Package 'capwire'

July 2, 2014

Type Package

Title Estimates population size from non-invasive sampling

Version 1.1.4

Date 2012-08-16

Author Matthew W. Pennell and Craig R. Miller

Maintainer Matthew W. Pennell <mwpennell@gmail.com>

Description Fits models from Miller et al. 2005 to estimate population sizes from natural populations. Several models are implemented. Package also includes functions to perform a likelihood ratio test to choose between models, perform parametric bootstrapping to obtain confidence intervals and multiple functions to simulate data.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-08-16 13:47:50

NeedsCompilation no

R topics documented:

`capwire-package` *Estimate Population Size from Non-Invasive Sampling*

Description

capwire uses capture-recapture models which allow for sampling individuals multiple times per session to estimate population size from non-invasive samples. These models were originally implemented in the Java program Capwire (Miller et al. 2005)

The models all assume that individuals can be identified with certainty (usually with, but not limited to) genetic data.

The package also includes functions to compare models, get confidence intervals using a parametric bootstrap and simulate data under a variety of conditions.

Details

Package:	capwire
Type:	Package
Version:	1.1.4
Date:	2012-08-16
License:	License: GPL (>= 2)

Author(s)

Matthew W. Pennell and Craig R. Miller

Maintainer: Matthew W. Pennell <mwpennell@gmail.com>

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

---

`bootstrapCapwire` *Parametric Bootstrap ECM and TIRM Models*

Description

Uses maximum likelihood parameter estimates from `fitEcm`, `fitTirm`, or `fitTirmPartition` to perform a parametric bootstrap to get confidence intervals for the estimate of the population size

Usage

```
bootstrapCapwire(x, bootstraps = 1000, CI = c(0.025, 0.975))
```

Arguments

`x`	An object inherited from `fitEcm`, `fitTirm`, or `fitTirmPartition`
`bootstraps`	The number of bootstraps to be performed (default is 1000)
`CI`	A vector of quantiles to to generate a confidence interval for the population estimate. The default is `c(0.025, 0.975)`, denoting a symmetrical 95 percent confidence interval.

Details

This function uses the ML estimates obtained from fitting the model to simulate data under the model.

`bootstrapCapwire` inherits an object from `fitEcm`, `fitTirm`, or `fitTirmPartition` such that the model and parameter estimates do not need to be specified.

The ML estimate for the population size will also be returned but this will not be changed by `bootstrapCapwire`

Note that if the model is a poor fit to the data, the confidence intervals may not be reliable.

The lower confidence interval is bounded by the number of unique individuals in the sample.

Value

`ml.pop.size`	The maximum likelihood estimate for the population size obtained by fitting the model
`conf.int`	The confidence interval for the estimate of the population size

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

See Also

fitEcm, fitTirm, fitTirmPartition

Examples

```
## Simulate data under Equal Capture Model

data <- simEcm(n=40, s=150)

## Fit Equal Capture Model to Data

res <- fitEcm(data=data, max.pop=200)

## Perform bootstrap to get confidence intervals

ci <- bootstrapCapwire(x=res, bootstraps=50, CI = c(0.025, 0.975))
```

---

buildClassTable *Convert capture counts to data frame with capture classes*

Description

Converts a vector of capture counts into a two-column data frame consisting of all capture classes and the individuals associated with each class.

Usage

```
buildClassTable(counts)
```

Arguments

counts	A vector of capture count data

Value

A two-column data frame with the first column specifiying the capture class (where all individuals in class i were caught i times) and the second column specifying the number of individuals in this capture class.

The data frame can be used as the data argument for any of the model-fitting functions implemented in capwire

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

buildIndTable, indToClass, classToInd

Examples

```
## create a vector of capture counts

counts <- c(1,1,1,1,1,2,2,3,3,4,5)

## build table

data <- buildClassTable(counts)
data
```

---

buildIndTable *Convert capture counts to data frame with individuals*

Description

Converts a vector of capture counts into a two-column data frame consisting of all individuals and the capture counts associated with each

Usage

```
buildIndTable(counts)
```

Arguments

counts	A vector of capture count data

Value

A two-column data frame with the first column specifiying the individual IDs and the second column specifying the number of times each individual was captured. Note here that the individual IDs are arbitrarily defined.

