Package 'lfebd3'

May 7, 2026

Type Package

Title Generation and Analysis of 3-Level and 5-Level Factorial Block Designs

Version 0.2.0

Maintainer Sukanta Dash <sukanta.iasri@gmail.com>

Description Provides tools to generate and analyze 3-level and 5-level linear factorial block designs, including complete factorial layouts, fractional factorial layouts, confounded factorial layouts, and design-characteristic summaries. The package includes utilities for recursive construction, defining-contrast identification, alias and confounding summaries, incidence matrix construction, and selected design-characteristic diagnostics. The methodological framework follows foundational work on factorial block designs, including Gupta (1983) <doi:10.1111/j.2517-6161.1983.tb01253.x>.

License GPL-3

Encoding UTF-8

Imports MASS, Matrix, stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Vankudoth Kumar [aut],
Sukanta Dash [aut, cre],
Med Ram Verma [aut]

Repository CRAN

Date/Publication 2026-05-07 13:40:02 UTC

Contents

`build_block_matrix` *Build a treatment-by-block incidence matrix*

Description

Creates the incidence matrix for a block design from a list of treatment labels grouped by block.

Usage

```
build_block_matrix(blocks)
```

Arguments

`blocks`	Named or unnamed list of blocks, where each block is a vector of treatment labels.

Value

A binary matrix with treatments in rows and blocks in columns.

See Also

`convert_to_blocks()`, `FactChar()`

Examples

```
N <- build_block_matrix(list(B1 = c("00", "11"), B2 = c("01", "10")))
stopifnot(ncol(N) == 2)
```

---

build_effect_matrix *Build a model matrix for factorial effects*

Description

Generates the treatment-effect model matrix for a factorial design using sum-to-zero contrasts.

Usage

```
build_effect_matrix(trts, factor_levels)
```

Arguments

`trts`	Character vector of treatment labels. Included for interface compatibility with the original script.
`factor_levels`	Integer vector giving the number of levels for each factor.

Value

A model matrix with one row per treatment combination.

See Also

`FactChar()`

Examples

```
X <- build_effect_matrix(NULL, c(3, 3))
stopifnot(nrow(X) == 9)
```

---

convert_to_blocks *Convert a generated design object to a block list*

Description

Converts either a treatment-run data frame or a block-wise design data frame into a named list of blocks.

Usage

```
convert_to_blocks(design_df)
```

Arguments

design_df	Data frame returned by a design-generation function.

Value

A named list where each element is a character vector of treatment labels belonging to one block.

See Also

build_block_matrix(), FactChar()

Examples

```
blocks <- convert_to_blocks(lfebd3(2))
stopifnot(length(blocks) == 1)
```

FactChar *Analyze factorial block-design characteristics*

Description

Computes incidence-based, estimability, balance, confounding, discrepancy, and optimality summaries for a factorial block design.

Usage

```
FactChar(factor_levels, blocks)
```

Arguments

factor_levels	Integer vector giving the number of levels for each factor.
blocks	List of blocks, where each block is a character vector of treatment labels.

Details

This function contains several local helper functions for pseudo-inverse calculation, contrast construction, Das-style diagnostics, discrepancy measures, and confounding checks. Those helper functions are scoped locally and are not intended to be documented as separate package-level functions.

Value

An object of class "lfebd3_analysis" containing the incidence matrix, C-matrix, design-property flags, confounding summary, discrepancy criteria, optimality diagnostics, and Das-style summaries. Use print() to display the formatted report.

See Also

build_block_matrix(), lfebd3_analyze(), lfebd3.cf.full()

Examples

```
d <- lfebd3.cf.full(2, 1)
res <- FactChar(c(3, 3), convert_to_blocks(d))
stopifnot(inherits(res, "lfebd3_analysis"))
```

FactChar_fast *Fast factorial block-design screening diagnostics*

Description

Computes a lightweight subset of the diagnostics returned by FactChar(). This function is intended for large designs, especially 5-level designs with n > 3, where the exact all-effects diagnostics require large generalized inverses and pairwise distance matrices. It preserves the same printed-object class as FactChar() so the result can still be displayed with print.lfebd3_analysis().

Usage

```
FactChar_fast(factor_levels, blocks)
```

Arguments

factor_levels	Integer vector giving the number of levels for each factor.
blocks	List of blocks, where each block is a character vector of treatment labels.

Value

An object of class "lfebd3_analysis" containing fast design properties and block-confounding summaries. Expensive exact diagnostics are set to NA or skipped and reported as such by print().

