Package ‘quickNmix’

October 13, 2022

Title Asymptotic N-Mixture Model Fitting

Version 1.1.1

Date 2022-02-28

Maintainer Matthew RP Parker <mrparker909@gmail.com>

Description For fitting N-mixture models using either FFT or asymptotic approaches. FFT N-mixture models extend the work of Cowen et al. (2017) <doi:10.1111/biom.12701>. Asymptotic N-mixture models extend the work of Dail and Madsen (2011) <doi:10.1111/j.1541-0420.2010.01465.x>, to consider asymptotic solutions to the open population N-mixture models. The FFT models are derived and described in ``Parker, M.R.P., Elliott, L., Cowen, L.L.E. (2022). Computational efficiency and precision for replicated-count and batch-marked hidden population models [Manuscript in preparation]. Department of Statistics and Actuarial Sciences, Simon Fraser University.''. The asymptotic models are derived and described in: ``Parker, M.R.P., Elliott, L., Cowen, L.L.E., Cao, J. (2022). Fast asymptotic solutions for N-mixtures on large populations [Manuscript in preparation]. Department of Statistics and Actuarial Sciences, Simon Fraser University.''.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown

Depends R (>= 2.10)

Imports foreach, parallel, optimParallel, doParallel,

NeedsCompilation no

Author Matthew RP Parker [aut, cre] (<https://orcid.org/0000-0003-3021-7959>),
Lloyd Elliott [aut] (<https://orcid.org/0000-0003-2187-7314>),
Laura LE Cowen [aut] (<https://orcid.org/0000-0002-0853-1450>),
Jiguo Cao [aut] (<https://orcid.org/0000-0001-7417-6330>)

Repository CRAN

Date/Publication 2022-03-21 08:00:05 UTC

R topics documented:

`anmu` *Ancient Murrelet Chick Counts*

Description

Parker et al. (2020) include a data set collected by the Laskeek Bay Conservation Society on yearly Ancient Murrelet chick counts from the year 1990 to 2006. The data is collected for six sampling sites on East Limestone Island.

Usage

```
anmu
```

Format

A matrix with 6 rows and 17 columns. Each row represents a sampling location, and each column represents a sampling occasion:

Source

Parker et al. (2020) doi: 10.1007/s10651020004553

References

Parker, M.R.P., Pattison, V. & Cowen, L.L.E. Estimating population abundance using counts from an auxiliary population. Environ Ecol Stat 27, 509–526 (2020). doi: 10.1007/s10651020004553

`Ax_log` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
Ax_log(logA, logx)
```

Arguments

`logA`	PARAM_DESCRIPTION
`logx`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`eagles` *Golden Eagle Counts Data*

Description

Golden Eagle counts for the years 1993 to 2020, collected by the Rocky Mountain Eagle Research Foundation (RMERF). Counts are made during the spring from April 1st until March 22nd each year. Data is available from Eaglewatch.ca.

Usage

```
eagles
```

Format

A data frame with 28 rows and 11 columns. Each row represents a spring observation period, and each column represents a variable:

Year	observation year
Hours	hours spent collecting observations
Eagles	number of Golden Eagles observed
TotPrec	total precipitation measured in mm
UniqueObservers	number of principle observers who performed observations
PeterSherrington_Year	indicator variable which is 1 for years in which Peter Sherrington was the most prevalent principle observer
observerPC1	top 5 principal component scores calculated from the top twelve most prevalent principal observers
observerPC2	top 5 principal component scores calculated from the top twelve most prevalent principal observers
observerPC3	top 5 principal component scores calculated from the top twelve most prevalent principal observers
observerPC4	top 5 principal component scores calculated from the top twelve most prevalent principal observers
observerPC5	top 5 principal component scores calculated from the top twelve most prevalent principal observers

Source

RMERF (2020) EagleWatch.ca

References

Rocky Mountain Eagle Research Foundation (RMERF). EagleWatch.ca. (2020).

`fitNmix` *Fit Asymptotic N-mixture Model*

Description

Fit an open population N-mixture model using the asymptotic approximation. The four parameters are mean initial site abundance lambda, mean recruitments gamma, survival probability omega, and probability of detection pdet. Parameters can be made to vary over sites and over times by including parameter covariates. Note that this function is essentially a wrapper for optim acting on the nll function.

Usage