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

buildClassTable, indToClass, classToInd

Examples

```
## create a vector of capture counts

mycounts <- c(1,1,1,1,1,2,2,3,3,4,5)

## build table

data <- buildIndTable(mycounts)
data
```

---

classToInd *Convert a table of capture class data into a table of individuals*

Description

Converts a data frame where individuals are grouped by capture class into a data frame consisting of individuals and their associated capture counts.

Usage

```
classToInd(x)
```

Arguments

x	A two-column data frame with the first column specifiying the capture class (where all individuals in class i were caught i times) and the second column specifying the number of individuals in this capture class.

Details

This is primarily used as an internal function through capwire

Value

A two-column data frame with the first column specifiying the individual IDs and the second column specifying the number of times each individual was captured. Note here that the individual IDs are arbitrarily defined.

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

buildIndTable, buildClassTable, indToClass

Examples

```
## create a vector of capture counts

counts <- c(1,1,1,2,2,2,3,3,4)

## make into a table of classes

class.data <- buildClassTable(counts)

## convert to table of individuals

data <- classToInd(class.data)

data
```

---

drawCapRatesBeta *Draw Capture Rates from Beta Distribution*

Description

This function is just a wrapper for rbeta which can be used in conjuction with simCapture to simulate capture data where the rates are drawn from a beta distribution

Usage

```
drawCapRatesBeta(shape1, shape2)
```

Arguments

shape1	The first shape parameter of the beta distribution
shape2	The second shape parameter of the beta distribution

Details

This function returns a function which can be used as an argument for simCapture

Value

A function which takes the number of capture rates to draw as an argument

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

simCapture, drawCapRatesUnif, drawCapRatesGamma, drawCapRatesExp, drawCapRatesGeom

Examples

```
## Specify the distribution

dist <- drawCapRatesBeta(shape1=1, shape2=0.5)

## Simulate a data set with the capture rates drawn from dist

data <- simCapture(n=30, s=100, dist.func=dist)

data
```

---

drawCapRatesExp *Draw Capture Rates from Exponential Distribution*

Description

This function is just a wrapper for rexp which can be used in conjuction with simCapture to simulate capture data where the rates are drawn from an exponential distribution

Usage

```
drawCapRatesExp(r)
```

Arguments

r	Rate parameter of an exponenital distirbution

Details

This function returns a function which can be used as an argument for simCapture

Value

A function which takes the number of capture rates to draw as an argument

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

simCapture, drawCapRatesUnif, drawCapRatesGamma, drawCapRatesBeta, drawCapRatesGeom

Examples

```
## Specify the distribution

dist <- drawCapRatesExp(r=0.5)

## Simulate a data set with the capture rates drawn from dist

data <- simCapture(n=30, s=100, dist.func=dist)

data
```

---

drawCapRatesGamma *Draw Capture Rates from Gamma Distribution*

Description

This function is just a wrapper for rgamma which can be used in conjuction with simCapture to simulate capture data where the rates are drawn from a gamma distribution

Usage

```
drawCapRatesGamma(shape, rate)
```

Arguments

shape	Shape parameter of gamma distribution
rate	Rate parameter of gamma distribution

Details

This function returns a function which can be used as an argument for simCapture

Value

A function which takes the number of capture rates to draw as an argument

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

simCapture, drawCapRatesUnif, drawCapRatesExp, drawCapRatesBeta, drawCapRatesGeom

Examples

```
## Specify the distribution

dist <- drawCapRatesGamma(shape=0.5, rate=0.5)

## Simulate a data set with the capture rates drawn from dist

data <- simCapture(n=30, s=100, dist.func=dist)

data
```

---

drawCapRatesGeom *Draw Capture Rates from Geometric Distribution*

Description

This function is just a wrapper for rgeom which can be used in conjuction with simCapture to simulate capture data where the rates are drawn from a geometric distribution

Usage

```
drawCapRatesGeom(p)
```

Arguments

p	Parameter of geometric distribution

Details

This function returns a function which can be used as an argument for simCapture

Value

A function which takes the number of capture rates to draw as an argument

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

simCapture, drawCapRatesUnif, drawCapRatesExp, drawCapRatesBeta, drawCapRatesGamma

