Package 'BTSPAS'

February 19, 2015

Version 2014.0901

Date 2014-09-01

Title Bayesian Time-Strat. Population Analysis

Author Carl J Schwarz <cschwarz@stat.sfu.ca> and
Simon J Bonner <simon.bonner@uky.edu>

Maintainer Carl J Schwarz <cschwarz@stat.sfu.ca>

Imports coda, splines, actuar, rjags, R2OpenBUGS, ggplot2, plyr

Suggests BRugs

SystemRequirements JAGS

Description BTSPAS provides advanced Bayesian methods to estimate
abundance and run-timing from temporally-stratified
Petersen mark-recapture experiments. Methods include
hierarchical modelling of the capture probabilities
and spline smoothing of the daily run size. This version
uses JAGS to sample from the posterior distribution.

License GPL (>= 2)

URL http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2

NeedsCompilation no

Repository CRAN

Date/Publication 2014-08-26 07:51:36

R topics documented:

BTSPAS-package *Bayesian Time-Stratified Population Analysis*

Description

BTSPAS provides advanced Bayesian methods to estimate abundance and run-timing from temporally-stratified Petersen mark-recapture experiments. Methods include hierarchical modelling of the capture probabilities and spline smoothing of the daily run size.

Full details are available at http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2

Details

Package:	BTSPAS
Version:	2014.09
Date:	2014-09-01
Depends:	R2WinBUGS, coda, splines, actuar, plyr, ggplot2
License:	GPL (>=2)
URL:	http://www.r-project.org
Repository:	R-Forge
Repository/R-Forge/Project:	btspas
Repository/R-Forge/Revision:	8
Date/Publication:	2014-09-02 02:49:34
Packaged:	2009-12-02 21:03:35 UTC; rforge
Built:	R 2.10.0; ; 2009-12-03 23:01:26 UTC; windows

Index:

Author(s)

Carl J Schwarz <cschwarz@stat.sfu.ca> and Simon J Bonner <s.bonner@stat.ubc.ca>

Maintainer: Simon J Bonner <s.bonner@stat.ubc.ca>

`BTSPAS-internal` *Internal functions not normally called by the user*

Description

These are internal functions not normally called by the user. (a) Compute the Predictive Posterior Distributions and make the Bayesian p-value plots. (b) Estimate the runtimings once the posterior distribution is found. (c) Compute simple Petersen estimates. (d) Generate initial values. (e) Miscellaneous functions.

`logit` *Logit and anti-logit function.*

Description

Compute the logit or anti-logit.

Usage

```
logit(p)
expit(theta)
```

Arguments

`p`	probability between 0 and 1.
`theta`	logit between -infinity and +infinity

Value

Computed logit or anti-logit

Author(s)

C.J.Schwarz <cschwarz@stat.sfu.ca>

Examples

```
##---- compute the logit and its inverse
logitp <- logit(.3)
p <- expit(-.84)
```

`TimeStratPetersenDiagErrorWHChinook_fit`

*Wrapper (*_fit) and function to call the Time Statified Petersen Estimator with Diagonal Entries and separating Wild from Hatchery Chinook function.*

Description

Takes the number of marked fish released, the number of recaptures, and the number of unmarked fish and uses Bayesian methods to fit a fit a spline through the population numbers and a hierarchical model for the trap efficiencies over time. The output is written to files and an MCMC object is also created with samples from the posterior.

Normally use the *_fit to pass the data to the fitting function.

