

Package ‘FunCZIDM’

September 19, 2025

Title Functional Concurrent Zero-Inflated Dirichlet Multinomial Regression

Version 0.1.0

Description This package contains functions to sample from FunC-ZIDM regression model, and has the nessicary functions to process the samples.

License `use_mit_license()`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LazyData true

LinkingTo Rcpp, RcppArmadillo, splines, abind, MCMCpack

Imports Rcpp, RcppArmadillo, splines, abind, MCMCpack, shiny, reticulate, shinyMatrix, shinyjqui, rhandsontable

NeedsCompilation yes

Author Brody Erlandson [aut, cre]

Maintainer Brody Erlandson <erlandsonbrody@gmail.com>

Depends R (>= 3.5.0)

Suggests knitr,
rmarkdown,
ggtern

VignetteBuilder knitr

R topics documented:

calcAlphaDiv	2
calcDeltaAlphaDiv	2
calcDeltaRA	3
calcRA	3
combineOutputs	4
FunCZIDM	4
generateData	8
getBetaFunctions	9
getPropNullInCI	9
runPlotApp	10
Index	11

calcAlphaDiv	<i>Calculate α-diversity Statistic</i>
--------------	--

Description

Calculates the $\alpha_p(t)$ statistic using the Hill diversity index from the FunCZIDM samples.

Usage

```
calcAlphaDiv(output, l = 0, covProfile = NULL)
```

Arguments

output	(list) Output list from FunCZIDM
l	(numeric, default=0) Parameter l the diversity weight parameter. If 0, the Shannon diversity index is calculated, if 1, the count diversity index is calculated. Closer to 1 gives more weight to rare categories.
covProfile	(numeric vector, default=NULL) Covariate profile. If NULL, the baseline profile is used (ie all zeros).

Value

A matrix of $\alpha_p(t)$ samples

calcDeltaAlphaDiv	<i>Calculate the Multiplicative Change in α-diversity Statistic</i>
-------------------	---

Description

Calculates the $\Delta_v \alpha_p(t)$ statistic using the Hill diversity index from the FunCZIDM samples.

Usage

```
calcDeltaAlphaDiv(output, change, forCovs, l = 0, covProfile = NULL)
```

Arguments

output	(list) Output list from FunCZIDM
change	(numeric vector) The amount change of the covariate.
forCovs	(numeric vector) Indices of covariates for which $\Delta_v \alpha_p(t)$ to be calculated.
l	(numeric, default=0) Parameter l the diversity weight parameter. If 0, the Shannon diversity index is calculated, if 1, the count diversity index is calculated. Closer to 1 gives more weight to rare categories.
covProfile	(numeric vector, default=NULL) Covariate profile. If NULL, the baseline profile is used (ie all zeros).

Value

A list of matrix of $\Delta_v \alpha_p(t)$ samples

calcDeltaRA

*Calculate the Multiplicative Change in Relative Abundance***Description**

This function calculates the $\Delta_v RA_{jp}(t)$ statistic from the samples of FunCZIDM.

Usage

```
calcDeltaRA(output, change, forCovs, covProfile = NULL, forCats = NULL)
```

Arguments

output	(list) Output list from FunCZIDM
change	(numeric vector) The change of the covariate.
forCovs	(integer vector) Indices of covariates for which $\Delta_v RA_{jp}(t)$ to be calculated.
covProfile	(numeric vector, default=NULL) Covariate profile. If NULL, the baseline profile is used (ie all zeros).
forCats	(integer vector, default=NULL) Indices of categories for which $\Delta_v RA_{jp}(t)$ is calculated. If NULL, then all categories are used.

Value

A list of 3-dimensional array $\Delta_v RA_{jp}(t)$ samples

calcRA

*Calculates the Relative Abundance***Description**

This function calculates the $RA_j(t)$ from the from the output of FunCZIDM

Usage

```
calcRA(output, covProfile = NULL)
```

Arguments

output	(list) Output list from FunCZIDM
covProfile	(numeric vector, default=NULL) Covariate profile (default is baseline, i.e. all zeros).