Examples

```
## Specify the distribution

dist <- drawCapRatesGeom(p=0.5)

## Simulate a data set with the capture rates drawn from dist

data <- simCapture(n=30, s=100, dist.func=dist)

data
```

---

drawCapRatesUnif *Draw Capture Rates from Uniform Distribution*

Description

This function is just a wrapper for `runif` which can be used in conjuction with `simCapture` to simulate capture data where the rates are drawn from a uniform distribution

Usage

```
drawCapRatesUnif(lower, upper)
```

Arguments

`lower`	Lower bound of the unifrom distribution
`upper`	Upper bound of the uniform distribution

Details

This function returns a function which can be used as an argument for `simCapture`

Value

function	A function which takes the number of capture rates to draw as an argument

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

simCapture, drawCapRatesGeom, drawCapRatesExp, drawCapRatesBeta, drawCapRatesGamma

Examples

```
## Specify the distribution

dist <- drawCapRatesUnif(lower=0.01, upper=1)

## Simulate a data set with the capture rates drawn from dist

data <- simCapture(n=30, s=100, dist.func=dist)

data
```

---

fitEcm *Fit Equal Capture Model (ECM)*

Description

Fits the Equal Capture Model (ECM) to count data to obtain the MLE for population size

Usage

```
fitEcm(data, max.pop)
```

Arguments

data	A two-column data frame with the first column specifiying the capture class (i.e. individuals in class i were caught i times) and the second column specifying the number of individuals in each class
max.pop	The maximum population size

Details

The ECM model fit by this function assumes that all individuals are equally likely to be "captured".

The value is specified for `max.pop` is not likely to matter as long as it is much greater than the maximum likelihood estimate for population size.

Note that if the data contains only singletons, the data is not informative and the maximum likelihood estimate for population size will be equal to `max.pop`

Value

`model`	The model specified
`likelihood`	The likelihood of the model
`ml.pop.size`	The maximum likelihood estimate for population size
`cap.ind`	The mean number of captures per individual
`sampled.ind`	The total number of individuals in the sample
`sample.size`	Total number of samples in the data set
`max.pop`	The maximum population size specified by `max.pop`

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

See Also

simEcm

Examples

```
## Simulate data under Equal Capture Model

data <- simEcm(n=40, s=150)

## Fit Equal Capture Model to Data

res <- fitEcm(data=data, max.pop=200)

res
```

`fitTirm` *Fit Two Innate Rates Model (TIRM)*

Description

Fits the Two Innate Rates Model (TIRM) to count data to obtain the MLE for population size

Usage

```
fitTirm(data, max.pop)
```

Arguments

`data`	A two-column data frame with the first column specifiying the capture class (i.e. individuals in class i were caught i times) and the second column specifying the number of individuals in each class
`max.pop`	The maximum population size

Details

The TIRM model fit by this function assumes that individuals can be assigned to two classes. Class A represent the frequently captured individuals. Class B represents the infrequently captured individuals.

The value is specified for `max.pop` is not likely to matter as long as it is much greater than the maximum likelihood estimate for population size.

Note that if the data contains only singletons, the data is not informative and the maximum likelihood estimate for population size will be equal to `max.pop`

Value

`model`	The model specified
`likelihood`	The likelihood of the model
`ml.pop.size`	The maximum likelihood estimate for population size
`ml.na`	The maximum likelihood estimate for the number of individuals in class A
`ml.nb`	The maximum likelihood estimate for the number of individuals in class B
`alpha`	The ratio of the rates of captures between class A and class B individuals
`cap.ind`	The mean number of captures per individual
`sampled.ind`	The total number of individuals in the sample
`sample.size`	Total number of samples in the data set
`max.pop`	The maximum population size specified by `max.pop`

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

See Also

simTirm

Examples

```
## Simulate data under Two Innate Rates Model

data <- simTirm(na=20, nb=15, alpha=4, s=150)

## Fit Two Innate Rates Model to Data

res <- fitTirm(data=data, max.pop=200)

res
```

---

fitTirmPartition *Fit the Two Innate Rates Model After Partiioning Data*

Description

Partitions count data into three groups based on PART algorithm of Stansbury et al. in prep. The lower two groups are retained for the population estimation procedure implemented in capwire. The upper group is excluded from the estimation procedure. This is done to handle overdispersed data which violate the assumptions of the Two-Innate Rates Model (TIRM).

