

# Package ‘FunCZIDM’

August 11, 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

**Imports** Rcpp, RcppArmadillo, splines, abind

**NeedsCompilation** yes

**Author** Brody Erlandson [aut, cre]

**Maintainer** Brody Erlandson <erlandsonbrody@gmail.com>

**Depends** R (>= 3.5.0)

## R topics documented:

calcAlphaDiv . . . . .	2
calcDeltaAlphaDiv . . . . .	2
calcDeltaRA . . . . .	3
calcRA . . . . .	3
combineOutputs . . . . .	4
FunCZIDM . . . . .	4
generateData . . . . .	7
getBetaFunctions . . . . .	7
<b>Index</b>	<b>8</b>

---

calcAlphaDiv	<i>Calculate <math>\alpha</math>-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 3-dimensional array of  $\alpha_p(t)$  samples

---

calcDeltaAlphaDiv	<i>Calculate The Multiplicative Change in <math>\alpha</math>-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, l = 0, covProfile = NULL, forCovs = NULL)
```

**Arguments**

output	(list) Output list from FunCZIDM
change	(numeric) The amount change of the covariate.
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).
forCovs	(numeric vector, default=NULL) Indices of covariates for which $\Delta_v \alpha_p(t)$ to be calculated.

**Value**

A list of 3-dimensional arrays of  $\Delta_v \alpha_p(t)$  samples

---

calcDeltaRA	<i>Calculate the Multiplicative Change in Relative Abundance</i>
-------------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

---

calcRA	<i>Calculates the Relative Abundance</i>
--------	--

---

**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"
)
```

**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.

**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,
  rCols = NULL,
  iter = 10000,
  burnIn = 5000,
  thin = 1,
  adjustFreq = 250,
  proposalCap = 0,
  ZIGrouped = TRUE,
  returnBurnIn = FALSE,
  printProgress = TRUE,
```

```

    toReturn = NULL,
    betaInitial = NULL,
    rInitial = NULL,
    priors = NULL,
    proposalVars = NULL,
    covWithVC = 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.
rCols	(integer vector, default=NULL) Indices of covariates with subject- and category-specific effects. By default, it will only be the intercept. For this model, the intercept always has subject- and category- specific effects, and does not need to be specified. Check the VarCoZIDM function for a model without any subject- and category- specific effects.
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 $\beta_{jpd}^*$ proposals are adjusted.
proposalCap	(numeric, default=0) Cap for proposal variances, if 0 there will be no cap.
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"> <li>"beta": The <math>\beta_{jpd}^*</math> coefficients.</li> <li>"betaAcceptProp": The acceptance proportion for the <math>\beta_{jpd}^*</math> proposals.</li> <li>"rAcceptProp": The acceptance proportion for the <math>r_{jp}</math> proposals.</li> <li>"rMeans": The mean of the <math>r_{jp}</math> coefficients.</li> <li>"etaMeanPropZeros": The average level of zero-inflation indicated.</li> </ul>

Returnable parameters are:

- "eta": The zero-inflation indicator.
- "c": The zero-inflation concentration parameter.

	<ul style="list-style-type: none"> <li>• "u": The auxiliary variable.</li> <li>• "r": The individual- and category-specific coefficients.</li> <li>• "phi": The individual- and category-specific coefficients sd</li> <li>• "lambda": The local parameter of the horseshoe prior.</li> <li>• "nu": The local parameter auxiliary variable of the horseshoe prior.</li> <li>• "tau": The global parameter of the horseshoe prior.</li> <li>• "xi": The global parameter auxiliary variable of the horseshoe prior.</li> <li>• "RA": The mean estimated relative abundance of each sample.</li> </ul>
	If NULL, only the default parameters are returned.
betaInitial	(matrix, default=NULL) Initialization values for $\beta_{jpd}^*$ . If NULL, the initial values will be drawn from Unif(-.75,.75).
rInitial	(matrix, default=NULL) Initialization values for $r_{jp}$ . If NULL, the initial values will be drawn from 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"> <li>• "first beta sd": The prior sd for the first <math>\beta_{jpd}^*</math> coefficient (<math>\beta_{j00}^*</math>).</li> <li>• "a": The prior shape parameter for the variance of the <math>r_{jp}</math>.</li> <li>• "b": The prior scale parameter for the variance of the <math>r_{jp}</math>.</li> <li>• "alpha": The prior first shape parameter for the probability of the zero-inflation indicator.</li> <li>• "beta": The prior second shape parameter for the probability of the zero-inflation indicator.</li> </ul> <p>NULL gives the default values, which are:</p> <ul style="list-style-type: none"> <li>• "first beta sd": 1</li> <li>• "a": 3</li> <li>• "b": 9</li> <li>• "alpha": .01</li> <li>• "beta": 10</li> </ul>
proposalVars	<p>(list, default=NULL) List with proposal sd for <math>\beta_{jpd}^*</math> and <math>r_{jp}</math>. If NULL, the default proposal sd are used, which are:</p> <ul style="list-style-type: none"> <li>• "beta proposal sd": .3</li> <li>• "r proposal sd": 1</li> </ul>
covWithVC	(vector, default=NULL) Indices of the covariates that will 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.

---

generateData	<i>Generate Data</i>
--------------	----------------------

---

**Description**

Generates synthetic data for testing the model.

**Usage**

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

**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.

**Value**

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

---

getBetaFunctions	<i>Get <math>\beta_{jp}(t)</math> 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.

# Index

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