Usage

```
TimeStratPetersenDiagErrorWHChinook_fit(title="TSPDE-WHChinook",
    prefix="TSPDE-WHChinook-",
    time, n1, m2, u2.A, u2.N, clip.frac.H, sampfrac, hatch.after = NULL,
    bad.m2 = c(), bad.u2.A=c(), bad.u2.N=c(),
    logitP.cov = rep(1, length(n1)),
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha = 1, tauU.beta = 0.05, taueU.alpha = 1, taueU.beta = 0.05,
    mu_xiP = logit(sum(m2, na.rm = TRUE)/sum(n1, na.rm = TRUE)),
    tau_xiP = 1/var(logit((m2 + 0.5)/(n1 + 1)), na.rm = TRUE),
    tauP.alpha = 0.001, tauP.beta = 0.001, run.prob = seq(0, 1, 0.1),
    debug = FALSE, debug2 = FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))

TimeStratPetersenDiagErrorWHChinook(
    title, prefix, time, n1, m2, u2.A, u2.N,
    hatch.after=NULL, clip.frac.H=.25,
    logitP.cov,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit(sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=1/var(logit((m2+.5)/(n1+1)), na.rm=TRUE),
    tauP.alpha=.001, tauP.beta=.001,
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
    InitialSeed)

TimeStratPetersenDiagErrorWHChinook2_fit(
    title="TSPDE-WHChinook2", prefix="TSPDE-WHChinook2-",
    time, n1, m2,
```

```
    u2.A.YoY, u2.N.YoY, u2.A.1, u2.N.1,
    clip.frac.H.YoY, clip.frac.H.1, sampfrac,
    hatch.after.YoY=NULL,
   bad.m2=c(), bad.u2.A.YoY=c(), bad.u2.N.YoY=c(), bad.u2.A.1=c(), bad.u2.N.1=c(),
    logitP.cov=rep(1,length(n1)),
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit(sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=1/var(logit((m2+.5)/(n1+1)), na.rm=TRUE),
    tauP.alpha=.001, tauP.beta=.001,
    run.prob=seq(0,1,.1),  # what percentiles of run timing are wanted
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))

TimeStratPetersenDiagErrorWHChinook2 (
    title, prefix, time, n1, m2,
    u2.A.YoY, u2.N.YoY, u2.A.1, u2.N.1,
    hatch.after.YoY=NULL,
    clip.frac.H.YoY=.25, clip.frac.H.1 = .25,
    logitP.cov,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit(sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=1/var(logit((m2+.5)/(n1+1)), na.rm=TRUE),
    tauP.alpha=.001, tauP.beta=.001,
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
    InitialSeed)
```

Arguments

title	A character string used for a title on reports and graphs
prefix	A character string used as the prefix for created files. All created graph files are of the form prefix-xxxxx.pdf.
time	A numeric vector of time used to label the strata. For example, this could be julian week for data stratified at a weekly level.
n1	A numeric vector of the number of marked fish released in each time stratum.
m2	A numeric vector of the number of marked fish from n1 that are recaptured in each time stratum. All recaptures take place within the stratum of release. Use the TimeStratPetersenNonDiagError_fit function for cases where recaptures take place outside the stratum of release. **For the Chinook problem, only a fraction of hatchery raised fish have adipose fin clipped. consequently, unclipped fish are a mixture of wild and hatchery fish.**
u2.A	A numeric vector of the number of unmarked fish with adipose clips captured in each stratum.
u2.N	A numeric vector of the number of unmarked fish with NO-adipose clips captured in each stratum.

`u2.A.YoY` A numeric vector of the number of unmarked YoY fish with adipose clips captured in each stratum.

`u2.N.YoY` A numeric vector of the number of unmarked YoY fish with adipose clips captured in each stratum.

`u2.A.1` A numeric vector of the number of unmarked Age1 fish with adipose clips captured in each stratum.

`u2.N.1` A numeric vector of the number of unmarked Age1 fish with adipose clips captured in each stratum.

`clip.frac.H` A numeric value for the fraction of the hatchery fish that have the adipose fin clipped (between 0 and 1).

`clip.frac.H.YoY` A numeric value for the fraction of the YoY hatchery fish that have the adipose fin clipped (between 0 and 1).

`clip.frac.H.1` A numeric value for the fraction of the Age1 hatchery fish that have the adipose fin clipped (between 0 and 1).

`sampfrac` A numeric vector with entries between 0 and 1 indicating what fraction of the stratum was sampled. For example, if strata are calendar weeks, and sampling occurred only on 3 of the 7 days, then the value of `sampfrac` for that stratum would be 3/7.

`hatch.after` A numeric vector with elements belonging to `time`. At which point do hatchery fish arrive? They arrive in the immediate stratum AFTER these entries.

`hatch.after.YoY` A numeric vector with elements belonging to `time`. At which point do YoY hatchery fish arrive? They arrive in the immediate stratum AFTER these entries.

`bad.m2` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of recovered fish should be ignored. For example, poor handling is suspected to induce handling induced mortality in the marked fish and so only very few are recovered. The values of `m2` will be set to NA for these strata.