The function also returns a p-value to evaluate whether partitioning the data is statistically warranted.

The function then fits the Two-Innate Rates Model (TIRM) to the lower set of the partitioned data and adds the individuals from the upper set to the estimate of population size

Usage

```
fitTirmPartition(data, max.pop, n.boots.part=100)
```

Arguments

data	A two-column data frame with the first column specifiying the capture class (i.e. individuals in class i were caught i times) and the second column specifying the number of individuals in each class
max.pop	The maximum population size
n.boots.part	The number of bootraps used to generate the null distribution for partitioning in order to estimate a p-value associated with the partitioning scheme used (see details). For most applications the default value (n.boots.part=100) should be appropriate.

Details

This function implements a new partitioning (PART) method that removes individuals detected a large number of times from the data. These individuals provide little information about the population size and, because they are inconsistent with the modeling assumption of just two capture rates, they cause estimation problems in capwire. Specifically, the PART algorithm attempts to remove very large capture counts and leave individuals captures between few and a moderate number of times.

The PART algorithm first fits the Two-Innate Rates Model (TIRM) to the entire data set. Assuming the distribution of capture counts can be approximated as either a two (null) or three (alternative) class multinomial distribution, the algorithm considers all possible ways to partition the count data under both the null and alternative distributions. The partitioning schemes which maximizes the respective likelihoods is retained. A test-statistic, the ratio of multinomial (log) likelihoods (3:2) is obtained for the observed data.

The MLE for number of individuals in class a, number of individuals in class b and the ratio of capture probabilities (alpha) are used to simulate `n.boots.part` data sets under the TIRM. The test-statistic is calculated for each data set as described above.

The observed test-statistic is then compared to the distribution of simulated test-statistics. When the observed test-statistic falls in the tail of the distribution of the test-statisitic from the simulated data, the null model can be rejected.

The TIRM model is then fit to the lower two classes obtained from partitioning the data in order to obtain the MLE for population size. Individuals from the upper class (i.e. those that were captured most often) to the population size estimate. It is therefore assumed that we have captured all individuals in the population with the highest probability of capture and that the class designation has been made without error.

The TIRM model fit by this function assumes that individuals can be assigned to two classes. Class A represent the frequently captured individuals. Class B represents the infrequently captured individuals.

The value is specified for `max.pop` is not likely to matter as long as it is much greater than the maximum likelihood estimate for population size.

Note that if the data contains only singletons, the data is not informative and the maximum likelihood estimate for population size will be equal to `max.pop`

Value

`model`	The model specified
`likelihood`	The likelihood of the model
`ml.pop.size`	The maximum likelihood estimate for population size including the addition of individuals excluded by partitioning method
`ml.na`	The maximum likelihood estimate for the number of individuals in class A
`ml.nb`	The maximum likelihood estimate for the number of individuals in class B
`alpha`	The ratio of the rates of captures between class A and class B individuals
`cap.ind`	The mean number of captures per individual
`sampled.ind`	Total number of individuals in the sample (prior to partitioning)
`sample.size`	Total number of samples in the data set (post partitioning)

`max.pop` The maximum population size specified by `max.pop`

`excluded` A two-column data frame specifying the capture classes and the number of individuals in this capture class for individuals excluded from the estimation procedure. These individuals are added to the population size estimate

`p.value.partition` The p-value obtained from the parametric bootstraps for partitioning the data into the three groups

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

Stansbury C.R., D.E. Ausband, P. Zager, C.M. Mack, C.R. Miller, M.W. Pennell, and L.P. Waits. in prep. Non-invasive genetic sampling of rendezvous sites and population estimation of grey wolves in Idaho, USA

See Also

`fitTirm`, `partitionCountData`

Examples

```
## Use dummy data set with a few individuals having very high capture counts
d <- c(1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,4,4,6,6,7,8,10,10,14,17,19,22,22,25)

data <- buildClassTable(d)

## Partition the data
## Fit the TIRM to the lower two classes obtained from partitioning data set

res <- fitTirmPartition(data=data, max.pop=200, n.boots.part=10)

res

## Compare population estimate from fitting TIRM
## to partitioned data set to population estimate obtained from fitting all data

res.all <- fitTirm(data=data, max.pop=200)$ml.pop.size

res.all
```

indToClass *Convert a table of individuals into a table of class data*

Description

Converts a data frame consisting of individuals and their associated capture counts into a data frame where individuals are grouped by capture class.

