



cbrATLAS

An R Software Package for estimating **ACTIVE TAG LIFE ADJUSTED SURVIVAL**

COLUMBIA BASIN RESEARCH

SCHOOL OF AQUATIC AND FISHERY SCIENCES

W UNIVERSITY of WASHINGTON

Package ‘cbrATLAS’

March 14, 2022

Title Active Tag Life Adjusted Survival Modeling

Version 0.0.1.3

Description cbrATLAS is based on a stand-alone program provided by Columbia Basin Research (University of Washington) to analyze active tag mark-recapture studies. The use of an active tag increases subsequent detection probabilities but relies on a signal that may fail due to battery or equipment failure. Without adjusting for this probability, estimated survival will be under-estimated. The survival adjustments rely on tag failure data from a tag-life study using representative sample(s) of tags. The adjustments to the survival estimates are conservative. See Townsend et al., (2006) <[doi:10.1198/108571106X111323](https://doi.org/10.1198/108571106X111323)> and Skalski et al., (1998) <[doi:10.1139/cjfas-55-6-1484](https://doi.org/10.1139/cjfas-55-6-1484)>.

License GPL-3

Imports failCompare,
devtools,
remotes

Depends R (>= 4.1.0)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

AdjSurv.fn	2
atlas2flat.fn	3
boot.L	4
cjs.fn	4
cjs.lik	5
cjs.taglife.corr	6
correct.fn	7
dayhr.fn	7
harmonic	8

mean_tt2site.fn	8
single.rel	9
taglife.data	9
thist0	10

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

AdjSurv.fn

Active Tag Life Adjusted Survival Model Fitting

Description

This function analyzes release-recapture data from a study that uses active-tag technology. It estimates survival and detection probabilities and, if provided with tag-life data, the average probability a tag is active at each detection site. These estimates are then used to adjust estimated survival for potential tag-failure that may otherwise be interpreted as mortality. Methods are documented in [Program ATLAS 1.4:Active Tag Life Adjusted Survival](#).

Usage

```
AdjSurv.fn(
  taghist.file,
  taghist.format = "atlas",
  taglife.file = NULL,
  taglife.model = NULL,
  num.release = 1,
  num.bootstrap = 250,
  adjust.cjs = T,
  rounding = 4,
  plot.taglife = T
)
```

Arguments

taghist.file	Required. Dataframe or a filename of .csv file with detection histories
taghist.format	Required. Format of tag detection histories "atlas" format based on Program ATLAS input files. Eight columns. Each tag has one line per possible detection site. "flat" format has one line per tag: tag.code, activation date, release date, and one column per detection site filled with first detection times at that site
taglife.file	Optional. Name of .csv file with taglife times in first column. Other columns ignored. Header expected
taglife.model	fc_obj. Failure time model object. (default = NULL). If no fc_obj is provided, function will try models available in failCompare and select best fit
num.release	(default = 1) If more than one release group, first column will be added to flat format file to denote group name (not implemented in this version)
num.bootstrap	(default = 1000) Number of resample iterations to estimate additional variance on survival estimates
adjust.cjs	(T/F) (default = T) adjust CJS estimates for estimated tag-life
rounding	(default = 4) Number of decimal places on estimate
plot.taglife	(T/F) (default = T) Plot the estimated tag-life curve

Details

All date-times are assumed to be character vector with format: "%m/%d/%Y %H:%M"

Value

Returns a list "out" with (if provided):

taghist tagfile filename

unadjusted.cjs table of unadjusted survival estimates

tagfile tag life filename

taglife.model taglife model name

mean.tag.pLive expect proportion tags active to each site

adjust.cjs adjusted survival estimates

atlas2flat.fn	<i>Converts ATLAS data file to a flat file format</i>
---------------	---

Description

cbrATLAS allows the mark-recapture data to be loaded in one of two file formats. The ATLAS file format (detailed in the [ATLAS 1.4 manual](#), page 17) is a vertical file with one line per detection site. The default input format for the cbrATLAS package uses a flat file format, which details the history of each tag on one line, including release and detection times at each site. This function converts an ATLAS formatted file to a flat file.

Usage

```
atlas2flat.fn(data.in)
```

Arguments

data.in	Table: columns = release group name, bin number, tag id, tag activation date/time, tag release date/time, site name, detection (1 = yes, 0 = no), detection date/time
---------	---

Value

Returns a table with columns: release group name, bin number, tag id, activation date/time, tag release date/time, one column per site name with detection time (blank if no detection)

boot.L	<i>Bootstraps observed "activation to 1st detection per site" times to get standard errors on estimated $p(L_i)$</i>
--------	---

Description

To better estimate the variance on the adjusted-for-tag-failure survival estimates, a bootstrap is conducted on both the data used to estimate tag-life and on the observed times the study tags were active. Statistical methods are described in the ATLAS 1.4 manual, Appendix B2.

