain_yield",  
  treatment = "treatment",  
  control = "Control",  
  season = "season",  
  block = "block",  
  strata = NULL,  
  season_levels = NULL,  
  adjust = "holm",  
  level = 0.95  
)
```

Arguments

data	Plot-level data frame.
response	Column with the response (e.g. grain yield).
treatment	Column with treatment labels.
control	Label of the control treatment.
season	Column identifying the crop or season.
block	Column identifying blocks, or NULL.

strata	Optional columns to analyse separately (e.g. "fert_N", site).
season_levels	Order of seasons in the crop sequence (defaults to order of appearance).
adjust	Method passed to <code>stats::p.adjust()</code> .
level	Confidence level for intervals.

Details

For every stratum and season the model $\text{response} \sim \text{block} + \text{treatment}$ is fitted (randomized complete block design; without a block term if `block = NULL`). Treatment-versus-control contrasts use the model estimates and standard errors, with p-values adjusted across treatments by `adjust`. The percentage change is the contrast divided by the control mean; the log response ratio uses the pooled residual variance in a delta-method standard error. `retention_vs_first` expresses each seasonal effect as a percentage of the effect in the first season.

The pooled analysis fits $\text{response} \sim \text{season}:\text{block} + \text{treatment} + \text{treatment}:\text{season}$ across seasons; a significant treatment x season term indicates that the effect changes between crops. Seasonal variances are assumed homogeneous, which should be checked. The cumulative effect is the sum of seasonal contrasts, with a standard error that assumes independence between seasons and is therefore approximate.

Value

An object of class "gm_carryover": a list with effects (contrasts per stratum, season and treatment), anova (treatment F-tests within season), pooled (pooled analysis, or NULL with one season), cumulative, response and control.

See Also

`greenmanure_n_balance()`, `gm_joint_analysis()`

Examples

```
co <- carryover_effect(gm_agronomy, response = "grain_yield",
                      strata = "fert_N",
                      season_levels = c("Kharif", "Rabi"))
co
```

decomp_fit

Fit litterbag decomposition models

Description

Fits single-pool, double-pool and asymptotic decay models to litterbag mass-loss data, separately for each group (for example each green manure species), and ranks the models by AICc.

Usage

```
decomp_fit(
  data,
  time = "time",
  mass = "mass",
  group = NULL,
  models = c("single", "double", "asymptotic"),
  mass_scale = c("percent", "fraction")
)
```

Arguments

<code>data</code>	A data frame in long format with one row per litterbag and sampling date.
<code>time</code>	Name of the column holding time since incorporation (e.g. days).
<code>mass</code>	Name of the column holding the mass remaining.
<code>group</code>	Character vector of grouping columns (species, site, ...), or NULL to fit a single group.
<code>models</code>	Any of "single", "double", "asymptotic".
<code>mass_scale</code>	Either "percent" (default) or "fraction", the scale of mass.

Details

With M the percentage of the initial mass remaining and t the time since incorporation, the models are:

- single: $M = 100 * \exp(-k * t)$
- double: $M = 100 * (A * \exp(-k_f * t) + (1 - A) * \exp(-k_s * t))$, fitted with $k_s = r * k_f$ and $0 < r \leq 1$, so the fast pool can never decay more slowly than the slow pool.
- asymptotic: $M = C + (100 - C) * \exp(-k * t)$, where C is the recalcitrant residue (percent of initial mass).

The initial mass is fixed at 100%, so time-zero observations should be included (or the data expressed relative to the initial mass). Each model is fitted by nonlinear least squares (port algorithm with bounds) from a small grid of starting values, and the best-fitting converged solution is kept.

Value

An object of class "gm_decomp", a list with components `table` (one row per group and model with parameters k , k_{slow} , pool_A , $C_{\text{resistant}}$, fit statistics `RSS`, `RMSE`, `R2`, `AIC`, `AICc`, `dAICc`, Akaike weight, and selected marking the lowest `AICc`), `fits` (the `nls` objects), `data`, `cols` and `groups`. For the double-pool model k is the fast-pool rate and k_{slow} the slow-pool rate.

See Also

[decomp_t50\(\)](#), [nutrient_release\(\)](#)

Examples