Value

A 3-dimensional array of $RA_j(t)$ samples.

combineOutputs	<i>Combine FunCZIDM Outputs</i>
----------------	---------------------------------

Description

Combines multiple FunCZIDM output files into a single output list.

Usage

```
combineOutputs(
  outputFiles,
  saveToFile = TRUE,
  fileName = "combinedOutputs.rds",
  compress = "xz"
)
```

Arguments

outputFiles	(Character vector) Character vector of file paths to the saved outputs.
saveToFile	(logical, default=TRUE) Whether to save output to file.
fileName	(character, default="combinedOutputs.rds") File name for output.
compress	(character, default="xz") If saveToFile is TRUE, the compression method for saving the output file. Options are "xz", "gzip", or "bzip2".

Value

If saveToFile is TRUE, returns NULL after saving the combined output to file. If saveToFile is FALSE, returns the combined output list.

FunCZIDM	<i>FunCZIDM</i>
----------	-----------------

Description

This function is used to sample from the FunC-ZIDM regression model.

Usage

```
FunCZIDM(
  counts,
  covariates,
  ids,
  varyingCov,
  iter = 10000,
  burnIn = 5000,
  thin = 1,
  adjustFreq = 250,
  ZIGrouped = TRUE,
  returnBurnIn = FALSE,
```

```

printProgress = TRUE,
toReturn = NULL,
betaInitial = NULL,
rInitial = NULL,
priors = NULL,
proposalVars = NULL,
covWithoutVC = NULL,
df = 4,
degree = 3,
basisFunc = splines::bs,
saveToFile = TRUE,
fileName = "output.rds",
saveNullInBetaCI = TRUE,
nullInBetaCIFileName = "nullInBetaCI.csv"
)

```

Arguments

counts	(integer matrix) Each categories counts for each sample.
covariates	(data.frame) The covariates used for the regression, should not include the intercept. These should correspond to the columns of counts.
ids	(integer vector) The index of subject ids in the 'covariates'.
varyingCov	(numeric vector) The index of the covariate the varying coefficients will be a function of in covariates. This is usually time.
iter	(integer, default=10000) Total number of iterations.
burnIn	(integer, default=5000) Number of burn-in iterations, must be less than iter.
thin	(integer, default=1) Number of iterations to thin by.
adjustFreq	(numeric, default=250) Frequency at which β_{jpd}^* proposals are adjusted.
ZIGrouped	(logical, default=TRUE) Whether to use the zero inflation indicator for all samples of an id (TRUE), or each sample have their own indicator (FALSE).
returnBurnIn	(logical, default=FALSE) Whether to return burn-in samples.
printProgress	(logical, default=TRUE) Whether to print progress bar.
toReturn	(vector, default=NULL) Vector of parameter names to return in addition to the default parameters. The default parameters are: <ul style="list-style-type: none"> "beta": The β_{jpd}^* coefficients. "betaAcceptProp": The acceptance proportion for the β_{jpd}^* proposals. "rAcceptProp": The acceptance proportion for the r_{jp} proposals. "rMeans": The mean of the r_{jp} coefficients. "etaMeanPropZeros": The average level of zero-inflation indicated.

Returnable parameters are:

- "eta": The zero-inflation indicator.
- "c": The zero-inflation concentration parameter.
- "u": The auxiliary variable.
- "r": The individual- and category-specific coefficients.
- "phi": The individual- and category-specific coefficients sd
- "lambda": The local parameter of the horseshoe prior.
- "nu": The local parameter auxiliary variable of the horseshoe prior.

