Package ‘NB’

February 19, 2015

Type Package

Title Maximum Likelihood method in estimating effective population size from genetic data

Version 0.9

Date 2014-10-03

Author Tin-Yu Hui @ Imperial College London

Maintainer Tin-Yu Hui <`tin-yu.hui11@imperial.ac.uk`>

Description Estimate the effective population size of a closed population using genetic data collected from two or more data points.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-06 17:37:10

R topics documented:

`NB-package` *Estimating effective population size from genetic data*

Description

The concept of effective population size (N_e) is an inportant concept in population genetics. It tells the "effective" number of individuals within a population, that is, those who are able to reproduce. The allele frequencies in a closed population change over time as known as genetic drift, and the magnitude of this fluctuation is directly related to N_e. This package aims to estimate this quantity from genetic samples collected over multiple time points.

Details

Package:	NB
Type:	Package
Version:	0.9
Date:	2014-10-03
License:	GPL (>= 2)

Author(s)

Tin-Yu Hui

References

Please email me for referencing or citation.

`NB.estimator` *Maximum likelihood estimation of effective population size from temporally-spaced genetic data*

Description

The concept of effective population size is imporatnt in population genetics. This function estimates the effective population size N_e from temporally-sapced genetic data using maximum likelihood method with continuous approximation.

Usage

```
NB.estimator(infile, alleles, sample.interval, bound = c(50, 1e+07),
profile.likelihood=FALSE)
```

Arguments

`infile`	The input dataset. It should be a plain text file (*.txt) containing the number of alleles of each allele, at each locus, from each temporal sample. For example, Suppose we have i temporal samples from the focal population, j loci, and K_j alleles at locus j. Denoting the number of copies of allele k at locus j from sample i as $n_{k,j,i}$, then the input format of data is as follows: $n_{1,1,1}$~$n_{2,1,1}$~...~$n_{K_1,1,1}$ $n_{1,2,1}$~$n_{2,2,1}$~...~$n_{K_2,2,1}$ ... $n_{1,j,1}$~$n_{2,j,1}$~...~$n_{K_j,j,1}$ $n_{1,1,2}$~$n_{2,1,2}$~...~$n_{K_1,1,2}$ $n_{1,2,2}$~$n_{2,2,2}$~...~$n_{K_2,2,2}$ ... $n_{1,j,2}$~$n_{2,j,2}$~...~$n_{K_j,j,2}$ ... Note: A space is required to separate the allele counts.
`alleles`	A vector containing the number of alleles at each locus. For example, `c(4, 4, 4)` would mean that 3 loci are sampled, with 4 alleles each.
`sample.interval`	A vector stating at which generations the samples were taken. For example, `c(0, 8)` would indicate that two samples were collected from the 0th and 8th generation.
`bound`	Lower and upper bound for searching for the effective population size. Default values are `c(50, 1e7)`.
`profile.likelihood`	`TRUE` if you would like to `plot` the log-likelihood function, or to look at the details of the log-likelihood values between your confidence interval. `FALSE` will return you the lower and upper 95% confidence interval only.

Details

The input arguments above largely follow the input format of MLNE (Wang, 2001; Wang and Whitlock, 2003) to allow users to switch between platforms with the minimal effort. The infile should be in a plain text (*.txt) file format. It contains the same information as the input data as MLNE does. The built-in `optim` optimiser is wrapped within this function.

Value

`N`	The point estimate of the effective population size N_e.
`CI`	The approximate 95% confidence interval calculated from the log-likelihood. They are the region where the log-likelihood is 2 units below the maxima.
`log.like`	The value of the maximised log-likelihood.
`profile.CI`	A list of log-likelihood values as a function of N_e within the 95% confidence interval.

Note

If you would like to use your own optimisation algorithm or customerise the parameters within the `optim` function, please consider using `NB.likelihood` from this package.

See Also

`NB.likelihood`. `NB.example.dataset`.

Examples

```
## CREATE SAMPLE DATASET
NB.example.dataset()

## RUN THE FUNCTION
NB.estimator(infile='sample_data.txt', alleles=rep(4, 50),
sample.interval=c(0, 8), profile.likelihood=FALSE)

#####
## NUMERICAL RESULTS
#$N
# [1] 1241.079
#
#$CI
#[1]  594.195 6375.933
#
#$log.like
#[1] -543.9159
#####
```

`NB.example.dataset` *Generate sample dataset*

Description

To create a sample dataset in plain text format to illustrate the use of this package.

Usage

```
NB.example.dataset()
```

Details

No arguments required. Just run the function.

Note

A plain text (*txt) file will be created in your current working directory. Please make sure that you have the permission to write onto the working directory.

See Also

`NB.estimator`

Examples

```
## CREATE A SAMPLE DATASET IN PLAIN TEXT FORM
NB.example.dataset()

## CHECK YOUR WORKING DIRECTORY!
```

`NB.likelihood` *Log-likelihood values for the NB estimator*

Description

This will return return the log-likelihood value given a value of N.

Usage

```
NB.likelihood(N, infile, alleles, sample.interval)
```

Arguments

`N`	The effective population size dor diploid individuals.
`infile`	Your input data file in a plain text format. This data contains the allele counts at each locus and from each sampling time point.
`alleles`	a vector containing the number of alleles at each locus. For example, c(4, 4, 4) would mean that 3 loci are sampled, with 4 alleles each.
`sample.interval`	a vector stating at which generations the samples were taken. For example, c(0, 8) would indicate that two samples were collected from the 0th and 8th generation.

Details

More details please see `NB.estimator`.

Value

This function returns one single element, the log-likelihood of the model given the effective population size N and your inputs.

Note

This function allows you to use your own optimisation algorithms or customise the parameters using `optim` or `nlm`. Otherwise please use `NB.estimator` which has the internal `optim` wrapped inside.

See Also

`NB.estimator`.

Examples

```
## CREATE SAMPLE DATASET
NB.example.dataset()

##SEE WHAT'S THE LOG-LIKELIHOOD VALUE IS WHEN N=1000
NB.likelihood(N=1000, infile='sample_data.txt',
alleles=rep(4, 50), sample.interval=c(0, 8))

#####
# NUMERICAL RESULT
#[1] -544.0405
#####
```

`NB.plot.likelihood` *Plot profile (log)-likelihood*

Description

This function does not maximise the likelihood function but rather provide a range of log-likelihood values as a function of effective population size.

Usage

```
NB.plot.likelihood(infile, alleles, sample.interval, lb, ub, step, plotit = TRUE)
```

Arguments

`infile`	The input data. Please refer to `infile` from `NB.estimator`.
`alleles`	Number of alleles at each locus. Please refer to `alleles` from `NB.estimator`.
`sample.interval`	Please refer to `sample.interval` from `NB.estimator`.
`lb`	The lower bound of N.
`ub`	The upper bound of N.
`step`	How many points do you want to evaluate at.
`plotit`	Do you want to plot the log-likelihood out?

Details

The input arguments `infile`, `alleles`, and `sample.interval` have the same definitions as those in `NB.estimator`.

Value

Returns a range of log-likelihood values with the associated effective population size. If `plotit==TRUE` then a plot of the log-likelihood will also be produced.

Note

This would be a good way to examine whether the maximization converges to the global maximum.

See Also

`NB.estimator`.

Examples

```
## CREATE SAMPLE DATASET
NB.example.dataset()

## PLOT THE LOG-LIKELIHOOD
NB.plot.likelihood(infile='sample_data.txt', alleles=rep(4, 50),
sample.interval=c(0, 8), lb=300, ub=2000, step=200)
```

Index