```
fit <- decomp_fit(gm_litterbag, time = "time", mass = "mass",
                 group = "treatment")
fit
fit$table[fit$table$selected, c("treatment", "model", "k", "R2")]
```

decomp_t50	<i>Time to a given percentage of mass (or nutrient) loss</i>
------------	--

Description

Solves the fitted decay model for the time at which a proportion p of the initial mass has been lost ($p = 0.5$ gives $t50$), with an optional Monte-Carlo confidence interval.

Usage

```
decomp_t50(
  fit,
  p = 0.5,
  model = "best",
  ci = TRUE,
  nsim = 2000,
  level = 0.95,
  seed = 123
)
```

Arguments

<code>fit</code>	An object returned by <code>decomp_fit()</code> .
<code>p</code>	Proportion lost, strictly between 0 and 1 (0.5 gives $t50$, 0.95 gives $t95$).
<code>model</code>	"best" (lowest AICc, default) or the name of a model.
<code>ci</code>	Logical; compute the Monte-Carlo interval?
<code>nsim</code>	Number of Monte-Carlo draws.
<code>level</code>	Confidence level.
<code>seed</code>	Integer seed used for the draws. The global random number state is left unchanged. Use NULL to draw without fixing a seed.

Details

The interval is obtained by drawing `nsim` parameter vectors from a multivariate normal distribution centred on the estimates with the covariance matrix of the `nls` fit, discarding draws outside the admissible parameter space, and taking quantiles of the resulting times. If the model never reaches the target (for example an asymptotic model with a residue above the target), NA is returned.

Value

A data frame with the grouping columns, the model used, p, the time (column named e.g. t50) and the interval limits lwr and upr.

See Also

[decomp_fit\(\)](#)

Examples

```
fit <- decomp_fit(gm_litterbag, group = "treatment")
decomp_t50(fit)
decomp_t50(fit, p = 0.95, ci = FALSE)
```

gm_agronomy

Simulated two-season agronomic data

Description

Simulated field data for a control and three green manures at three fertilizer N levels, in four blocks, for a kharif crop followed by a rabi crop. Generated with [simulate_greenmanure\(\)](#) (seed 42); the values are synthetic.

Usage

```
gm_agronomy
```

Format

A data frame with 96 rows and 10 variables:

treatment Control or green manure species.

fert_N Fertilizer N applied (kg/ha).

block Block (1-4).

season Crop season (Kharif, Rabi).

gm_biomass Green manure dry matter (t/ha).

gm_N Green manure N concentration (%).

grain_yield, straw_yield Yields (t/ha).

grain_N, straw_N N concentrations (%).

Source

Simulated; see data-raw/simulate-data.R.

See Also

[simulate_greenmanure\(\)](#), [greenmanure_n_balance\(\)](#), [carryover_effect\(\)](#)

gm_joint_analysis	<i>Joint analysis of decomposition, nutrient release and crop response</i>
-------------------	--

Description

Builds one row per treatment that joins the decomposition rate and t50, the cumulative release of one nutrient by a chosen day, apparent N recovery and the percentage yield response in each season, and reports Spearman rank correlations among these variables.

Usage