Usage

```
boot.L(at.time.matrix, model.in, num.boots = 100)
```

Arguments

at.time.matrix	Matrix of time from activation to detection at site i
model.in	Output from taglife.fn
num.boots	(default = 100) Number of desired resampling bootstraps to estimate the standard error for each taglife estimate

Value

Returns a list called L.matrix that contains:

L.matrix means from bootstrapping that is used to correct survival

L2.matrix matrix of bootstrapped L_i w/resampled taglife tags and active times to detection

cjs.fn	<i>Summarizes detection histories and provides Cormack-Jolly-Seber estimates with or without adjustment of potential tag-failure</i>
--------	--

Description

CJS.fn summarizes the individual detection histories and estimates seeds for Cormack-Jolly-Seber (CJS) likelihood function. It is based on Skalski et al. (1998). This likelihood is also used to adjust survival for estimated tag life as described in Townsend et al. (2006). If no tag-life data provided, only the unadjusted CJS survival will be returned.

Usage

```
cjs.fn(detect.in, L.in = NULL, seeds.in = NULL, se.out = F, d = NULL)
```

Arguments

<code>detect.in</code>	Detection history in wide ATLAS format without <code>rel.group</code> or <code>bin</code> columns
<code>L.in</code>	<code>L.in</code> : Vector of mean probabilities tag is working given detection at each site. If <code>NULL</code> , function will estimate the CJS parameters without tag failure
<code>seeds.in</code>	Optional starting estimates for CJS model parameters S_i , p_i , l_i where i is 1:(total number of sites-1)
<code>se.out</code>	(TIF) Should standard error of cjs parameters be estimated. Uses variance matrix based on Hessian from MLE of cjs likelihood
<code>d</code>	proportion of detected tags censored at a location

Value

Returns a two-column matrix with CJS estimates and standard errors (if estimated)

<code>cjs.lik</code>	<i>A likelihood function for unadjusted- or adjusted-for-taglife Cormack-Jolly-Seber (CJS) estimates</i>
----------------------	--

Description

This is a likelihood function for unadjusted- or adjusted-for-taglife Cormack-Jolly-Seber (CJS) estimates. It is based on Skalski et al. (1998). Adjustment to survival for estimated tag life as described in Townsend et al. (2006).

Usage

```
cjs.lik(params, counts.in, num.period, use.hist, L.in = NULL, d.in = NULL)
```

Arguments

<code>params</code>	The likelihood MLE estimators (S, p, l) in a single vector
<code>counts.in</code>	Vector of summed detections by history (names are history)
<code>num.period</code>	Number of detection periods in history
<code>use.hist</code>	Matrix of unique detection histories
<code>L.in</code>	The probability that a tag is working at a detection site at a given time. Set to 1 if active tags not indicated. $p(\text{tag working})$ is cumulative to each site, not a multiplicate from site to site
<code>d.in</code>	The probability a tag is censored given that it's detected

Value

This function returns a negative log-likelihood

cjs.taglife.corr	<i>Estimates the length of time a tag is active (tag activation to first detection time)</i>
------------------	--

Description

This function uses the length of time a tag has been active to estimate the probability it would fail when detected. The average $P(\text{failure})$ is estimated at each site. The estimated survival is divided by the mean $P(\text{failure})$ at that site to adjust for the predicted tag failure in the study (Townsend et al. 2006).

Usage

```
cjs.taglife.corr(
  activetime.matrix,
  site.names = NULL,
  num.period = num.period,
  taglife.fit = taglife.fit,
  num.boots = 0,
  cjs.est = NULL
)
```

Arguments

activetime.matrix	Matrix of time elapsed after tag activation to first detection at site. One column per site
site.names	Vector of site designations
num.period	Number of detection sites
taglife.fit	Results from fitting tag-life study tags: Model name, estimated parameters, mean time to fail
num.boots	Number of Bootstrap iterations for variance on estimated $P(L_i)$ for each site. Uses 0 for initial fittings
cjs.est	2-column Matrix with unadjusted Cormack-Jolly-Seber estimates and standard errors

Value

returns a list with:

model.out name of model used to fit tag-life curve

L vector of probability tag active to each detection site

L.se vector of standard errors on each estimated L

adj.Si.se vector of estimated standard errors for adjusted survival estimates

 correct.fn

Restrains uncorrected probability estimates between 0 and 1

Description

Estimation of Cormack-Jolly-Seber (CJS) survival parameters can be greater than 1 when detection probabilities are low. To reduce computational failure when maximizing the likelihood, correct.fn bounds the CJS parameter estimates between 0 and 1 in the likelihood. This may result in survival estimates biased low, but low detection rates do not usually occur in studies with active tag technology.