Usage

```
indToClass(x)
```

Arguments

x	A two-column data frame with the first column specifiying the individual IDs and the second column specifying the number of times each individual was captured.

Value

A two-column data frame with the first column specifiying the capture class (where all individuals in class i were caught i times) and the second column specifying the number of individuals in this capture class.

The data frame can be used as the data argument for any of the model-fitting functions implemented in capwire

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

buildIndTable, buildClassTable, classToInd

Examples

```
## create a vector of capture counts

counts <- c(1,1,1,2,2,2,3,3,4)

## make into a table of individuals

ind.data <- buildIndTable(counts)
```

```
## convert to table of classes

data <- classToInd(ind.data)

data
```

`lrtCapwire` *Likelihood Ratio Test Comparing TIRM to ECM*

Description

Performs Likelihood Ratio Test to compare model fits between the Equal Capture Model (ECM) and Two Innate-Rates Model (TIRM).

A parametric bootstrap is used to generate a null-distribution for the likelihood ratio test-statistic to assess significance

Usage

```
lrtCapwire(ecm, tirm, bootstraps = 100)
```

Arguments

`ecm`	An object inherited from `fitEcm`
`tirm`	An object inherited from `fitTirm`
`bootstraps`	The number of parameteric bootstraps to perform to generate null-distribution for likelihood ratio

Details

The maximum likelihood estimate for population size under the ECM model is used to simulate data under the ECM model. The data is then fit to both the ECM and TIRM models and the likelihood ratio test is evaluated. The observed Likelihood Ratio statistic is then compared to the distribution of likelihood ratio statistics observed when the null model is true in order to estimate a p-value.

Note that the likelihood ratio test should not be performed on an object inherited from `fitTirmPartition` as the partitioning algorithm assumes that there is heterogeneity in capture rates among individuals and that the rates can be divided into classes

In most applications of likelihood ratio tests, the more parameterized model will always have a higher likelihood. However, this is not necessarily the case for the models implemented in `capwire`. The likelihood involves a combinatorial term which denotes the number of ways a population could give rise to the observed vector of counts. As such, for some data sets (such as those for which the assumptions of the ECM model are valid), there will be more ways to obtain the observed data if the population is not subdivided into classes.

Value

LR	Likelihood Ratio
p.value	P value for the siginficance of the likelihood ratio test derived from a parametric bootstrap

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

Stansbury et al. in prep

See Also

fitEcm, fitTirm, simEcm

Examples

```
## Simulate data under Equal Capture Model

data <- simEcm(n=20, s=60)

## Fit Equal Capture Model (ECM) to count data

ecm <- fitEcm(data, max.pop=200)

## Fit Two-Innate Rates Model (TIRM) to count data

tirm <- fitTirm(data, max.pop=200)

## Perform Likelihood Ratio Test

lrtCapwire(ecm=ecm, tirm=tirm, bootstraps=10)
```

---

partitionCountData *Partition the Count Data*

Description

Partitions count data into three groups based on PART algorithm of Stansbury et al. in prep. The lower two groups are retained for the population estimation procedure implemented in capwire. The upper group is excluded from the estimation procedure. This is done to handle overdispersed data which violate the assumptions of the Two-Innate Rates Model (TIRM).

The function also returns a p-value to evaluate whether partitioning the data is statistically warranted.

Usage

```
partitionCountData(data, n.boots.part = 100, max.pop=500)
```

Arguments

`data`	A two-column data frame with the first column specifying the individual IDs and the second column specifying the number of times each individual was identified
`n.boots.part`	Number of parametric bootstraps to perform to obtain a null distribution of the test-statisic for partitioning the data (see details)
`max.pop`	The maximum population size used for fitting the Two-Innate Rates Model (see `fitTirm` for details)

Details

This function implements a new partitioning (PART) method that removes individuals detected a large number of times from the data. These individuals provide little information about the population size and, because they are inconsistent with the modeling assumption of just two capture rates, they cause estimation problems in capwire. Specifically, the PART algorithm attempts to remove very large capture counts and leave individuals captures between few and a moderate number of times.