`bad.u2.N` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked fish with NO adipose fin clip should be ignored.

`bad.u2.N.YoY` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked YoY with NO adipose fin clip should be ignored.

`bad.u2.A` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked fish with an adipose fin clip should be ignored.

`bad.u2.A.YoY` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked YoY with an adipose fin clip should be ignored.

`bad.u2.N.1` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked Age1 with NO adipose fin clip should be ignored.

bad.u2.A.1	A numeric vector with elements belonging to time. In some cases, something goes wrong in the stratum, and the number of unmarked Age1 with an adipose fin clip should be ignored.
logitP.cov	A numeric matrix for covariates to fit the logit(catchability). Default is a single intercept, i.e. all strata have the same mean logit(catchability).
n.chains	Number of parallel MCMC chains to fit.
n.iter	Total number of MCMC iterations in each chain.
n.burnin	Number of burn-in iterations.
n.sims	Number of simulated values to keeps for posterior distribution.
tauU.alpha	One of the parameters along with tauU.beta for the prior for the variance of the random noise for the smoothing spline.
tauU.beta	One of the parameters along with tauU.alpha for the prior for the variance of the random noise for the smoothing spline.
taueU.alpha	One of the parameters along with taueU.beta for the prior for the variance of noise around the spline.
taueU.beta	One of the parameters along with taueU.alpha for the prior for the variance of noise around the spline.
mu_xiP	One of the parameters for the prior for the mean of the logit(catchability) across strata
tau_xiP	One of the parameter for the prior for the mean of the logit(catchability) across strata
tauP.alpha	One of the parameters for the prior for the variance in logit(catchability) among strata
tauP.beta	One of the parameters for the prior for the variance in logit(catchability) among strata
run.prob	Numeric vector indicating percentiles of run timing should be computed.
debug	Logical flag indicating if a debugging run should be made. In the debugging run, the number of samples in the posterior is reduced considerably for a quick turn around.
debug2	Logical flag indicated if additional debugging information is produced. Normally the functions will halt at browser() calls to allow the user to peek into the internal variables. Not useful except to package developers.
engine	Which MCMC sampler should be used. JAGS=default, OpenBugs=alternate. Case not relevant.
InitialSeed	Numeric value used to initialize the random numbers used in the MCMC iterations.

Value

An MCMC object with samples from the posterior distribution. A series of graphs and text file are also created in the working directory.

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

References

Refer to the Trinity River Restoration Project report by Schwarz, C.J. et al. (2009) available at http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2. Please contact <cschwarz@stat.sfu.ca> for more details.

Examples

```
##---- See the demo files for examples of how to use this package
##
##    demo("demo-TSPDE-WHchinook",    package='BTSPAS', ask=FALSE)  # the simplest usage
##    demo("demo-TSPDE-WHchinook2",   package='BTSPAS', ask=FALSE)  # the simplest usage
##
```

TimeStratPetersenDiagErrorWHSteel_fit

*Wrapper (*_fit) and function to call the Time Statified Petersen Estimator with Diagonal Entries and separating Wild from Hatchery Steelhead function.*

Description

Takes the number of marked fish released, the number of recaptures, and the number of unmarked fish and uses Bayesian methods to fit a fit a spline through the population numbers and a hierarchical model for the trap efficiencies over time. The output is written to files and an MCMC object is also created with samples from the posterior.

Normally, data is passed to the wrapper which then calls the fitting function.

Usage

```
TimeStratPetersenDiagErrorWHSteel_fit(title="TSPDE-WHSteel", prefix="TSPDE-WHSteel-",
    time, n1, m2, u2.W.YoY, u2.W.1, u2.H.1, sampfrac, hatch.after = NULL,
    bad.m2 = c(), bad.u2.W.YoY=c(), bad.u2.W.1=c(), bad.u2.H.1=c(),
    logitP.cov = rep(1, length(n1)),
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha = 1, tauU.beta = 0.05, taueU.alpha = 1, taueU.beta = 0.05,
    mu_xiP = logit(sum(m2, na.rm = TRUE)/sum(n1, na.rm = TRUE)),
    tau_xiP = 1/var(logit((m2 + 0.5)/(n1 + 1)), na.rm = TRUE),
    tauP.alpha = 0.001, tauP.beta = 0.001, run.prob = seq(0, 1, 0.1),
    debug = FALSE, debug2 = FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))
```

```
TimeStratPetersenDiagErrorWHSteel(
    title, prefix,
    time, n1, m2, u2.W.YoY, u2.W.1, u2.H.1,
    hatch.after=NULL,
    logitP.cov,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit( sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=1/var( logit((m2+.5)/(n1+1)),na.rm=TRUE),
    tauP.alpha=.001, tauP.beta=.001,
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
    InitialSeed)
```