Examples

```
d <- as.data.frame(lfebd5.fr(3, c = 1))
blk <- list(B1 = apply(d, 1, paste0, collapse = ""))
res <- FactChar_fast(rep(5, 3), blk)
stopifnot(inherits(res, "lfebd3_analysis"))
```

fractional_confounding_summary *Summarize aliasing and confounding in a fractional LFBD*

Description

Produces a low-order summary for main effects and two-factor interactions under the defining subgroup of a fractional 3-level factorial design.

Usage

```
fractional_confounding_summary(n, dc_basis)
```

Arguments

n	Integer. Number of factors.
dc_basis	Matrix whose columns form the basis of defining contrasts.

Value

A data frame describing each listed effect, its order, low-order aliases, and simple estimability/confounding indicators.

See Also

find_defining_contrasts(), lfebd3.fr()

generate_Tn_full *Generate the full ternary construction matrix*

Description

Builds the full recursive ternary matrix used as the foundation for generating complete, fractional, and confounded 3-level factorial layouts in this script.

Usage

```
generate_Tn_full(n)
```

Arguments

n	Integer. Number of recursion levels (or factors).

Details

The construction starts from a single-column seed and applies the 3-by-3 recursive expansion rule `n` times. The returned matrix is transposed so that downstream functions can extract the square design submatrix directly.

Value

A matrix with entries in `0`, `1`, and `2` representing the full ternary construction after `n` recursive expansions.

References

Elsawah, A.M. Multiple doubling: a simple effective construction technique for optimal two-level experimental designs. *Statistical Papers* **62**, 2923–2967 (2021). DOI: 10.1007/s00362-020-01221-0

Gupta, S. C. (1983). Some new methods for constructing block designs having orthogonal factorial structure. Journal of the Royal Statistical Society, Series B (Methodological), 45, 297-307.

See Also

`get_Tn_square()`, `lfebd3()`, `lfebd3.fr()`, `lfebd3.cf()`

Examples

```
generate_Tn_full(1)
dim(generate_Tn_full(2))
get_Tn_square(1)
dim(get_Tn_square(2))
d <- lfebd3(2)
d
nrow(d)
d <- lfebd3.fr(2, 1)
d
attr(d, "defining_contrasts")
lfebd3.cf(2, 1)
dim(lfebd3.cf(3, 2))
d <- lfebd3.cf.full(2, 1)
d
colnames(d)
d <- lfebd3(2)
blk <- convert_to_blocks(d)
blk
d <- lfebd3.cf.full(2, 1)
convert_to_blocks(d)
d <- lfebd3.cf.full(2, 1)
blk <- convert_to_blocks(d)
build_block_matrix(blk)
X <- build_effect_matrix(NULL, c(3, 3))
dim(X)
head(X)
d <- lfebd3.cf.full(2, 1)
```

```
blk <- convert_to_blocks(d)
res <- FactChar(c(3, 3), blk)
print(res)
res <- lfebd3_analyze(
  type = "lfebd3.cf.full",
  n = 2,
  r = 1,
  show_design = FALSE,
  run_analysis = FALSE
)
names(res)
```

`get_Tn_square` *Extract the square design matrix for a 3-level factorial construction*

Description

Computes the square submatrix used as the base design matrix for the LFBD generators.

Usage

```
get_Tn_square(n)
```

Arguments

`n`	Integer. Number of factors.

Value

A `3^n x n?` style square matrix derived from the full ternary construction, with the first `3^n` rows retained after transposition.

See Also

`generate_Tn_full()`, `lfebd3()`, `lfebd3.fr()`, `lfebd3.cf()`

Examples

```
T2 <- get_Tn_square(2)
stopifnot(nrow(T2) == 9)
```

lfebd3 *Generate a complete factorial LFBD*

Description

Creates the complete 3-level factorial layout from the ternary design construction and returns treatment labels in run order.

Usage

```
lfebd3(n)
```

Arguments

n	Integer. Number of factors.

Value

A data frame with columns:

Run Run labels such as "run1", "run2", ...

Treatment Treatment combination labels formed by concatenating factor levels coded as 0, 1, and 2.

See Also

lfebd3.fr(), lfebd3.cf.full(), lfebd3_analyze()

Examples

```
d <- lfebd3(2)
stopifnot(nrow(d) == 9, all(c("Run", "Treatment") %in% names(d)))
```

lfebd3.cf *Generate the principal block for a confounded LFBD*

Description

Extracts the principal block of size 3^r from a complete 3-level factorial design using the recursive confounded selection rule.