Usage

```
correct.fn(x)
```

Arguments

x Value to be rounded up/down to be between 0 to 1

Value

Returns original value if between 0 and 1, otherwise rounds up to 1e-10 or down to 1-1e-10

dayhr.fn

Standardizes date-time input

Description

This function converts date and time into a "yyyy-mm-dd hh:mm:ss" format.

Usage

```
dayhr.fn(x, secs = T, AMPM = F)
```

Arguments

x Character vector of date-time values currently accepted as input formats
 "m/d/Y I:M:S p" AM/PM
 "Y-m-d H:M:S" or "m/d/Y H:M:S" with seconds
 "Y-m-d H:M" or "m/d/Y H:M" without seconds

secs (T|F) Seconds are included in time values

AMPM (T|F) Time format is in AM/PM format

Value

Returns a strptime object in "%Y-%m-%d %H:%M:%S" or "%Y-%m-%d %H:%M" format

Examples

```
## Not run: dayhr.fn("2021-05-10 14:22:34", secs=T , AMPM = F)
```

harmonic	<i>Estimates harmonic mean</i>
----------	--------------------------------

Description

This function estimates harmonic mean.

Usage

```
harmonic(t.in)
```

Arguments

t.in	Numeric vector (no NAs).
------	--------------------------

Value

This function returns a character vector of harmonic mean and standard error

mean_tt2site.fn	<i>Estimates travel time harmonic mean</i>
-----------------	--

Description

This function estimates the travel time harmonic mean.

Usage

```
mean_tt2site.fn(data.in, num.period, site.names)
```

Arguments

data.in	Detection history and detection times (flat format)
num.period	Number of detection sites
site.names	Vector of site names

Value

This function returns a list with:

harmonic.tt harmonic mean travel time

tt.matrix individual travel time, release to first detection time for each site (days)

activetime.matrix individual time tag active, time of tag activation to first detection for each site (days)

single.rel	<i>Single Release ATLAS Practice File.</i>
------------	--

Description

Single Release ATLAS Practice File.

Usage

single.rel

Format

These data are from a study of acoustic-tagged salmon using a release above McNary Dam in 2014. A data frame with 8 columns and 9996 rows in the ATLAS file format:

- V1** Release group (ie. R1)
- V2** tag lot number (ie. 1,2,...)
- V3** tag id code
- V4** tag activation date/time (yyyy-mm-dd hh:mm:ss)
- V5** release date/time (yyyy-mm-dd hh:mm:ss)
- V6** detecton site designation
- V7** detection indicator (0: not detected; 1: detected; 2: detected & censored)
- V8** tag detection date/time (yyyy-mm-dd hh:mm:ss) at that site ...

References

[McNary Dam survival study](#)

taglife.data	<i>Taglife ATLAS Practice File.</i>
--------------	-------------------------------------

Description

Taglife ATLAS Practice File.

Usage

taglife.data

Format

Taglife study results with length of time to failure for 100 tags. Tags were observed continually until failure. A data frame with 1 columns and 100 rows:

tag_life_days length of time from activation to tag failure in days.

References

[McNary Dam survival study](#)

thist0	<i>Creates a list with counts per history and corresponding matrix of unique histories</i>
--------	--

Description

This function creates a list with counts per history and a corresponding matrix of unique histories.

Usage

```
thist0(x)
```

Arguments

x	Detection history in wide ATLAS format without rel.group or bin columns
---	---

Value

This function returns a list with:

count total counts in the same order as unique histories in hist.matrix

hist.matrix unique detection matrix, columns indicate detection site. (0=not detected/1=detected)

Index

* datasets

- single.rel, [9](#)
- taglife.data, [9](#)

AdjSurv.fn, [2](#)
atlas2flat.fn, [3](#)

boot.L, [4](#)

cjs.fn, [4](#)
cjs.lik, [5](#)
cjs.taglife.corr, [6](#)
correct.fn, [7](#)

dayhr.fn, [7](#)

harmonic, [8](#)

mean_tt2site.fn, [8](#)

single.rel, [9](#)

taglife.data, [9](#)
thist0, [10](#)