Arguments

`title`	A character string used for a title on reports and graphs
`prefix`	A character string used as the prefix for created files. All created graph files are of the form prefix-xxxxx.pdf.
`time`	A numeric vector of time used to label the strata. For example, this could be julian week for data stratified at a weekly level.
`n1`	A numeric vector of the number of marked fish released in each time stratum.
`m2`	A numeric vector of the number of marked fish from n1 that are recaptured in each time stratum. All recaptures take place within the stratum of release. Use the `TimeStratPetersenNonDiagError_fit` function for cases where recaptures take place outside the stratum of release. **For the Steelhead, all hatchery raised fish are clipped and so any unclipped fish is known to be a wild fish.**
`u2.W.YoY`	A umeric vector of the number of unmarked wild Young-of-Year fish captured in each stratum.
`u2.W.1`	A umeric vector of the number of unmarked wild age 1+ fish captured in each stratum.
`u2.H.1`	A umeric vector of the number of unmarked hatchery age 1+ fish (i.e. adipose fin clipped) captured in each stratum.
`sampfrac`	A numeric vector with entries between 0 and 1 indicating what fraction of the stratum was sampled. For example, if strata are calendar weeks, and sampling occurred only on 3 of the 7 days, then the value of `sampfrac` for that stratum would be 3/7.
`hatch.after`	A numeric vector with elements belonging to `time`. At which point do hatchery fish arrive? They arrive in the immediate stratum AFTER these entries.
`bad.m2`	A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of recovered fish should be ignored. For example, poor handling is suspected to induce handling induced mortality in the marked fish and so only very few are recovered. The values of `m2` will be set to NA for these strata.

bad.u2.W.YoY	A numeric vector with elements belonging to time. In some cases, something goes wrong in the stratum, and the number of wild unmarked Young-of-Year fish should be ignored.
bad.u2.W.1	A numeric vector with elements belonging to time. In some cases, something goes wrong in the stratum, and the number of wild unmarked age 1+ fish should be ignored.
bad.u2.H.1	A numeric vector with elements belonging to time. In some cases, something goes wrong in the stratum, and the number of hatchery unmarked (but adipose fin clipped) age 1+ fish should be ignored.
logitP.cov	A numeric matrix for covariates to fit the logit(catchability). Default is a single intercept, i.e. all strata have the same mean logit(catchability).
n.chains	Number of parallel MCMC chains to fit.
n.iter	Total number of MCMC iterations in each chain.
n.burnin	Number of burn-in iterations.
n.sims	Number of simulated values to keeps for posterior distribution.
tauU.alpha	One of the parameters along with tauU.beta for the prior for the variance of the random noise for the smoothing spline.
tauU.beta	One of the parameters along with tauU.alpha for the prior for the variance of the random noise for the smoothing spline.
taueU.alpha	One of the parameters along with taueU.beta for the prior for the variance of noise around the spline.
taueU.beta	One of the parameters along with taueU.alpha for the prior for the variance of noise around the spline.
mu_xiP	One of the parameters for the prior for the mean of the logit(catchability) across strata
tau_xiP	One of the parameter for the prior for the mean of the logit(catchability) across strata
tauP.alpha	One of the parameters for the prior for the variance in logit(catchability) among strata
tauP.beta	One of the parameters for the prior for the variance in logit(catchability) among strata
run.prob	Numeric vector indicating percentiles of run timing should be computed.
debug	Logical flag indicating if a debugging run should be made. In the debugging run, the number of samples in the posterior is reduced considerably for a quick turn around.
debug2	Logical flag indicated if additional debugging information is produced. Normally the functions will halt at browser() calls to allow the user to peek into the internal variables. Not useful except to package developers.
engine	Which MCMC sampler should be used. JAGS=default, alternate=OpenBugs. Case is not important.
InitialSeed	Numeric value used to initialize the random numbers used in the MCMC iterations.