Usage

```
lfebd3.cf(n, r)
```

Arguments

n	Integer. Number of factors.
r	Integer. Exponent determining the principal block size, so the principal block contains 3^r treatment combinations.

Value

A matrix representing the principal block.

See Also

lfebd3.cf.full(), reduce_repeated()

Examples

```
B <- lfebd3.cf(3, r = 2)
stopifnot(nrow(B) == 9)
```

---

lfebd3.cf.full *Generate the full confounded factorial design*

Description

Builds the complete set of blocks induced by the principal block of a confounded 3-level factorial design.

Usage

```
lfebd3.cf.full(n, r, max_blocks_display = 12)
```

Arguments

n	Integer. Number of factors.
r	Integer. Exponent determining the principal block size.
max_blocks_display	Integer. Reserved display helper parameter from the original script.

Value

An invisible data frame whose first column identifies units and whose remaining columns correspond to blocks (blk1, blk2, ...).

See Also

lfebd3.cf(), convert_to_blocks(), FactChar()

Examples

```
d <- lfebd3.cf.full(2, r = 1)
stopifnot("Unit" %in% names(d))
```

`lfebd3.fr` *Generate a fractional 3-level LFBD*

Description

Constructs a fractional layout from the complete ternary design by repeated fractional reduction and returns the design together with defining contrasts and a low-order confounding summary.

Usage

```
lfebd3.fr(n, c)
```

Arguments

`n`	Integer. Number of factors.
`c`	Integer. Degree of fractionation, so the final number of runs is `3^(n - c)`.

Value

An object of class `"lfebd3_fr"` containing:

design A data frame with run labels and treatment labels.

design_matrix The reduced numeric design matrix.

defining_contrasts Character vector of defining contrast labels.

confounding_summary Data frame of main-effect and 2FI summaries.

See Also

`lfebd3()`, `lfebd3.cf()`, `lfebd3_analyze()`

Examples

```
d <- lfebd3.fr(2, 1)
d$design
d$defining_contrasts
```

`lfebd3_analyze` *Generate and analyze a 3-level LFBD in one call*

Description

Generates a complete, fractional, or confounded 3-level factorial block design and, optionally, computes design-characteristic diagnostics. The factor-level vector is inferred automatically as `rep(3, n)`.

Usage

```
lfebd3_analyze(
  type = c("lfebd3", "lfebd3.fr", "lfebd3.cf", "lfebd3.cf.full"),
  n,
  c = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE,
  fast = NULL,
  exact_limit = Inf,
  print_limit = 50
)
```

Arguments

`type`	Character string specifying the generator to use. Use `"lfebd3"` for complete factorial designs, `"lfebd3.fr"` for fractional factorial designs, and `"lfebd3.cf"` or `"lfebd3.cf.full"` for confounded factorial block designs.
`n`	Positive integer. Number of factors.
`c`	Optional non-negative integer. Degree of fractionation for `type = "lfebd3.fr"`.
`r`	Optional positive integer. Block-size exponent for `type = "lfebd3.cf"`; each block has `3^r` treatments.
`block_size`	Optional positive integer used to split run-ordered complete or fractional designs into consecutive blocks. Required for `type = "lfebd3"` and `type = "lfebd3.fr"`; ignored for confounded designs because they are generated in block-wise form.
`show_design`	Logical. Stored in the returned object and used by `print()` to decide whether to display the generated design.
`run_analysis`	Logical. If `TRUE`, compute design-characteristic diagnostics.
`fast`	Logical or `NULL`. If `NULL`, fast mode is used automatically when the number of treatments is greater than `exact_limit`. If `TRUE`, use `FactChar_fast()` and skip the slow exact C-matrix, all-effects, Hamming, and J2 diagnostics. If `FALSE`, use exact `FactChar()`.
`exact_limit`	Positive integer. Maximum number of treatments for which exact `FactChar()` is used automatically when `fast = NULL`.

print_limit	Positive integer. Maximum number of rows printed from large generated designs and alias tables by print(). Full objects are still returned invisibly.

Value

An object of class "lfebd3_analyze_result", a list with components type, factor_levels, generated_design, blocks, analysis, show_design, and show_confounding. Use print() to display a formatted report.

See Also

lfebd3(), lfebd3.fr(), lfebd3.cf.full(), FactChar()

Examples

```
res <- lfebd3_analyze(
  type = "lfebd3.fr",
  n = 3,
  c = 1,
  block_size = 9,
  show_design = FALSE,
  run_analysis = FALSE
)
stopifnot(inherits(res, "lfebd3_analyze_result"))
```

---

lfebd5 *Generate a complete 5-level factorial design*

Description

Creates the complete 5-level factorial layout for n factors. Factor columns are named F1, F2, and so on, and levels are coded as integers 0 through 4.