	<ul style="list-style-type: none"> • "tau": The global parameter of the horseshoe prior. • "xi": The global parameter auxiliary variable of the horseshoe prior. • "RA": The mean estimated relative abundance of each sample.
	If NULL, only the default parameters are returned.
betaInitial	(matrix, default=NULL) Initialization values for β_{jpd}^* . If NULL, the initial values will be drawn from $\text{Unif}(-.75, .75)$.
rInitial	(matrix, default=NULL) Initialization values for r_{jp} . If NULL, the initial values will be drawn from $\text{Unif}(-.05, .05)$.
priors	<p>(list, default=NULL) List with prior hyperparameters specifications. The hyperparameters that can be specified are:</p> <ul style="list-style-type: none"> • "first beta sd": The prior sd for the first β_{jpd}^* coefficient (β_{j00}^*). • "a": The prior shape parameter for the variance of the r_{jp}. • "b": The prior rate parameter for the variance of the r_{jp}. • "alpha": The prior first shape parameter for the probability of the zero-inflation indicator. • "beta": The prior second shape parameter for the probability of the zero-inflation indicator. • "kappaShape": The prior shape parameter for the κ_{jp}, i.e. the student-t portion of the regularized horseshoe. • "kappaRate": The prior rate parameter for the κ_{jp}, i.e. the student-t portion of the regularized horseshoe. <p>NULL gives the default values, which are:</p> <ul style="list-style-type: none"> • "first beta sd": 1 • "a": 3 • "b": 9 • "alpha": .01 • "beta": 10 • "kappaShape": 100 • "kappaRate": 900
proposalVars	<p>(list, default=NULL) List with proposal sd for β_{jpd}^* and r_{jp}. If NULL, the default proposal sd are used, which are:</p> <ul style="list-style-type: none"> • "beta proposal sd": .1 • "r proposal sd": 1
covWithoutVC	(vector, default=NULL) Indices of the covariates that will not have varying coefficients. If NULL, all covariates will have varying coefficients. Intercept will always have varying coefficients.
df	(numeric, default=4) Degrees of freedom for the basis function.
degree	(numeric, default=3) Degree for the basis function.
basisFunc	(function, default=splines::bs) Function to generate basis functions.
saveToFile	(logical, default=TRUE) Whether to save output to file.
fileName	(character, default="output.rds") File name for output.
saveNullInBetaCI	(logical, default=TRUE) Whether to a csv that contains the proportion of the $\beta_{jp}(t)$ coefficients that contain the null value in their credible interval. Gives an idea of which covariate and category combinations may have meaningful effects.
nullInBetaCIFilename	(character, default="nullInBetaCI.csv") File name for null CI statistics.

Value

Depending on `saveToFile`, either returns NULL after saving or an output list. The output list contains:

- "beta": The β_{jpd}^* samples.
- "betaAcceptProp": The acceptance proportion for the β_{jpd}^* proposals after burn-in.
- "rAcceptProp": The acceptance proportion for the r_{jp} proposals after burn-in.
- "rMeans": The mean of the r_{jp} samples.
- "etaMeanPropZeros": The average level of zero-inflation indicated.
- "interiorKnots": The interior knots of the basis functions.
- "boundaryKnots": The boundary knots of the basis functions.
- "df": The degrees of freedom used for the basis functions.
- "basisFunc": The basis function used.
- "varyingCov": The values of the covariate that the varying coefficients are a function of.
- "colMapping": A list mapping the covariate indices to the column names of the design matrix.
- "XvartoXColMapping": A vector mapping the columns of the design matrix with the basis design matrix.
- "catNames": The category names from the counts.
- "centerScaleList": A list containing the center and scale of each continuous covariate. If a covariate is a factor, then "no c/s" is stored.

See Also

[calcRA](#), [calcDeltaRA](#), [calcAlphaDiv](#), [calcDeltaAlphaDiv](#), [FunCZIDM-package](#)