Value

An MCMC object with samples from the posterior distribution. A series of graphs and text file are also created in the working directory.

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

References

Refer to the Trinity River Restoration Project report by Schwarz, C.J. et al. (2009) available at http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2. Please contact <cschwarz@stat.sfu.ca> for more details.

Examples

```
##---- See the demo files for examples of how to use this package
##
##     demo("demo-TSPDEWHSteel",     package='BTSPAS', ask=FALSE)  # the simplest usage
##
```

TimeStratPetersenDiagError_fit

*Wrapper (*_fit) and function to call the Time Statified Petersen Estimator with Diagonal Entries function.*

Description

Takes the number of marked fish released, the number of recaptures, and the number of unmarked fish and uses Bayesian methods to fit a fit a spline through the population numbers and a hierarchical model for the trap efficiencies over time. The output is written to files and an MCMC object is also created with samples from the posterior.

Normally, the wrapper (*_fit) function is called which then calls the fitting routine.

Use the TimeStratPetersenNonDiagError_fit function for cases where recaptures take place outside the stratum of release.

Usage

```
TimeStratPetersenDiagError_fit(title="TSDPE", prefix="TSPDE-",
    time, n1, m2, u2, sampfrac,
    jump.after = NULL, bad.n1=c(), bad.m2 = c(), bad.u2=c(),
    logitP.cov = rep(1, length(n1)),
    n.chains = 3, n.iter = 2e+05, n.burnin = 1e+05, n.sims = 2000,
    tauU.alpha = 1, tauU.beta = 0.05, taueU.alpha = 1, taueU.beta = 0.05,
    mu_xiP = logit(sum(m2, na.rm = TRUE)/sum(n1, na.rm = TRUE)),
    tau_xiP = 1/var(logit((m2 + 0.5)/(n1 + 1)), na.rm = TRUE),
```

```
    tauP.alpha = 0.001, tauP.beta = 0.001, run.prob = seq(0, 1, 0.1),
    debug = FALSE, debug2 = FALSE,
    engine=c("jags","openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))

TimeStratPetersenDiagError(title, prefix, time, n1, m2, u2, jump.after = NULL,
    logitP.cov,
    n.chains = 3, n.iter = 2e+05, n.burnin = 1e+05, n.sims = 2000,
    tauU.alpha = 1, tauU.beta = 0.05, taueU.alpha = 1, taueU.beta = 0.05,
    mu_xiP = logit(sum(m2, na.rm = TRUE)/sum(n1, na.rm = TRUE)),
    tau_xiP = 1/var(logit((m2 + 0.5)/(n1 + 1)), na.rm = TRUE),
    tauP.alpha = 0.001, tauP.beta = 0.001, debug = FALSE, debug2 = FALSE,
    engine=c("jags","openbugs")[1],
    InitialSeed )
```

Arguments

`title`	A character string used for a title on reports and graphs
`prefix`	A character string used as the prefix for created files. All created graph files are of the form prefix-xxxxx.pdf.
`time`	A numeric vector of time used to label the strata. For example, this could be julian week for data stratified at a weekly level.
`n1`	A numeric vector of the number of marked fish released in each time stratum.
`m2`	A numeric vector of the number of marked fish from n1 that are recaptured in each time stratum. All recaptures take place within the stratum of release.
`u2`	A numeric vector of the number of unmarked fish captured in each stratum. These will be expanded by the capture efficiency to estimate the population size in each stratum.
`sampfrac`	A numeric vector with entries between 0 and 1 indicating what fraction of the stratum was sampled. For example, if strata are calendar weeks, and sampling occurred only on 3 of the 7 days, then the value of `sampfrac` for that stratum would be 3/7.
`jump.after`	A numeric vector with elements belonging to `time`. In some cases, the spline fitting the population numbers should be allowed to jump. For example, the population size could take a jump when hatchery released fish suddenly arrive at the trap. The jumps occur AFTER the strata listed in this argument.
`bad.n1`	A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of marked fish released should be ignored. The values of `m2` for this stratum will also be set to NA for these strata.
`bad.m2`	A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of recovered fish should be ignored. For example, poor handling is suspected to induce handling induced mortality in the marked fish and so only very few are recovered. The values of `m2` will be set to NA for these strata.
`bad.u2`	A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked fish should be ignored.