```
gm_joint_analysis(
  decomp,
  nutrient,
  nbal,
  carry,
  key = "treatment",
  nutrient_name = "N",
  at_day = 60,
  anr_season = NULL,
  fert_N_level = NULL
)
```

Arguments

decomp	A gm_decomp object from decomp_fit() (one grouping column named as key).
nutrient	A gm_nutrient object from nutrient_release() .
nbal	A gm_nbalance object from greenmanure_n_balance() .
carry	A gm_carryover object from carryover_effect() .
key	Name of the treatment column shared by all inputs.
nutrient_name	Nutrient (as named in nutrient_release()) whose release is linked to crop response.
at_day	Day at which model-based cumulative release is evaluated.
anr_season, fert_N_level	Optional filters restricting the plots used to average apparent N recovery: the first by stratum of nbal and the fertilizer N level.

Details

With only a few treatments (for example three green manure species) the correlations are descriptive only, and a warning is issued when fewer than five rows are available. To draw inference, pool several sites or years.

Value

An object of class "gm_joint": a list with data (the linkage table) and cor (Spearman correlation matrix, or NULL).

See Also

[decomp_fit\(\)](#), [nutrient_release\(\)](#), [greenmanure_n_balance\(\)](#), [carryover_effect\(\)](#)

Examples

```
fit <- decomp_fit(gm_litterbag, group = "treatment")
nr <- nutrient_release(gm_litterbag, group = "treatment",
                      biomass = "biomass_tha")
nb <- greenmanure_n_balance(gm_agronomy, by = "season")
co <- carryover_effect(gm_agronomy, strata = "fert_N")
ja <- suppressWarnings(
  gm_joint_analysis(fit, nr, nb, co, at_day = 60, fert_N_level = 0))
ja$data
```

gm_litterbag

Simulated litterbag decomposition data

Description

Simulated litterbag data for three green manure species, with mass and nutrient concentrations over 112 days. Generated with [simulate_greenmanure\(\)](#) (seed 42); the values are synthetic.

Usage

```
gm_litterbag
```

Format

A data frame with 108 rows and 8 variables:

treatment Green manure species (Mucuna, Sunhemp, Dhaincha).

rep Litterbag replicate (1-4).

time Days after incorporation (0-112).

mass Dry mass remaining (% of initial).

N, P, K Nutrient concentrations (% dry weight).

biomass_tha Green manure dry matter incorporated (t/ha).

Source

Simulated; see data-raw/simulate-data.R.

See Also

[simulate_greenmanure\(\)](#), [decomp_fit\(\)](#), [nutrient_release\(\)](#)

Description

Print, prediction and plotting methods for the objects returned by `decomp_fit()`, `nutrient_release()`, `greenmanure_n_balance()`, `carryover_effect()` and `gm_joint_analysis()`. Plot methods return ggplot objects and require the 'ggplot2' package.

Usage

```
## S3 method for class 'gm_decomp'
print(x, digits = 4, ...)

## S3 method for class 'gm_decomp'
predict(object, times, model = "best", ...)

## S3 method for class 'gm_nutrient'
print(x, digits = 3, ...)

## S3 method for class 'gm_nbalance'
print(x, digits = 3, ...)

## S3 method for class 'gm_carryover'
print(x, digits = 3, ...)

## S3 method for class 'gm_joint'
print(x, digits = 3, ...)

## S3 method for class 'gm_decomp'
plot(x, y, model = "best", ...)

## S3 method for class 'gm_nutrient'
plot(x, y, ...)

## S3 method for class 'gm_nbalance'
plot(x, y, ...)

## S3 method for class 'gm_carryover'
plot(x, y, ...)
```

Arguments

x, object	An object returned by one of the package functions.
digits	Number of significant digits to print.
...	Further arguments (ignored).

times	Times (e.g. days) at which to predict mass remaining.
model	"best" (lowest AICc) or the name of a model.
y	Unused; present for consistency with the <code>plot()</code> generic.

Value

print methods return x invisibly. `predict.gm_decomp()` returns a data frame with the grouping columns, model, time and pred (predicted mass remaining, %). plot methods return a ggplot object.

Examples

```
fit <- decomp_fit(gm_litterbag, group = "treatment")
predict(fit, times = c(30, 60, 90))
if (requireNamespace("ggplot2", quietly = TRUE)) plot(fit)
```

greenmanure_n_balance *Nitrogen inputs, crop uptake and apparent balance*

Description

Plot-level nitrogen accounting for green manure experiments, with treatment summaries.

Usage

