

# Package ‘codalm’

October 12, 2022

**Type** Package

**Title** Transformation-Free Linear Regression for Compositional Outcomes and Predictors

**Version** 0.1.2

**Maintainer** Jacob Fiksel <jfiksel@gmail.com>

**Description** Implements the expectation-maximization (EM) algorithm as described in Fiksel et al. (2021) <[doi:10.1111/biom.13465](https://doi.org/10.1111/biom.13465)> for transformation-free linear regression for compositional outcomes and predictors.

**License** GPL-2

**biocViews**

**Imports** SQUAREM (>= 2020.3), future, future.apply

**Encoding** UTF-8

**URL** <https://github.com/jfiksel/codalm>

**BugReports** <https://github.com/jfiksel/codalm/issues>

**RoxygenNote** 7.1.1

**Suggests** knitr, gtools, remotes, testthat, rmarkdown

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**LazyData** true

**NeedsCompilation** no

**Author** Jacob Fiksel [aut, cre] (<<https://orcid.org/0000-0001-7067-1334>>),  
Abhirup Datta [aut]

**Repository** CRAN

**Date/Publication** 2021-07-26 18:40:02 UTC

## R topics documented:

codalm . . . . .	2
codalm_ci . . . . .	3
codalm_indep_test . . . . .	4
educFM . . . . .	5
predict_codalm . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

codalm	<i>Transformation-free Linear Regression for Compositional Outcomes and Predictors</i>
--------	--

---

### Description

Implements the expectation-maximization (EM) algorithm as described in Fiksel et al. (2020) for transformation-free linear regression for compositional outcomes and predictors.

### Usage

```
codalm(y, x, accelerate = TRUE)
```

### Arguments

y	A matrix of compositional outcomes. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
x	A matrix of compositional predictors. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
accelerate	A logical variable, indicating whether or not to use the Squarem algorithm for acceleration of the EM algorithm. Default is TRUE.

### Value

A  $D_s \times D_r$  compositional coefficient matrix, where  $D_s$  and  $D_r$  are the dimensions of the compositional predictor and outcome, respectively

### References

<https://onlinelibrary.wiley.com/doi/full/10.1111/biom.13465>

### Examples

```
data("educFM")
father <- as.matrix(educFM[,2:4])
y <- father / rowSums(father)
mother <- as.matrix(educFM[,5:7])
x <- mother/rowSums(mother)
codalm(y, x)
```

---

codalm_ci	<i>Bootstrap Confidence Intervals Linear Regression for Compositional Outcomes and Predictors</i>
-----------	---

---

**Description**

Implements percentile based bootstrapping to estimate the confidence intervals for the regression coefficients when doing linear regression for compositional outcomes and predictors

**Usage**

```
codalm_ci(
  y,
  x,
  accelerate = TRUE,
  nboot = 500,
  conf = 0.95,
  parallel = FALSE,
  ncpus = NULL,
  strategy = NULL,
  init.seed = 123
)
```

**Arguments**

y	A matrix of compositional outcomes. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
x	A matrix of compositional predictors. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
accelerate	A logical variable, indicating whether or not to use the Squarem algorithm for acceleration of the EM algorithm. Default is TRUE
nboot	The number of bootstrap repetitions to use. Default is 500
conf	A scalar between 0 and 1 containing the confidence level of the required intervals. Default is .95.
parallel	A logical variable, indicating whether or not to use a parallel operation for computing the permutation statistics
ncpus	Optional argument. When provided, is an integer giving the number of clusters to be used in parallelization. Defaults to the number of cores, minus 1.
strategy	Optional argument. When provided, this will be the evaluation function (or name of it) to use for parallel computation (if parallel = TRUE). Otherwise, if parallel = TRUE, then this will default to multisession. See <a href="#">plan</a> .
init.seed	The initial seed for the permutations. Default is 123.