Examples

```
## Not run:
# toy example with included data
library(FunCZIDM)

data(infantData)
# Convert the categorical variables to factors
infantCovariates$gender <- as.factor(infantCovariates$gender)
infantCovariates$mode.of.birth <- as.factor(infantCovariates$mode.of.birth)
infantCovariates$Room.category <- as.factor(infantCovariates$Room.category)
infantCovariates$milk <- as.factor(infantCovariates$milk)
infantCovariates$milk <- relevel(infantCovariates$milk, ref="< 10%")
infantCovariates$Period.of.study <- as.factor(infantCovariates$Period.of.study)
infantCovariates$Period.of.study <- relevel(infantCovariates$Period.of.study,
                                           ref="before")

# getting the index of the subject id's and the time-varying variable
idIdx <- which(colnames(infantCovariates) == "Subject")
timeIdx <- which(colnames(infantCovariates) == "Day.of.life.sample.obtained")
# extracting the subject ids and tv variable
ids <- infantCovariates[, idIdx]
time <- infantCovariates[, timeIdx]
infantCovariates <- infantCovariates[, -c(idIdx, timeIdx), drop = FALSE]

idIdx <- which(colnames(infantCounts) == "Subject")
```

```

timeIdx <- which(colnames(infantCounts) == "Day.of.life.sample.obtained")
infantCounts <- infantCounts[, -c(idIdx, timeIdx), drop = FALSE]
infantCounts <- as.matrix(infantCounts)

output <- FunCZIDM(counts=infantCounts,
                   covariates=infantCovariates,
                   ids=ids,
                   varyingCov=time,
                   iter=85,
                   burnIn=75,
                   thin=10,
                   covWithoutVC=c(which(grepl("Period", colnames(infantCovariates)))),
                   saveToFile=FALSE,
                   saveNullInBetaCI=FALSE)

## End(Not run)

```

generateData

Generate Data

Description

Generates synthetic data for testing the model.

Usage

```
generateData(n, c, p, totalCountRange = c(100, 200), numActive = 4)
```

Arguments

n	(integer) Number of individuals.
c	(integer) Number of categories.
p	(integer) Number of functional covariates.
totalCountRange	(numeric vector, default=c(100, 200)) Range of total counts.
numActive	(integer, default=4) Number of categories with active covariates. Always 1:numActive categories are active.

Value

A list containing the generated data: counts, covariates, ids, timepoints and the true parameters.

getBetaFunctions	<i>Get $\beta_{jp}(t)$ Functions</i>
------------------	---

Description

Extracts the $\beta_{jp}(t)$ function samples from the output of FunCZIDM

Usage

```
getBetaFunctions(output, cov = NULL)
```

Arguments

output	(list) Output list returned from FunCZIDM
cov	(integer, default=NULL) The index of covariates from FunCZIDM for which $\beta_{jp}(t)$ functions to extract. Intercept if NULL.

Value

A 3-dimensional array of $\beta_{jp}(t)$ function samples for the specified covariate.

getPropNullInCI	<i>Get Proportion of 0 in 95% CI of $\beta_{jp}(t)$</i>
-----------------	--

Description

Calculates the proportion of $\beta_{jp}(t)$ functions that have the zero function in their 95% credible interval.

Usage

```
getPropNullInCI(output, varCovRange = NULL)
```

Arguments

output	(list) Output list from FunCZIDM.
varCovRange	(numeric vector, default=NULL) Range of the varying covariate to evaluate the $\beta_{jp}(t)$ functions. By default, the range is the min and max of the varying covariate in the data.

Value

A matrix with columns "category", "covariate", and "Proportion 0 in 95 CI"

runPlotApp	<i>Runs the Shiny App for Visualization</i>
------------	---

Description

Runs the Shiny app for visualizing the output of FunCZIDM.

Usage

```
runPlotApp(useCondaEnv = FALSE, ...)
```

Arguments

useCondaEnv	(logical, default=FALSE) Whether to use the conda environment for running the app. If TRUE, the conda environment named "shinyPy" will be used. If FALSE, the python from reticulate will be used.
...	Additional arguments to pass to shinyAppDir.

Index

`calcAlphaDiv`, [2](#), [7](#)
`calcDeltaAlphaDiv`, [2](#), [7](#)
`calcDeltaRA`, [3](#), [7](#)
`calcRA`, [3](#), [7](#)
`combineOutputs`, [4](#)

`FunCZIDM`, [4](#)

`generateData`, [8](#)
`getBetaFunctions`, [9](#)
`getPropNullInCI`, [9](#)

`runPlotApp`, [10](#)