

Package ‘ARZIMM’

February 3, 2020

Title Statistical modeling and inference of microbial interaction and stability for microbial dynamical systems

Version 0.0.0.9000

Author Linchen He, Huilin Li

Maintainer Linchen He <Linchen.He@nyulangone.org>

Description This package is developed to model microbial dynamical systems from longitudinal microbiome data and infer microbial interaction and stability. ARZIMM models the excess zero abundance and the non-zero abundances separately; and use a random effect model to borrow strength across subjects.

Depends R (>= 3.3.0), expm, ggplot2, glmnet, lme4, phyloseq, stringr

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.0.2.9000

LazyData true

URL <https://github.com/Hlch1992/ARZIMM>

BugReports <http://github.com/Hlch1992/ARZIMM/issues>

R topics documented:

ARZIMM-package	2
ARZIMM	3
ARZIMM.control	5
emnr	6
parameters	7
phyExample	8
sampladata	9
simMixTime	9

Index	11
--------------	-----------

ARZIMM-package

ARZIMM: Statistical modeling and inference of microbial interaction and stability for microbial dynamical systems

Description

This package is developed to model microbial dynamical systems from longitudinal microbiome data and infer microbial interaction and stability.

Details

Package: ARZIMM
Type: Package
Version: 0.0.0.9000
Date: 2020-02-02
License: GPL (>= 2)

Author(s)

Linchen He, Huilin Li Maintainer: Linchen He <Linchen.He@nyulangone.org>

References

Modeling and inference of microbial interaction and stability on the longitudinal microbiome data

Examples

```
## Not run:
require(ARZIMM),
data(sampleparameters)
set.seed(1234)
simMixTime(baseFdataN2M=parameters$baseFdataN2M, conFdataN2L=parameters$conFdataN2L, timeN=parameters$timeN,
interceptM=parameters$interceptM, betaMM=parameters$betaMM, gammaLM=parameters$gammaLM,
sigmaM=parameters$sigmaM, biN2M=parameters$biN2M)

data(sampleddata)
Varname=colnames(sampleddata)[1:20]
Conname=colnames(sampleddata)[21:26]
Tname=colnames(sampleddata)[27]
IDname=colnames(sampleddata)[28]
ARZIMM::ARZIMM(Varname = Varname, Conname = Conname, fdata = sampleddata,
IDname = IDname, Tname = Tname, bootpara=list(bootpval=TRUE, nboot=100))

## End(Not run)
```

ARZIMM

*Auto-Regressive Zero-Inflated Mixed Model***Description**

This function allows you to fit the ARZIMM.

Usage

```
ARZIMM(
  phy,
  Varname,
  Conname = NA,
  Covname = NA,
  IDname = "ID",
  Tname = "Time",
  fdata = NULL,
  family = "Poisson",
  ...
)
```

Arguments

phy	A phyloseq-class experiment-level object example data. See \DataphyExample
Varname	a vector of character string indicating the taxa names in the non-zero auto-regressive model
Conname	a vector of character string indicating the concomitant variable names in the zero state logit model
Covname	a vector of character string indicating the covariate names in the non-zero auto-regressive model
IDname	a character string indicating the subject ID. Default is ID
Tname	a character string indicating the time variable. Default is Time
fdata	a data frame containing all variables to be analyzed
family	a character string indicating the distribution. default is Poisson
...	See ARZIMM.control

Value

an object of class "ARZIMMObject" is returned, which is a list with the ingredients of fit.

nwttable	the matrix of network table of fit
mseest	a list of mean square error: rmseest the root square of mean standard error rmseest the root square of mean pearson standardized error
paralist	a list of parameter estimates: beta the matrix of fixed effects for the non-zero auto-regressive model gamma the matrix of fixed effects for the zero state logit model

	sigma the measurement error standard deviation for both the non-zero auto-regressive model and the zero state logit model
	ciest the estimated random effects part of both the non-zero auto-regressive model and the zero state logit model
runtime	running time of the program
datalist	an object of class "ARZIMMData"
resultall	a list of parameter estimates of the fits with ingredients of the lambda
bootparapval	a list of p values obtain via bootstrap with componenets: betapval a vector of p values of fixed effects for the non-zero auto-regressive model gammapval a vector of p values of fixed effects for the zero state logit model sigmapval a vector of p values of the measurement error standard deviation for both the non-zero auto-regressive model and the zero state logit model
tunlist	the values of parameters used in the fits with components: lambdabeta the values of lambda used in the non-zero auto-regressive model lambdagamma the values of lambda used in the zero state logit model weight1all observation weights used in the non-zero auto-regressive model
parasetup	a list of parameters used to initial the ARZIMM program with components: initpara a list of initial parameter inputs; if the inputs are absent, default values are included lambda the values of lambda used in the fits. tunpara a list of tuning parameter inputs; if the inputs are absent, default values are included selectpara a list of selection parameter inputs; if the inputs are absent, default values are included

