Package: cldreducer (via r-universe)

October 21, 2026

Title Reduce Compact Letter Displays

Version 0.2.0

Description Reduces compact letter displays (CLDs) of all pairwise comparisons. A CLD gives two groups a common letter exactly when they are not significantly different. The package finds a display with the fewest letter assignments that keeps every one of these relationships. This is the assignment-minimum clique covering problem of Ennis, Fayle, and Ennis (2012) <doi:10.1145/2133803.2275596>, solved as a mixed-integer linear program with 'HiGHS'. Among the displays with the fewest assignments, it returns one fixed canonical display, so the result does not depend on the solver. The package includes the wheat yield example of Piepho (2004) <doi:10.1198/1061860043515>.

License MIT + file LICENSE

Copyright See file COPYRIGHTS for example data sources and reuse status.

URL https://github.com/aigorahub/cld-reducer

BugReports https://github.com/aigorahub/cld-reducer/issues

Depends R (>= 4.0.0)

Imports highs (>= 1.14.0), Matrix

Suggests testthat (>= 3.1.7)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.0

LazyData true

NeedsCompilation no

Author John Ennis [aut, cre], Aigora [cph, fnd]

Maintainer John Ennis <john.m.ennis@aigora.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-21 14:50:13 UTC

RemoteUrl https://github.com/cran/cldreducer

RemoteRef HEAD

RemoteSha 9a2689eb9d65e1e71ef16012fa4769b501af57d1

Contents

piepho2004_wheat 2
reduce_letters 3
simple_abc_means 5
simple_abc_pairs 6

Index **7**

`piepho2004_wheat` *Piepho (2004) wheat yield example*

Description

All 190 pairwise comparisons of 20 wheat treatments (varieties) from the CIMMYT multi-environment yield trial analyzed by Piepho (2004) and reproduced in Table 7 of Ennis, Fayle, and Ennis (2012). The comparisons are "not significantly different" for 172 pairs. The standard (maximal) display has 4 letters and 56 letter assignments. The reduced display from `reduce_letters()` also has 4 letters, but only 44 assignments.

Usage

```
piepho2004_wheat
```

Format

A data frame with 190 rows and 3 columns:

group1 Label of the first treatment, a character string from `"1"` to `"20"`.

group2 Label of the second treatment.

significant Logical. `TRUE` when the two treatments differ significantly.

Source

Piepho, H.-P. (2004). An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics*, 13(2), 456-466. doi:10.1198/1061860043515, as tabulated in Ennis, Fayle, and Ennis (2012), Table 7. doi:10.1145/2133803.2275596

See Also

`reduce_letters()`

Examples

```
dim(piepho2004_wheat)
mean(!piepho2004_wheat$significant)
```

`reduce_letters` *Reduce a compact letter display*

Description

`reduce_letters()` and `reduce_from_adjacency()` find a compact letter display (CLD) with the fewest letter assignments in which two groups share a letter exactly when they are not significantly different. This is the assignment-minimum clique covering problem of Ennis, Fayle, and Ennis (2012). The programs are solved with 'HiGHS'. When several displays have the same, smallest number of assignments, the function returns the canonical one defined in the specification (`docs/algorithm.md` in the source repository), so the result does not depend on the solver. The same method is available in Python and JavaScript, and all three return the same display.

Usage

```
reduce_letters(
  pairs,
  means = NULL,
  group1 = "group1",
  group2 = "group2",
  significant = "significant",
  method = "assignment_minimum",
  time_limit = NULL,
  max_cliques = 10000L
)

reduce_from_adjacency(
  adjacency,
  groups = NULL,
  means = NULL,
  method = "assignment_minimum",
  time_limit = NULL,
  max_cliques = 10000L
)
```

Arguments

`pairs` A data frame with one row for every pair of groups. The columns named by `group1` and `group2` hold the two group labels, and the column named by `significant` tells whether the pair differs significantly. Significance can be logical, the numbers 0 and 1, or text such as `"yes"`, `"no"`, `"ns"`, `"significant"`. Every unordered pair of groups needs exactly one row. Labels are converted to character.

means
: Optional group means, used for the order of the letters (the letter that holds the highest mean comes first). Either a named numeric vector, or a data frame with the columns `group` and `mean` (otherwise its first two columns). In `reduce_letters()`, the order of the means is also the order of the groups; without means, the groups are in order of first appearance in `group1`, then `group2`. Every mean must be finite.