	For example, the trap didn't work properly in this stratum. The values of u2 will be set to NA for these strata.
logitP.cov	A numeric matrix for covariates to fit the logit(catchability). Default is a single intercept, i.e. all strata have the same mean logit(catchability).
n.chains	Number of parallel MCMC chains to fit.
n.iter	Total number of MCMC iterations in each chain.
n.burnin	Number of burn-in iterations.
n.sims	Number of simulated values to keeps for posterior distribution.
tauU.alpha	One of the parameters along with tauU.beta for the prior for the variance of the random noise for the smoothing spline.
tauU.beta	One of the parameters along with tauU.alpha for the prior for the variance of the random noise for the smoothing spline.
taueU.alpha	One of the parameters along with taueU.beta for the prior for the variance of noise around the spline.
taueU.beta	One of the parameters along with taueU.alpha for the prior for the variance of noise around the spline.
mu_xiP	One of the parameters for the prior for the mean of the logit(catchability) across strata
tau_xiP	One of the parameter for the prior for the mean of the logit(catchability) across strata
tauP.alpha	One of the parameters for the prior for the variance in logit(catchability) among strata
tauP.beta	One of the parameters for the prior for the variance in logit(catchability) among strata
run.prob	Numeric vector indicating percentiles of run timing should be computed.
debug	Logical flag indicating if a debugging run should be made. In the debugging run, the number of samples in the posterior is reduced considerably for a quick turn around.
debug2	Logical flag indicated if additional debugging information is produced. Normally the functions will halt at browser() calls to allow the user to peek into the internal variables. Not useful except to package developers.
engine	Which MCMC sampler should be used. Default is "JAGS". Also available is "OpenBugs" or "Winbugs". Case not relevant.
InitialSeed	Numeric value used to initialize the random numbers used in the MCMC iterations.

Value

An MCMC object with samples from the posterior distribution. A series of graphs and text file are also created in the working directory.

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

References

Refer to the Trinity River Restoration Project report by Schwarz, C.J. et al. (2009) available at http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2. Please contact <cschwarz@stat.sfu.ca> for more details.

Examples

```
##---- See the demo files for examples of how to use this package
##
##     demo("demo-TSPDE",     package='BTSPAS', ask=FALSE)  # the simplest usage
##    demo("demo-TSPDE-cov", package='BTSPAS', ask=FALSE)  # including a covariate for logit(P)
##
```

TimeStratPetersenNonDiagError_fit

*Wrapper (*_fit) and function to call the Time Statified Petersen Estimator with NON Diagonal Entries function.*

Description

Takes the number of marked fish released, the number of recaptures, and the number of unmarked fish and uses Bayesian methods to fit a fit a spline through the population numbers and a hierarchical model for the trap efficiencies over time. The output is written to files and an MCMC object is also created with samples from the posterior.

Normally the user makes a call to the *_fit function which then calls the fitting function.

Use the TimeStratPetersenDiagError_fit function for cases where recaptures take place ONLY in the stratum of release, i.e. the diagonal case.

The non-diagonal case fits a log-normal distribution for the travel time. The *NP functions fit a non-parametric distribution for the travel times. The *MarkAvail functions extend the *NP functions to allow for reductions in mark availability because of fall back, immediate tagging mortality, etc.