See Also

ARZIMM.control

Examples

```
data(phyExample)

Varname=colnames(otu_table(phyExample))
Conname=colnames(sample_data(phyExample))[1:6]
Tname=colnames(sample_data(phyExample))[7]
IDname=colnames(sample_data(phyExample))[8]

ARZIMMresult=ARZIMM::ARZIMM(phyExample,Varname = Varname,Conname = Conname,
IDname = IDname,Tname = Tname,bootpara=list(bootpval=TRUE,nboot=5))

data(sampledata)

Varname=colnames(sampledata)[1:20]
Conname=colnames(sampledata)[21:26]
Tname=colnames(sampledata)[27]
IDname=colnames(sampledata)[28]
```

```
ARZIMMresult=ARZIMM::ARZIMM(phy=NULL,Varname = Varname,Conname = Conname,fdata=sampledData,
IDname = IDname,Tname = Tname,bootpara=list(bootpval=TRUE,nboot=5))
```

ARZIMM.control

internal ARZIMM parameters

Description

View and/or change the factory default parameters in ARZIMM

Usage

```
ARZIMM.control(
  initpara = list(betaini = NULL, gammaini = NULL, sigmaini = NULL, weight1all = NULL,
    weight2all = NULL, calini = TRUE, iniw = FALSE),
  tunpara = list(lambdaseq1 = NULL, lambdaseq2 = NULL, ntun = 50, epsilon = c(50000, 5)),
  selectpara = list(selgamma = TRUE, selcri = "BIC", pen = "adalasso"),
  bootpara = list(bootpval = TRUE, nboot = 500)
)
```

Arguments

initpara	<p>a named list of user-specified initial values:</p> <p>betaini the matrix of fixed effects for the non-zero auto-regressive model</p> <p>gammaini the matrix of fixed effects for the zero state logit model</p> <p>sigmaini the measurement error standard deviation for both the non-zero auto-regressive model and the zero state logit model</p> <p>weight1all observation weights for the non-zero auto-regressive model. Default is 1 for each observation</p> <p>weight2all observation weights for the zero state logit model. Default is 1 for each observation</p> <p>calini=T logical; should initial parameters be calculated. Default is Ture</p> <p>iniw logical; should observation weights be calculated according to initial parameters. Default is False When this list of initial values does not contain some of these components or contains components not of the appropriate length, then the default initial values are used instead.</p>
tunpara	<p>a list of control values with components:</p> <p>lambdaseq1 a user supplied lambda sequence for the non-zero auto-regressive model. Typical usage is to have the program compute its own lambda sequence. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda</p> <p>lambdaseq2 a user supplied lambda sequence for the zero state logit model. Typical usage is to have the program compute its own lambda sequence. Supplying a value of lambda overrides this. WARNING: use with care. Avoid supplying a single value for lambda</p> <p>ntun the number of lambda values - default is 50</p> <p>epsilon the range of lambda values; default= c(5e4,5)</p>
selectpara	<p>a list of control values with components:</p>

selgamma logical; should concomitant variables in the zero state logit model be selected

selcri method to be use for variable selection. Currently three options. The default is BIC. Other choices are AIC and CV

pen penalty; default is 'adalasso'

bootpara a list of control values with components:

bootpval logical; should p value be calculated using bootstrap

nboot the number of bootstrap simulations; default is 500

Details

If called with no arguments, `ARZIMM.control()` returns a list with the current settings of these parameters. Any arguments included in the call sets those parameters to the new values, and then silently returns. The values set are persistent for the duration of the R session.

Value

A list with named elements as in the argument list

Author(s)

Linchen He
 Maintainer: Linchen He <Linchen.He@nyulangone.org>

See Also

ARZIMM

Examples

```
ARZIMM.control(bootpara=list(bootpval=FALSE))
ARZIMM.control(selectpara=list(selgamma=FALSE,pen='lasso'))
```

emnr

EM algorithm

Description

EM algorithm used in the ARZIMM model.

Usage

```
emnr(data, para, weight, family, selgamma = FALSE)
```

Arguments

data	a list of data with componenets: yFdata a vector of xFdata a matrix of variables corresponding to the non-zero auto-regressive model conFdata a matrix of covariates corresponding to the zero state logit model group a vector of numbers as group indicator
para	a list of parameter estimates: beta initial value for beta gamma initial value for gamma sigma initial value for sigma
weight	a vector of observation weightsfor both the non-zero auto-regressive model and the zero state logit model
family	a character string indicating the distribtuion. default is Poisson
selgamma	logical; should concomitant variables in the zero state logit model be selected

Value

a list of fits	
para	a list of parameter estimates: beta beta estimates gamma gamma estimates sigma sigma estimates
ciestm	the estimated random effects
conv	logical; did the algorithm converged
df	the number of non-zero parameter estimates
bic	a vector of BIC, AIC, and log likelihood
mse	a vector of square root of mean (pearson) standard error
lambda	a vector of lambda sequence

parameters

A list of parameters for simulation

Description

This data is a list object which contains the baseline OTU table, sample data, time number and parameters beta matrix gamma matrix sigma and random effects. Details see reference.