group1, group2, significant
: Names of the columns of `pairs`.

method
: The reduction method. `"assignment_minimum"` is the only one (`"assignment-minimum"` is also accepted).

time_limit
: One time budget in seconds for all solves of the call, or `NULL` for no limit.

max_cliques
: The largest number of maximal cliques to enumerate before stopping with an error, or `NULL` for no limit.

adjacency
: A square logical or 0/1 matrix. `adjacency[i, j]` is `TRUE` when groups `i` and `j` are not significantly different and must share a letter. It must be symmetric with `TRUE` on the diagonal.

groups
: Group labels for the rows of `adjacency`. The default is `"1"`, `"2"`, and so on.

Value

An object of class `cld_reduction`, a list with

- `letters`: a named character vector with the display of each group (the letters of a group are joined with no separator when every letter is one character, and with spaces after `Z`, when letters like `"AA"` appear);
- `assignments`: a named list with the letters of each group as a character vector;
- `stats`: a list with `assignments_before`, `assignments_after`, `reduction_pct` (not rounded), `num_letters_before`, `num_letters_after`, `num_groups`, `num_edges`, `solver_status`, and `objective`;
- `method`, `groups`, `relationship_preserved`, and `adjacency` (a logical matrix).

The `print()` method shows the display and the counts, and `as.data.frame()` gives a table with the columns `group`, `letters`, and `assignments`.

Errors

Malformed input stops with a condition of class `cldreducer_invalid_input`. Solver problems, an invalid `time_limit` or `max_cliques`, and too many cliques stop with a condition of class `cldreducer_solver_error`. Both also have the class `cldreducer_error`.

References

Ennis, J. M., Fayle, C. M., and Ennis, D. M. (2012). Assignment-minimum clique coverings. *ACM Journal of Experimental Algorithmics*, 17, Article 1.5. doi:10.1145/2133803.2275596

Examples

```
# Five groups; groups 1 to 3, 2 to 4, and 3 to 5 do not differ.
result <- reduce_letters(simple_abc_pairs, simple_abc_means)
result$letters
result$stats$assignments_after

# The same example from the matrix of non-significant pairs
adjacency <- diag(TRUE, 5)
adjacency[cbind(c(1, 1, 2, 2, 3, 3, 4), c(2, 3, 3, 4, 4, 5, 5))] <- TRUE
adjacency <- adjacency | t(adjacency)
reduce_from_adjacency(adjacency, means = simple_abc_means)
```

---

simple_abc_means *Simple example: group means*

Description

The group means that go with simple_abc_pairs. They fix the order of the groups and of the letters.

Usage

```
simple_abc_means
```

Format

A data frame with 5 rows and 2 columns: `group` (character labels `"1"` to `"5"`) and `mean` (numeric).

Source

Ennis, J. M., Fayle, C. M., and Ennis, D. M. (2012). Assignment-minimum clique coverings. *ACM Journal of Experimental Algorithmics*, 17, Article 1.5. doi:10.1145/2133803.2275596

See Also

simple_abc_pairs, `reduce_letters()`

Examples

```
simple_abc_means
```

`simple_abc_pairs` *Simple example: pairwise results of five groups*

Description

The five-group example of Ennis, Fayle, and Ennis (2012) in which the standard display `ABC` of one group reduces to `AC`. Groups 1 to 3, 2 to 4, and 3 to 5 do not differ significantly; all other pairs do.

Usage

```
simple_abc_pairs
```

Format

A data frame with 10 rows and 3 columns: `group1` and `group2` (character labels `"1"` to `"5"`) and `significant` (logical).

Source

Ennis, J. M., Fayle, C. M., and Ennis, D. M. (2012). Assignment-minimum clique coverings. *ACM Journal of Experimental Algorithmics*, 17, Article 1.5. doi:10.1145/2133803.2275596

See Also

simple_abc_means, `reduce_letters()`

Examples

```
simple_abc_pairs
reduce_letters(simple_abc_pairs, simple_abc_means)$letters
```

Index

∗ **datasets**
piepho2004_wheat, 2
simple_abc_means, 5
simple_abc_pairs, 6

piepho2004_wheat, 2

reduce_from_adjacency *(*reduce_letters*)*, 3
reduce_letters, 3
reduce_letters(), *2*, *5*, *6*

simple_abc_means, 5, *6*
simple_abc_pairs, *5*, 6