```
greenmanure_n_balance(
  data,
  treatment = "treatment",
  control = "Control",
  rep = "block",
  by = NULL,
  gm_biomass = "gm_biomass",
  gm_N = "gm_N",
  fert_N = "fert_N",
  grain_yield = "grain_yield",
  grain_N = "grain_N",
  straw_yield = "straw_yield",
  straw_N = "straw_N",
  nfrv = TRUE
)
```

Arguments

data	Plot-level data frame.
treatment	Column with the treatment label.
control	Label of the treatment without green manure.

rep	Column identifying replicates or blocks (not used in the calculations; kept for documentation and future use).
by	Optional columns defining strata (e.g. season, site, year).
gm_biomass, gm_N	Columns with green manure dry matter (t/ha) and N concentration (%).
fert_N	Column with fertilizer N applied (kg/ha). If absent it is taken as zero.
grain_yield, grain_N, straw_yield, straw_N	Columns with yields (t/ha) and N concentrations (%). Straw columns may be absent.
nfrv	Logical; compute the fertilizer N replacement value?

Details

All quantities are in kg N/ha unless stated otherwise:

- N_gm_input: green manure dry matter (t/ha) x 1000 x N percent / 100.
- N_uptake: grain plus straw N removal.
- N_balance: (green manure N + fertilizer N) minus crop N uptake, an apparent balance (losses and soil N are not measured).
- ANR: apparent N recovery from green manure (percent): (uptake - mean control uptake at the same stratum and fertilizer N) / green manure N input x 100.
- AE_gmN: additional grain (kg) per kg of green manure N.
- NFRV: fertilizer N replacement value, the fertilizer N needed in the control plots (quadratic response fitted per by stratum, requiring at least three fertilizer levels) to match the green manure plot yield, minus the fertilizer N applied. NA when the yield lies outside the fitted response range.

Value

An object of class "gm_nbalance": a list with plot (plot-level results), summary (means, standard deviations and standard errors per stratum, treatment and fertilizer level), by and treatment.

See Also

[carryover_effect\(\)](#)

Examples

```
nb <- greenmanure_n_balance(gm_agronomy, by = "season")
nb
head(nb$summary[, c("season", "treatment", "fert_N", "ANR_mean")])
```

greenmanure_report	<i>Standardized tables, diagnostic plots and Word report</i>
--------------------	--

Description

Writes tables (CSV), figures (PNG) and, if the 'officer' package is installed, a Word report containing all tables and figures for any combination of analysis objects.

Usage

```
greenmanure_report(  
  decomp = NULL,  
  nutrient = NULL,  
  nbal = NULL,  
  carry = NULL,  
  joint = NULL,  
  outdir,  
  docx = TRUE,  
  dpi = 300,  
  digits = 3  
)
```

Arguments

decomp, nutrient, nbal, carry, joint	Objects returned by <code>decomp_fit()</code> , <code>nutrient_release()</code> , <code>greenmanure_n_balance()</code> , <code>carryover_effect()</code> and <code>gm_joint_analysis()</code> . Any may be NULL.
outdir	Directory in which to write the files (required).
docx	Logical; also write <code>GreenManureR_report.docx</code> ?
dpi	Resolution of the PNG figures.
digits	Significant digits used in the exported tables.

Details

Nothing is written unless `outdir` is supplied; the directory is created if needed. Figures require 'ggplot2' and the Word report requires 'officer'; when either is missing the corresponding output is skipped with a message and the remaining outputs are still produced.

Value

Invisibly, a list with tables (data frames) and figures (file paths).

Examples

```
fit <- decomp_fit(gm_litterbag, group = "treatment")
out <- file.path(tempdir(), "gm_report")
greenmanure_report(decomp = fit, outdir = out, docx = FALSE)
list.files(out)
```

nutrient_release	<i>Nutrient release from decomposing green manure residues</i>
------------------	--

Description

Calculates the percentage of the initial nutrient content remaining in litterbags, fits decay models to each nutrient and reports the time to 50% release and model-based cumulative release at chosen times.

Usage