The PART algorithm first fits the Two-Innate Rates Model (TIRM) to the entire data set. Assuming the distribution of capture counts can be approximated as either a two (null) or three (alternative) class multinomial distribution, the algorithm considers all possible ways to partition the count data under both the null and alternative distributions. The partitioning schemes which maximizes the respective likelihoods is retained. A test-statistic, the ratio of multinomial (log) likelihoods (3:2) is obtained for the observed data.

The MLE for number of individuals in class a, number of individuals in class b and the ratio of capture probabilities (alpha) are used to simulate `n.boots.part` data sets under the TIRM. The test-statistic is calculated for each data set as described above.

The observed test-statistic is then compared to the distribution of simulated test-statistics. When the observed test-statistic falls in the tail of the distribution of the test-statisitic from the simulated data, the null model can be rejected.

Value

`low.2.third`	A two-column data frame specifying the capture classes and the number of individuals in this capture class for the lower group

up.1.third	A two-column data frame specifying the capture classes and the number of individuals in this capture class for the upper group. This group will be excluded from the estimation procedure in fitTirmPartition
p.2.3	The p-value obtained from the parametric bootstraps for partitioning the data into the three groups

Note

This method is described in detail in Stansbury et al. in prep

The function partitionCountData is used internally in fitTirmPartition

Author(s)

Craig R. Miller and Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

Stansbury C.R., D.E. Ausband, P. Zager, C.M. Mack, C.R. Miller, M.W. Pennell, and L.P. Waits. in prep. Non-invasive genetic sampling of rendezvous sites and population estimation of grey wolves in Idaho, USA

See Also

fitTirmPartition, fitTirm

Examples

```
## Use dummy data set with a few individuals having very high capture counts
d <- c(1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,3,3,4,4,6,6,7,8,10,10,14,17,19,22,22,25)

data <- buildClassTable(d)

## Perform Partitioning Algorithm

part.data <- partitionCountData(data=data, n.boots.part=10, max.pop=500)

part.data
```

`simCapture` *Simulate data under various distributions of capture rates*

Description

Simulates capture count data where individual capture rates are assumed to be drawn from a specified distribution.

Data can be used as input for fitting Equal Capture Model (with `fitEcm`), Two-Innate Rates Model (with `fitTirm`) or the Two-Innate Rates Model with partitioned data (with `fitTirmPartition`)

Usage

```
simCapture(n, s, dist.func, return.cap.probs=FALSE)
```

Arguments

`n`	The number of individuals in the population
`s`	The total number of samples collected
`dist.func`	The distribution of capture rates within the population (see details)
`return.cap.probs`	Logical, signifying whether individual capture probabilities should be returned (`return.cap.probs=TRUE`) or not (`return.cap.probs=FALSE`)

Details

We assume that there is heterogeneity in the capturabilities of individuals within a population. That is, some individuals are more likely to be captured than others

We also assume that the individual capturabilities are drawn from some distribution.

The distribution is specified by the `dist.func` argument. `dist.func` takes a function with parameter n, where n specifies the number of samples to be drawn. `simCapture` can take any distribution of this form but the `capwire` package includes several functions which allow for users to draw capture rates from several standard distributions such as a uniform (`drawCapRatesUnif`), exponential (`drawCapRatesExp`), gamma (`drawCapRatesGamma`) and geometric (`drawCapRatesGeom`).

Value

If `return.cap.probs=FALSE`: A two-column data frame with the first column specifiying the capture class (i.e. individuals caught i times) and the second column specifying the number of individuals in each class.

If `return.cap.probs=TRUE`: A list consisting of two items 1) A three-column data frame with the first column specifiying the individual IDs, the second column specifying the number of times each individual was captured and the third denoting the capture probability associated with each individual and 2) A two-column data frame with the first column specifiying the capture class (i.e. individuals caught i times) and the second column specifying the number of individuals in each class.