Usage

```
lfebd5(n)
```

Arguments

n	Positive integer. Number of factors.

Value

A data frame with 5^n rows and n factor columns.

Examples

```
d <- lfebd5(2)
stopifnot(nrow(d) == 25, ncol(d) == 2)
```

lfebd5.cf *Generate a confounded 5-level factorial block design*

Description

Generates a 5-level confounded factorial design with `5^n` treatments arranged in blocks of size `5^r`. The result is returned in block-wise form.

Usage

```
lfebd5.cf(n, r = 1, return_info = FALSE)
```

Arguments

`n`	Integer. Number of factors. Must be at least 3.
`r`	Positive integer. Block-size exponent; each block has `5^r` plots.
`return_info`	Logical. If `TRUE`, return a list containing the block design, principal block, and selection diagnostics.

Value

A data frame of class `"lfebd5_cf_design"` unless `return_info = TRUE`.

Examples

```
d <- lfebd5.cf(3, r = 1)
stopifnot(nrow(d) == 5, ncol(d) == 26)
```

lfebd5.cf_independent_confounding *Summarize independent confounding for a 5-level block design*

Description

Returns independent confounded effects and related counts for a 5-level confounded design.

Usage

```
lfebd5.cf_independent_confounding(n, r, p = 5)
```

Arguments

`n`	Integer. Number of factors.
`r`	Positive integer. Block-size exponent.
`p`	Integer modulus. Defaults to 5.

Value

A list with confounding summaries and generator matrix.

Examples

```
info <- lfebd5.cf_independent_confounding(3, r = 1)
stopifnot(info$block_size == 5)
```

`lfebd5.fr` *Generate a fractional 5-level factorial design*

Description

Generates a 5-level fractional factorial design by repeated row selection and attaches defining-factor and aliasing summaries as attributes.

Usage

```
lfebd5.fr(n, c = 1, return_info = FALSE)
```

Arguments

`n`	Integer. Number of factors. Must be at least 3.
`c`	Non-negative integer. Degree of fractionation; the design has `5^(n - c)` runs.
`return_info`	Logical. If `TRUE`, return a list containing the design and selection diagnostics instead of a classed design data frame.

Value

A data frame of class `"lfebd5_fr_design"` unless `return_info = TRUE`.

Examples

```
d <- lfebd5.fr(3, c = 1)
stopifnot(nrow(d) == 25)
attr(d, "defining_factors")
```

`lfebd5_analyze` *Generate and analyze a 5-level LFBD in one call*

Description

Generates a complete, fractional, or confounded 5-level factorial block design and, optionally, computes design-characteristic diagnostics. The factor-level vector is inferred automatically as `rep(5, n)` unless supplied.

Usage

```
lfebd5_analyze(
  type = base::c("lfebd5", "lfebd5.fr", "lfebd5.cf"),
  factor_levels = NULL,
  n,
  c = NULL,
  r = NULL,
  block_size = NULL,
  show_design = TRUE,
  run_analysis = TRUE,
  fast = NULL,
  exact_limit = 125,
  print_limit = 50
)
```

Arguments

`type`	Character string specifying the generator to use. Use `"lfebd5"` for complete factorial designs, `"lfebd5.fr"` for fractional factorial designs, and `"lfebd5.cf"` for confounded factorial block designs.
`factor_levels`	Optional integer vector. Must equal `rep(5, n)` when supplied.
`n`	Positive integer. Number of factors.
`c`	Optional non-negative integer. Degree of fractionation for `type = "lfebd5.fr"`.
`r`	Optional positive integer. Block-size exponent for `type = "lfebd5.cf"`; each block has `5^r` treatments.
`block_size`	Optional positive integer used to split run-ordered complete or fractional designs into consecutive blocks. Required for `type = "lfebd5"` and `type = "lfebd5.fr"`; ignored for confounded designs because they are generated in block-wise form.
`show_design`	Logical. Stored in the returned object and used by `print()` to decide whether to display the generated design.
`run_analysis`	Logical. If `TRUE`, compute design-characteristic diagnostics.
`fast`	Logical or `NULL`. If `NULL`, fast mode is used automatically when the number of treatments is greater than `exact_limit`. If `TRUE`, use `FactChar_fast()` and skip the slow exact C-matrix, all-effects, Hamming, and J2 diagnostics. If `FALSE`, use exact `FactChar()`.

exact_limit	Positive integer. Maximum number of treatments for which exact FactChar() is used automatically when fast = NULL.
print_limit	Positive integer. Maximum number of rows printed from large generated designs and alias tables by print(). Full objects are still returned invisibly.