```
fitNmix(
  nit,
  K = NULL,
  starts = NULL,
  l_s_c = NULL,
  g_s_c = NULL,
  g_t_c = NULL,
  o_s_c = NULL,
  o_t_c = NULL,
  p_s_c = NULL,
  p_t_c = NULL,
  SMALL_a_CORRECTION = FALSE,
  VERBOSE = FALSE,
  outfile = NULL,
  method = "BFGS",
  ...
)
```

Arguments

`nit`	Matrix of counts data. Rows represent sites, columns represent sampling occasions. Note that if the data is a vector, then it will be converted to a matrix with a single row.
`K`	Upper bound on summations in the likelihood function. K should be chosen large enough that the negative log likelihood function is stable (unchanging as K increases). If K=NULL, K=5*max(nit) will be used as default. Default: NULL
`starts`	Either NULL for default starting values, or a vector of parameter values: `c(log(lambda), log(gamma), logit(omega), logit(pdet))`. Note that the parameter vector will need to be longer by one for each parameter coefficient if covariate values are supplied. The order of coefficients is: `c(lambda, l_s_c, gamma, g_s_c, g_t_c, omega, o_s_c, o_t_c, pdet, p_s_c, p_t_c)`
`l_s_c`	List of lambda site covariates, Default: NULL
`g_s_c`	List of gamma site covariates, Default: NULL
`g_t_c`	List of gamma time covariates, Default: NULL
`o_s_c`	List of omega site covariates, Default: NULL
`o_t_c`	List of omega time covariates, Default: NULL
`p_s_c`	List of pdet site covariates, Default: NULL
`p_t_c`	List of pdet time covariates, Default: NULL
`SMALL_a_CORRECTION`	If TRUE will apply the small a correction when calculating the transition probability matrix, Default: FALSE
`VERBOSE`	If TRUE, will print additional information during model fitting, Default: FALSE
`outfile`	Location of csv file to write/append parameter values, can be used to checkpoint long running model fits. Default: NULL

`method`	Optimization method, passed to optim function, options include: "BFGS", "Nelder-Mead", "CG". Default: "BFGS"
`...`	Additional arguments passed to the optimization function optim. For example: `control = list(trace=1, REPORT=1, reltol=1e-10)`

Value

Returns the fitted model object.

Examples

```
if (interactive()) {
nit = matrix(c(1,1,0,1,1), nrow=1) # observations for 1 site, 5 sampling occassions
model1 = fitNmix(nit, K=2)        # fit the model with population upper bound K=2
}
```

`fitNmixPara` *Fit Asymptotic N-mixture Model Using optimParallel*

Description

Fit an open population N-mixture model using the asymptotic approximation. The four parameters are mean initial site abundance lambda, mean recruitments gamma, survival probability omega, and probability of detection pdet. Parameters can be made to vary over sites and over times by including parameter covariates. Note that this function is essentially a wrapper for optim acting on the nll function.

Usage

```
fitNmixPara(
  cluster,
  nit,
  K = NULL,
  starts = NULL,
  l_s_c = NULL,
  g_s_c = NULL,
  g_t_c = NULL,
  o_s_c = NULL,
  o_t_c = NULL,
  p_s_c = NULL,
  p_t_c = NULL,
  SMALL_a_CORRECTION = FALSE,
  VERBOSE = FALSE,
  outfile = NULL,
  LowerBounds = NULL,
  ...
)
```

Arguments

`cluster`	cluster object created using makeCluster, for example: `cl <- makeCluster(parallel::detectCores()`
`nit`	Matrix of counts data. Rows represent sites, columns represent sampling occasions. Note that if the data is a vector, then it will be converted to a matrix with a single row.
`K`	Upper bound on summations in the likelihood function. K should be chosen large enough that the negative log likelihood function is stable (unchanging as K increases). If K=NULL, K=5*max(nit) will be used as default. Default: NULL
`starts`	Either NULL for default starting values, or a vector of parameter values: `c(log(lambda), log(gamma), logit(omega), logit(pdet))`. Note that the parameter vector will need to be longer by one for each parameter coefficient if covariate values are supplied. The order of coefficients is: `c(lambda, l_s_c, gamma, g_s_c, g_t_c, omega, o_s_c, o_t_c, pdet, p_s_c, p_t_c)`
`l_s_c`	List of lambda site covariates, Default: NULL
`g_s_c`	List of gamma site covariates, Default: NULL
`g_t_c`	List of gamma time covariates, Default: NULL
`o_s_c`	List of omega site covariates, Default: NULL
`o_t_c`	List of omega time covariates, Default: NULL
`p_s_c`	List of pdet site covariates, Default: NULL
`p_t_c`	List of pdet time covariates, Default: NULL
`SMALL_a_CORRECTION`	If TRUE will apply the small a correction when calculating the transition probability matrix, Default: FALSE
`VERBOSE`	If TRUE, will print additional information during model fitting, Default: FALSE
`outfile`	Location of csv file to write/append parameter values, can be used to checkpoint long running model fits. Default: NULL
`LowerBounds`	Lower bounds to be passed to optimParallel (if NULL, default values will be used), you may need to set this manually if you receive errors such as: "L-BFGS-B needs finite values of 'fn'".
`...`	Additional arguments passed to the optimization function optimParallel.

Value

Returns the fitted model object.

Examples