Usage

```
data("parameters")
```

Format

A list with 8 objects as following.

baseFdataN2M a numeric matrix containning baseline OTU table

conFdataN2L a numeric matrix containning time-independent concomitant variables

timeN a numeric indicating total number of time points to simulate

interceptM a numeric vector indicating intercept of non-zero autoregressive model

betaMM a numeric matrix indicating the interaction matrix

gammaLM a numeric matrix indicating fixed effects for the zero state logit model

sigmaM a numeric vector indicating the measurement error standard deviation for the non-zero auto-regressive model

biN2M a numeric vector indicating the random effects part of the non-zero auto-regressive model

Examples

```
data(parameters)
```

```
phyExample
```

A phyloseq-class experiment-level object example data.

Description

This data is a phyloseq-class object which contains the OTU table, sample data, taxonomy table and phylogenetic tree information. Details about the phyloseq class please see the R package "phyloseq".

Usage

```
data("phyExample")
```

Format

A phyloseq-class experiment-level object

otu_table() OTU Table: [20 taxa and 1050 samples]

..@ .Data: a matrix where each row represents the subject,each column represents the OTU.

..@ taxa_are_rows: logi FALSE

sample_data() Sample Data: [1050 samples by 8 sample variables]

A data frame with 1050 observations for six covariates w1...w6 and two variables time and subjid.

w1...w6: numeric vectors as binary covariates;

time: a numeric vector indicating the time of sampling;

subjid: a string vector indicating which subject the sample belonging to.

Examples

```
data(phyExample)
```

sampledata	<i>a data frame containing all variables to be analyzed</i>
------------	---

Description

This is a data frame object which contains OTU varalbes, concomitant variables, time variable and subject ID.

Usage

```
data("sampledata")
```

Format

A data frame with 1050 observations on the following 28 variables.

M1...M20 numeric vectors indicating the absolute abundance of taxa

w1...W6 numeric vectors indicating the value of concomitant variables

time a numeric vector indicating the time of sampling

subjid a factor indicating the subject IDs

Examples

```
data(sampledata)
```

simMixTime	<i>A simulation function</i>
------------	------------------------------

Description

Simulation for the ARZIMM model.

Usage

```
simMixTime(
  baseFdataN2M,
  conFdataN2L,
  timeN,
  interceptM,
  betaMM,
  gammaLM = NULL,
  sigmaM,
  biN2M = NULL,
  family = "Poisson"
)
```

Arguments

baseFdataN2M	a matrix of absolute counts at baseline time
conFdataN2L	a matrix of concomitant variables
timeN	a vector of the number of time points for each subjects
interceptM	a vector of the intercepts of the non-zero autoregression model
betaMM	a matrix of network table
gammaLM	a matrix of the zero state logit model
sigmaM	a vector of the measurement error standard deviation for the non-zero autoregression model
biN2M	a vector of the random effects for the non-zero autoregression model
family	a character string indicating the distribution. default is Poisson

Value

a list of simulations:

otu.tab	a data frame of OTU table
convar	a data frame of concomitant variables
subjid	a vector of subject-time IDs
df	the number of non-zero parameter estimates
zi	a list of the random effects

Examples

```
require(ARZIMM)
data(parameters)

set.seed(1234)
sim.otu.tab=simMixTime(baseFdataN2M=parameters$baseFdataN2M,conFdataN2L=parameters$conFdataN2L,
timeN=parameters$timeN,interceptM=parameters$interceptM,betaMM=parameters$betaMM,
gammaLM=parameters$gammaLM,sigmaM=parameters$sigmaM,biN2M=parameters$biN2M)

fdata=cbind(sim.otu.tab[[1]],sim.otu.tab[[2]],subjid=sim.otu.tab[[3]])
Vname=colnames(sim.otu.tab[[1]])
Conname=colnames(sim.otu.tab[[2]])[-ncol(sim.otu.tab[[2]])]
IDname='subjid'
Tname='time'

#### NOT RUN
# ARZIMMresult=ARZIMM::ARZIMM(phy=NULL,Vname = Vname,Conname = Conname,fdata=fdata,
# IDname = IDname,Tname = Tname)
```

Index

- *Topic **ARZIMM**
 - ARZIMM, [3](#)
- *Topic **EM**
 - emnr, [6](#)
- *Topic **Simulation**
 - simMixTime, [9](#)
- *Topic **algorithm**
 - emnr, [6](#)
- *Topic **controls**
 - ARZIMM.control, [5](#)
- *Topic **datasets**
 - parameters, [7](#)
 - phyExample, [8](#)
 - sampladata, [9](#)
- *Topic **microbial dynamic systems;
interaction; stability;
longitudinal microbiome
data**
 - ARZIMM-package, [2](#)
- *Topic **model**
 - ARZIMM.control, [5](#)

ARZIMM, [3](#)
ARZIMM-package, [2](#)
ARZIMM.control, [5](#)

emnr, [6](#)

parameters, [7](#)
phyExample, [8](#)

sampladata, [9](#)
simMixTime, [9](#)