Value

An object of class "lfebd5_analyze_result", a list with components type, factor_levels, generated_design, design_table, blocks, analysis, show_design, show_confounding, fast, and print_limit. Use print() to display a formatted report.

See Also

lfebd5(), lfebd5.fr(), lfebd5.cf(), FactChar()

Examples

```
res <- lfebd5_analyze(
  type = "lfebd5.fr",
  n = 3,
  c = 1,
  block_size = 25,
  show_design = FALSE,
  run_analysis = FALSE
)
stopifnot(inherits(res, "lfebd5_analyze_result"))
```

---

print.lfebd3_analysis *Print an lfebd3 analysis object*

Description

Displays the formatted report for an object returned by FactChar().

Usage

```
## S3 method for class 'lfebd3_analysis'
print(x, show_confounding = FALSE, ...)
```

Arguments

x	An object of class "lfebd3_analysis".
show_confounding	Logical. If TRUE, display the block-confounding summary.
...	Further arguments passed to or from other methods.

Value

Invisibly returns x.

Examples

```
d <- lfebd3.cf.full(2, 1)
res <- FactChar(c(3, 3), convert_to_blocks(d))
print(res)
```

`print.lfebd3_analyze_result` *Print a 3-level LFBD analysis result*

Description

Displays the generated design and analysis diagnostics stored in an object returned by `lfebd3_analyze()`. Printing is intentionally handled by this S3 method so that the constructor itself can be used silently in scripts, examples, and tests.

Usage

```
## S3 method for class 'lfebd3_analyze_result'
print(x, ...)
```

Arguments

`x`	An object returned by `lfebd3_analyze()`.
`...`	Further arguments passed to methods.

Value

Invisibly returns `x`.

Examples

```
res <- lfebd3_analyze(
  type = "lfebd3.fr",
  n = 3,
  c = 1,
  block_size = 9,
  show_design = FALSE,
  run_analysis = FALSE
)
print(res)
```

print.lfebd3_fr *Print a fractional 3-level LFBD object*

Description

Displays the run table, defining contrasts, and low-order aliasing summary stored in an object returned by `lfebd3.fr()`. Printing is handled by this S3 method so that the object can also be inspected programmatically.

Usage

```
## S3 method for class 'lfebd3_fr'
print(x, ...)
```

Arguments

`x`	Object of class `"lfebd3_fr"`.
`...`	Further arguments passed to methods.

Value

Invisibly returns `x`.

Examples

```
d <- lfebd3.fr(2, 1)
print(d)
```

print.lfebd5_analyze_result *Print a 5-level LFBD analysis result*

Description

Displays the generated design and analysis diagnostics stored in an object returned by `lfebd5_analyze()`. Printing is intentionally handled by this S3 method so that the constructor itself can be used silently in scripts, examples, and tests.

Usage

```
## S3 method for class 'lfebd5_analyze_result'
print(x, ...)
```

Arguments

`x`	An object returned by `lfebd5_analyze()`.
`...`	Further arguments passed to methods.

Value

Invisibly returns `x`.

Examples

```
res <- lfebd5_analyze(
  type = "lfebd5.fr",
  n = 3,
  c = 1,
  block_size = 25,
  show_design = FALSE,
  run_analysis = FALSE
)
print(res)
```

`print.lfebd5_cf_design` *Print a confounded 5-level design*

Description

Displays a confounded 5-level block design and its confounded-effect summaries.

Usage

```
## S3 method for class 'lfebd5_cf_design'
print(x, ...)
```

Arguments

`x`	Object of class `"lfebd5_cf_design"`.
`...`	Further arguments passed to `print()`.

Value

Invisibly returns `x`.

Examples

```
d <- lfebd5.cf(3, r = 1)
print(d)
```

print.lfebd5_fr_design *Print a fractional 5-level design*

Description

Displays a fractional 5-level design with defining factors and main-effect aliases.

Usage

```
## S3 method for class 'lfebd5_fr_design'
print(x, ...)
```

Arguments

x	Object of class "lfebd5_fr_design".
...	Further arguments passed to print().

Value

Invisibly returns x.

Examples

```
d <- lfebd5.fr(3, c = 1)
print(d)
```

Index