```
if (interactive()) {
cl <- makeCluster(parallel::detectCores()-1) # number of clusters should be 2*p+1 for optimal gains
nit = matrix(c(1,1,0,1,1,2,2), nrow=1) # observations for 1 site, 7 sampling occassions
model1 = fitNmixPara(cl, nit, K=100) # fit the model with population upper bound K=100
parallel::stopCluster(cl)
}
```

`logSubtractExp` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
logSubtractExp(x)
```

Arguments

`x`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`logSumExp` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
logSumExp(x)
```

Arguments

`x`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`log_tp_MAT_lse` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
log_tp_MAT_lse(M, omeg, gamm, corrections)
```

Arguments

`M`	PARAM_DESCRIPTION
`omeg`	PARAM_DESCRIPTION
`gamm`	PARAM_DESCRIPTION
`corrections`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

See Also

`foreach`

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`nll` *Negative Log Likelihood*

Description

Computes the negative log likelihood function for the open population asymptotic N-mixtures model.

Usage

```
nll(
  par,
  nit,
  K,
  l_s_c = NULL,
  g_s_c = NULL,
  g_t_c = NULL,
  o_s_c = NULL,
  o_t_c = NULL,
  p_s_c = NULL,
  p_t_c = NULL,
  SMALL_a_CORRECTION = FALSE,
  VERBOSE = FALSE,
  outfile = NULL
)
```

Arguments

`par`	Vector of parameter values: c(log(lambda), log(gamma), logit(omega), logit(pdet)). Note that the parameter vector will need to be longer if covariate values are supplied.
`nit`	Matrix of counts data. Rows represent sites, columns represent sampling occasions. Note that if the data is a vector, then it will be converted to a matrix with a single row.
`K`	Upper bound on summations in the likelihood function. K should be chosen large enough that the negative log likelihood function is stable (unchanging as K increases).
`l_s_c`	List of lambda site covariates, Default: NULL

`g_s_c`	List of gamma site covariates, Default: NULL
`g_t_c`	List of gamma time covariates, Default: NULL
`o_s_c`	List of omega site covariates, Default: NULL
`o_t_c`	List of omega time covariates, Default: NULL
`p_s_c`	List of pdet site covariates, Default: NULL
`p_t_c`	List of pdet time covariates, Default: NULL
`SMALL_a_CORRECTION`	If TRUE will apply the small a correction when calculating the transition probability matrix, Default: FALSE
`VERBOSE`	If TRUE, will print additional information, Default: FALSE
`outfile`	Location of csv file to write/append parameter values, Default: NULL

Details

DETAILS

Value

Returns the negative log likelihood function evaluated at par.

Examples

```
if (interactive()) {
nit = matrix(c(1,1,0,1,1), nrow=1) # observations for 1 site, 5 sampling occassions
par = c(1,1,1,0) # parameter values at which to calculate the negative log likelihood (nll)
nll(par, nit, K=10) # nll with K=10
nll(par, nit, K=10, SMALL_a_CORRECTION=TRUE) # nll with small a correction
}
```

`Pab` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
Pab(a, b, omega, gamma)
```

Arguments

`a`	PARAM_DESCRIPTION
`b`	PARAM_DESCRIPTION
`omega`	PARAM_DESCRIPTION
`gamma`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`Pab_asymptotic` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
Pab_asymptotic(a, b, omega, gamma, corrections = FALSE)
```