```
nutrient_release(
  data,
  time = "time",
  mass = "mass",
  nutrients = c("N", "P", "K"),
  group = NULL,
  biomass = NULL,
  at = c(7, 14, 30, 60, 90),
  models = c("single", "double", "asymptotic")
)
```

Arguments

data	Long data frame of litterbag observations including the initial (time zero) sampling.
time, mass	Names of the time and mass-remaining (%) columns.
nutrients	Names of the columns holding nutrient concentrations (% dry weight), e.g. c("N", "P", "K").
group	Character vector of grouping columns, or NULL.
biomass	Optional. Name of a column, or a single number, giving the dry-matter input in t/ha. If supplied, nutrient input and release are also reported in kg/ha.
at	Times at which model-based cumulative release is reported.
models	Models passed to decomp_fit() .

Details

Nutrient remaining (% of initial content) is $\text{mass} * \text{concentration} / \text{initial concentration}$, where mass is the percentage of the initial dry mass remaining and the initial concentration is the group mean at the earliest sampling time (which must therefore be present in data). Values above 100% indicate net immobilization. Decay models are fitted with `decomp_fit()` to the nutrient-remaining percentages.

Value

An object of class "gm_nutrient": a list with data (long table with remaining, released and optionally input_kg_ha, released_kg_ha), fits (a gm_decomp object per nutrient), t50, release_at, gcols and at.

See Also

`decomp_fit()`, `decomp_t50()`

Examples

```
nr <- nutrient_release(gm_litterbag, nutrients = c("N", "P", "K"),
                      group = "treatment", biomass = "biomass_tha",
                      at = c(14, 30, 60))
nr
subset(nr$release_at, nutrient == "N")
```

simulate_greenmanure *Simulate litterbag and two-season agronomic data*

Description

Generates synthetic data with the structure expected by the package functions: a litterbag decomposition experiment with three green manure species (Mucuna, sunhemp, dhaincha) and a replicated field experiment with a control, three fertilizer N levels and two successive crops (kharif and rabi). The values are simulated for testing and teaching and do not represent real measurements.

Usage

```
simulate_greenmanure(seed = 42)
```

Arguments

seed	Integer seed. The global random number state is left unchanged. Use NULL to draw without fixing a seed.
------	---

Value

A list with data frames litterbag (see `gm_litterbag`) and agronomy (see `gm_agronomy`).

Examples

```
sim <- simulate_greenmanure(seed = 1)
str(sim$litterbag)
```

Index

* datasets

- gm_agronomy, 6
- gm_litterbag, 8

carryover_effect, 2
carryover_effect(), 6–9, 11, 12

decomp_fit, 3
decomp_fit(), 5–9, 12–14
decomp_t50, 5
decomp_t50(), 4, 14

gm_agronomy, 6, 14
gm_joint_analysis, 7
gm_joint_analysis(), 3, 9, 12
gm_litterbag, 8, 14
gm_methods, 9
greenmanure_n_balance, 10
greenmanure_n_balance(), 3, 6–9, 12
greenmanure_report, 12

nutrient_release, 13
nutrient_release(), 4, 7–9, 12

plot(), 10
plot.gm_carryover (gm_methods), 9
plot.gm_decomp (gm_methods), 9
plot.gm_nbalance (gm_methods), 9
plot.gm_nutrient (gm_methods), 9
predict.gm_decomp (gm_methods), 9
print.gm_carryover (gm_methods), 9
print.gm_decomp (gm_methods), 9
print.gm_joint (gm_methods), 9
print.gm_nbalance (gm_methods), 9
print.gm_nutrient (gm_methods), 9

simulate_greenmanure, 14
simulate_greenmanure(), 6, 8
stats::p.adjust(), 3