**Value**

A list, with ci\_L and ci\_U, giving the lower and upper bounds of each element of the B matrix

## Examples

```
data("educFM")
father <- as.matrix(educFM[,2:4])
y <- father / rowSums(father)
mother <- as.matrix(educFM[,5:7] )
x <- mother/rowSums(mother)
codalm_ci(y, x, nboot = 50, conf = .95)
```

---

codalm_indep_test	<i>Permutation Test for Linear Independence Between Compositional Outcomes and Predictors</i>
-------------------	---

---

## Description

Implements the loss function based permutation test as described in Fiksel et al. (2020) for a test of linear independence between compositional outcomes and predictors.

## Usage

```
codalm_indep_test(
  y,
  x,
  nperms = 500,
  accelerate = TRUE,
  parallel = FALSE,
  ncpus = NULL,
  strategy = NULL,
  init.seed = 123
)
```

## Arguments

y	A matrix of compositional outcomes. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
x	A matrix of compositional predictors. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized
nperms	The number of permutations. Default is 500.
accelerate	A logical variable, indicating whether or not to use the Squarem algorithm for acceleration of the EM algorithm. Default is TRUE.
parallel	A logical variable, indicating whether or not to use a parallel operation for computing the permutation statistics
ncpus	Optional argument. When provided, is an integer giving the number of clusters to be used in parallelization. Defaults to the number of cores, minus 1.

`strategy` Optional argument. When provided, this will be the evaluation function (or name of it) to use for parallel computation (if `parallel = TRUE`). Otherwise, if `parallel = TRUE`, then this will default to `multisession`. See [plan](#).

`init.seed` The initial seed for the permutations. Default is 123.

**Value**

The p-value for the independence test

**Examples**

```
require(gtools)
x <- rdirichlet(100, c(1, 1, 1))
y <- rdirichlet(100, c(1, 1, 1))
codalm_indep_test(y, x)

data("educFM")
father <- as.matrix(educFM[,2:4])
y <- father / rowSums(father)
mother <- as.matrix(educFM[,5:7] )
x <- mother/rowSums(mother)
codalm_indep_test(y, x)
```

---

educFM	<i>education level of father (F) and mother (M)</i>
--------	---

---

**Description**

Education level of father (F) and mother (M) in percentages of low (l), medium (m), and high (h) of 31 countries in Europe.

**Usage**

```
educFM
```

**Format**

A data frame with 31 observations (rows) and 7 columns (country and education level)

**Details**

- country community code
- F.l percentage of females with low education level
- F.m percentage of females with medium education level
- F.h percentage of females with high education level

- F.l percentage of males with low education level
- F.m percentage of males with medium education level
- F.h percentage of males with high education level

### Source

from robCompositions R package, [https://cran.r-project.org/src/contrib/Archive/robCompositions/robCompositions\\_2.3.0.tar.gz](https://cran.r-project.org/src/contrib/Archive/robCompositions/robCompositions_2.3.0.tar.gz), which used the dataEurostat, <https://ec.europa.eu/eurostat/>

---

predict_codalM	<i>Prediction for Transformation-free Linear Regression for Compositional Outcomes and Predictors</i>
----------------	---

---

### Description

Obtains compositional predictions for new compositional covariates using an established codalm model.

### Usage

```
predict_codalM(object, newx)
```

### Arguments

object	A codalm model
newx	A matrix of compositional predictors. Each row is an observation, and must sum to 1. If any rows do not sum to 1, they will be renormalized

### Value

A  $D_s \times D_r$  compositional coefficient matrix, where  $D_s$  and  $D_r$  are the dimensions of the compositional predictor and outcome, respectively

### Examples

```
data("educFM")
father <- as.matrix(educFM[,2:4])
y <- father / rowSums(father)
mother <- as.matrix(educFM[,5:7])
x <- mother / rowSums(mother)
codalm_model <- codalm(y[1:20,], x[1:20,])
predict_codalM(codalm_model, x[-(1:20),])
```

# Index

## \* datasets

educFM, 5

codalm, 2

codalm\_ci, 3

codalm\_indep\_test, 4

educFM, 5

plan, 3, 5

predict\_codalm, 6