Arguments

`a`	PARAM_DESCRIPTION
`b`	PARAM_DESCRIPTION
`omega`	PARAM_DESCRIPTION
`gamma`	PARAM_DESCRIPTION
`corrections`	PARAM_DESCRIPTION, Default: FALSE

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`Pab_gamma` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
Pab_gamma(a, b, gamma)
```

Arguments

`a`	PARAM_DESCRIPTION
`b`	PARAM_DESCRIPTION
`gamma`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`Pab_omega` *FUNCTION_TITLE*

Description

FUNCTION_DESCRIPTION

Usage

```
Pab_omega(a, b, omega)
```

Arguments

`a`	PARAM_DESCRIPTION
`b`	PARAM_DESCRIPTION
`omega`	PARAM_DESCRIPTION

Details

DETAILS

Value

OUTPUT_DESCRIPTION

Examples

```
## Not run:
if(interactive()){
 #EXAMPLE1
 }

## End(Not run)
```

`pCountOpenFFT` *Fit FFT N-mixture Model*

Description

Fit an open population N-mixture model using the FFT method of computing the Transition Probability matrix. The four parameters are mean initial site abundance lambda, mean recruitments gamma, survival probability omega, and probability of detection pdet. Parameters can be made to vary over sites and over times by including parameter covariates. Note that this function is essentially a wrapper for optim acting on the nll_FFT function.

Usage

```
pCountOpenFFT(
  nit,
  K = NULL,
  starts = NULL,
  l_s_c = NULL,
  g_s_c = NULL,
  g_t_c = NULL,
  o_s_c = NULL,
  o_t_c = NULL,
  p_s_c = NULL,
  p_t_c = NULL,
  VERBOSE = FALSE,
  outfile = NULL,
  method = "BFGS",
  ...
)
```

Arguments

`nit`	Matrix of counts data. Rows represent sites, columns represent sampling occasions. Note that if the data is a vector, then it will be converted to a matrix with a single row.
`K`	Upper bound on summations in the likelihood function. K should be chosen large enough that the negative log likelihood function is stable (unchanging as K increases). If K=NULL, K=5*max(nit) will be used as default. Default: NULL
`starts`	Either NULL for default starting values, or a vector of parameter values: `c(log(lambda), log(gamma), logit(omega), logit(pdet))`. Note that the parameter vector will need to be longer by one for each parameter coefficient if covariate values are supplied. The order of coefficients is: `c(lambda, l_s_c, gamma, g_s_c, g_t_c, omega, o_s_c, o_t_c, pdet, p_s_c, p_t_c)`
`l_s_c`	List of lambda site covariates, Default: NULL
`g_s_c`	List of gamma site covariates, Default: NULL
`g_t_c`	List of gamma time covariates, Default: NULL
`o_s_c`	List of omega site covariates, Default: NULL
`o_t_c`	List of omega time covariates, Default: NULL
`p_s_c`	List of pdet site covariates, Default: NULL
`p_t_c`	List of pdet time covariates, Default: NULL
`VERBOSE`	If TRUE, will print additional information during model fitting, Default: FALSE
`outfile`	Location of csv file to write/append parameter values, can be used to checkpoint long running model fits. Default: NULL (no csv file created).
`method`	Optimization method, passed to optim function, options include: "BFGS", "Nelder-Mead", "CG". Default: "BFGS"
`...`	Additional arguments passed to the optimization function optim. For example: `control = list(trace=1, REPORT=1, reltol=1e-10)`

Value

Returns the fitted model object.

Examples

```
if (interactive()) {
# No Covariates
nit = matrix(c(1,1,0,2,3), nrow=1) # observations for 1 site, 5 sampling occassions
model1 = pCountOpenFFT(nit, K=10)  # fit the model with population upper bound K=10

# Site Covariates
o_s_c = list(cov1=c(0,0,1)) # omega site covariates, cov1 is categorical
nit = matrix(c(1,1,0,2,3,
               1,0,1,3,2,
               4,1,3,2,0), nrow=3, byrow=T) # 3 sites, 5 sampling occassions
model2 = pCountOpenFFT(nit, K=20, o_s_c=o_s_c) # fit the model with population upper bound K=20

# Time Covariates
g_t_c = list(temp=c(0.5,0.3,0.6,0.7,NA)) # transition covariates: only first T-1=4 values used
model3 = pCountOpenFFT(nit, K=10, g_t_c=g_t_c)  # fit the model with population upper bound K=10
}
```

Index