Author(s)

Matthew W. Pennell

References

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

drawCapRatesUnif, drawCapRatesGeom, drawCapRatesExp, drawCapRatesBeta, drawCapRatesGamma

Examples

```
## Specify that capture probabilities are drawn from a uniform distribution

rates <- drawCapRatesUnif(lower=0.1, upper=1)

## Simulate data using the specified distribution

data <- simCapture(n=20, s=100, dist.func=rates, return.cap.probs=FALSE)

data

## Example of sampling capture probabilities from a non-standard distribution
## We want to use a truncated normal for this purpose

## Create a function which takes argumens mean, sd, and trunc.point
drawCapRates.truncnorm <- function(mean, sd, trunc.point){

## Draw a large number of samples from a normal distribution with specified mean and sd
normal.dist <- rnorm(100000, mean, sd)

## Truncate the distribution at the point specified by trunc.point
trunc.dist <- normal.dist[which(normal.dist >= trunc.point)]

## Create a function which draws n samples from trunc.dist
function(n){
sample(trunc.dist, size=n, replace=TRUE)
}

}

## Specify the distribution

my.dist <- drawCapRates.truncnorm(mean=2, sd=1, trunc.point=0)

## Look at the function that is returned

my.dist

## Simulate data under the specified distribution
```

```
my.data <- simCapture(n=20, s=100, dist.func=my.dist, return.cap.probs=TRUE)

my.data
```

---

simEcm *Simulate Data Under Equal Capture Model*

Description

Simulates capture count data under the assumptions of the Equal Capture Model (ECM), where all individuals are assumed to have an equal probability of being captured

Usage

```
simEcm(n, s)
```

Arguments

n	Number of individuals in the population
s	Total number of samples collected

Value

A two-column data frame with the first column specifiying the capture class (i.e. individuals caught i times) and the second column specifying the number of individuals in each class.

The data frame can be used as the data argument for any of the model-fitting functions implemented in capwire

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

fitEcm

Examples

```
## Simualte data under the Equal Capture Model

data <- simEcm(n=25, s=100)

## Fit the Equal Capture Model

ecm <- fitEcm(data=data, max.pop=200)

ecm
```

---

simTirm *Simulate Data Under Two-Innate Rates Model*

Description

Simulates capture count data under the assumptions of the Two-Innate Rates Model (TIRM) where the individuals in the population are assumed to come from two classes: easy to capture individuals and difficult to capture individuals.

Usage

```
simTirm(na, nb, alpha, s)
```

Arguments

na	Number of individuals in the "easy to capture" class
nb	Number of individuals in the "difficult to capture" class
alpha	The ratio of the capture rates between the two classes (capture rate of class a individual/capture rate of class b individuals)
s	Total number of samples collected

Value

A two-column data frame with the first column specifiying the capture class (i.e. individuals caught i times) and the second column specifying the number of individuals in each class.

The data frame can be used as the data argument for any of the model-fitting functions implemented in capwire

Author(s)

Matthew W. Pennell

References

Miller C. R., P. Joyce and L.P. Waits. 2005. A new method for estimating the size of small populations from genetic mark-recapture data. Molecular Ecology 14:1991-2005.

Pennell M.W., C.R. Stansbury, L.P. Waits and C.R. Miller. submitted. capwire: A R Package for Estimating Population Census Size from Non-Invasive Genetic Sampling

See Also

fitTirm

Examples

```
## Simualte data under Two-Innate Rates Model

data <- simTirm(na=20, nb=10, alpha=4, s=100)

## Fit the Two-Innate Rates Model

tirm <- fitTirm(data=data, max.pop=200)

tirm
```

---

wombat *Wombat Capture Count Data*

Description

This is a data set consisting of capture count data published in Banks et al. 2003 from northern hairy-nosed wombats (Lasiorhinus kretii). It is used as an example in the capwire tutorial.

Usage

```
data(wombat)
```

Format

A two-column data frame

capture.class: class assignment where individuals in class i were caught i times

No.Ind: the number of individuals in each capture.class

Details

This data set can be used as input for fitEcm, fitTirm, fitTirmPartition, partitionCountData.

Source

Banks, S. C., S. D. Hoyle, A. Horsup, P. Sunnucks, and A. C. Taylor. 2003. Demographic monitoring of an entire species (the northern hairy-nosed wombat, Lasiorhinus kretii) by genetic analysis of non-invasively collected material. Animal Conservation 6:101-107. DOI: 10.1017/S1367943003003135

Examples

```
data(wombat);
wombat
```

Index