

# Package ‘quickNmix’

June 23, 2021

**Title** Asymptotic N-Mixture Model Fitting

**Version** 1.0.3

**Date** 2021-06-20

**Maintainer** Matthew RP Parker <mrparker909@gmail.com>

**Description** For fitting asymptotic N-mixture models. Asymptotic N-mixture models extend the work of Dail and Madsen (2011) <[doi:10.1111/j.1541-0420.2010.01465.x](https://doi.org/10.1111/j.1541-0420.2010.01465.x)>, to consider asymptotic solutions to the open population N-mixture models. The asymptotic models are derived and described in: ``Parker, M.R.P., Elliott, L., Cowen, L.L.E., Cao, J. (2021). Asymptotic solutions for N-mixture models with count data [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, 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** 2021-06-23 06:30:02 UTC

## R topics documented:

anmu	2
Ax_log	3
eagles	3

fitNmix . . . . .	4
logSubtractExp . . . . .	6
logSumExp . . . . .	7
log_tp_MAT_lse . . . . .	7
nll . . . . .	8
Pab . . . . .	10
Pab_asymptotic . . . . .	10
Pab_gamma . . . . .	11
Pab_omega . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

anmu	<i>Ancient Murrelet Chick Counts</i>
------	--------------------------------------

---

## 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 data frame 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](https://doi.org/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](https://doi.org/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  $p_{det}$ . 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**

<code>nit</code>	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.
<code>K</code>	Upper bound on summations in the likelihood function. <code>K</code> should be chosen large enough that the negative log likelihood function is stable (unchanging as <code>K</code> increases). If <code>K=NULL</code> , <code>K=5*max(nit)</code> will be used as default. Default: <code>NULL</code>
<code>starts</code>	Either <code>NULL</code> for default starting values, or a vector of parameter values: <code>c(log(lambda), log(gamma), 1, ...)</code> . 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: <code>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)</code>
<code>l_s_c</code>	List of lambda site covariates, Default: <code>NULL</code>
<code>g_s_c</code>	List of gamma site covariates, Default: <code>NULL</code>
<code>g_t_c</code>	List of gamma time covariates, Default: <code>NULL</code>
<code>o_s_c</code>	List of omega site covariates, Default: <code>NULL</code>
<code>o_t_c</code>	List of omega time covariates, Default: <code>NULL</code>
<code>p_s_c</code>	List of pdet site covariates, Default: <code>NULL</code>
<code>p_t_c</code>	List of pdet time covariates, Default: <code>NULL</code>
<code>SMALL_a_CORRECTION</code>	If <code>TRUE</code> will apply the small a correction when calculating the transition probability matrix, Default: <code>FALSE</code>
<code>VERBOSE</code>	If <code>TRUE</code> , will print additional information during model fitting, Default: <code>FALSE</code>
<code>outfile</code>	Location of csv file to write/append parameter values, can be used to checkpoint long running model fits. Default: <code>NULL</code>

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**

```
nit = matrix(c(1,1,0,1,1), nrow=1) # observations for 1 site, 5 sampling occasions
model1 = fitNmix(nit, K=10)         # fit the model with population upper bound K=10
```

---

logSubtractExp	<i>FUNCTION_TITLE</i>
----------------	-----------------------

---

**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	<i>FUNCTION_TITLE</i>
-----------	-----------------------

---

**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	<i>FUNCTION_TITLE</i>
----------------	-----------------------

---

**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: $c(\log(\lambda), \log(\gamma), \text{logit}(\omega), \text{logit}(\text{pdet}))$ . 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

```

nit = matrix(c(1,1,0,1,1), nrow=1) # observations for 1 site, 5 sampling occasions
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)
```

# Index

## \* datasets

anmu, 2  
eagles, 3

anmu, 2  
Ax\_log, 3

eagles, 3

fitNmix, 4  
foreach, 8

log\_tp\_MAT\_lse, 7  
logSubtractExp, 6  
logSumExp, 7

nll, 8

Pab, 10  
Pab\_asymptotic, 10  
Pab\_gamma, 11  
Pab\_omega, 12