Usage

```
TimeStratPetersenNonDiagError_fit(title="TSPNDE", prefix="TSPNDE-",
    time, n1, m2, u2, sampfrac, jump.after = NULL,
    bad.n1=c(), bad.m2 = c(), bad.u2=c(),
    logitP.cov = rep(1, length(u2)),
    logitP.fixed=NULL, logitP.fixed.values=NULL,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha = 1, tauU.beta = 0.05, taueU.alpha = 1, taueU.beta = 0.05,
    mu_xiP = logit(sum(m2, na.rm = TRUE)/sum(n1, na.rm = TRUE)),
```

```
    tau_xiP = .6666,
    tauP.alpha = 0.001, tauP.beta = 0.001, run.prob = seq(0, 1, 0.1),
    debug = FALSE, debug2 = FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))


TimeStratPetersenNonDiagError (
    title, prefix, time, n1, m2, u2,
    jump.after=NULL,
    logitP.cov, logitP.fixed=rep(NA,length(u2)),
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=-2, tau_xiP=.6666,
    tauP.alpha=.001, tauP.beta=.001,
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
    InitialSeed)

TimeStratPetersenNonDiagErrorNP_fit(
    title="TSPNDENP", prefix="TSPNDENP-",
    time, n1, m2, u2, sampfrac, jump.after=NULL,
    bad.n1=c(), bad.m2=c(), bad.u2=c(),
    logitP.cov=rep(1,length(u2)),
    logitP.fixed=NULL, logitP.fixed.values=NULL,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit(sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=.6666,   # need a better initial value for variation in catchability
    tauP.alpha=.001, tauP.beta=.001,
    Delta.max=NULL,
    prior.muTT=NULL,  # prior on movements
    tauTT.alpha=.1,tauTT.beta=.1,
    run.prob=seq(0,1,.1),  # what percentiles of run timing are wanted
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))


TimeStratPetersenNonDiagErrorNP (
     title, prefix, time, n1, m2, u2,
     jump.after=NULL,
     logitP.cov=rep(1,length(u2)), logitP.fixed=rep(NA,length(u2)),
     n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
     tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
     Delta.max,
     mean.muTT=rep(0,Delta.max),sd.muTT=rep(sqrt(.666),Delta.max),
     tauTT.alpha=.1, tauTT.beta=.1,
```

```
     mu_xiP=-2, tau_xiP=.6666,
     tauP.alpha=.001, tauP.beta=.001,
     debug=FALSE, debug2=FALSE,
     engine=c('jags',"openbugs")[1],
     InitialSeed)

TimeStratPetersenNonDiagErrorNPMarkAvail_fit(
    title="TSPNDENP", prefix="TSPNDENP-",
    time, n1, m2, u2, sampfrac, jump.after=NULL,
    bad.n1=c(), bad.m2=c(), bad.u2=c(),
    logitP.cov=rep(1,length(u2)),
    logitP.fixed=NULL, logitP.fixed.values=NULL,
    marked_available_n, marked_available_x,
    n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
    tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
    mu_xiP=logit(sum(m2,na.rm=TRUE)/sum(n1,na.rm=TRUE)),
    tau_xiP=.6666,   # need a better initial value for variation in catchability
    tauP.alpha=.001, tauP.beta=.001,
    Delta.max=NULL,tauTT.alpha=.1,tauTT.beta=.1,
    run.prob=seq(0,1,.1),  # what percentiles of run timing are wanted
    debug=FALSE, debug2=FALSE,
    engine=c('jags',"openbugs")[1],
   InitialSeed=ceiling(runif(1,min=0, max=if(engine=="jags"){1000000}else{14})))


TimeStratPetersenNonDiagErrorNPMarkAvail (
     title, prefix, time, n1, m2, u2,
     jump.after=NULL,
     logitP.cov=rep(1,length(u2)), logitP.fixed=rep(NA,length(u2)),
     ma.p.alpha, ma.p.beta,
     n.chains=3, n.iter=200000, n.burnin=100000, n.sims=2000,
     tauU.alpha=1, tauU.beta=.05, taueU.alpha=1, taueU.beta=.05,
     Delta.max,tauTT.alpha=.1, tauTT.beta=.1,
     mu_xiP=-2, tau_xiP=.6666,
     tauP.alpha=.001, tauP.beta=.001,
     debug=FALSE, debug2=FALSE,
     engine=c('jags',"openbugs")[1],
     InitialSeed)
#
```

Arguments

title	A character string used for a title on reports and graphs
prefix	A character string used as the prefix for created files. All created graph files are of the form prefix-xxxxx.pdf.
time	A numeric vector of time used to label the strata. For example, this could be julian week for data stratified at a weekly level.
n1	A numeric vector of the number of marked fish released in each time stratum.

`m2` A numeric matrix of the number of fish released in stratum [i] and recovered in [j-1] strata later. For example m2[3,5] is the number of marked fish released in stratum 3 and recovered 4 strata later in stratum 7. The first column is the number of marked fish recovered in the stratum of release, i.e. 0 strata later. Use the `TimeStratPetersenDiagError_fit` function for cases where recaptures take place ONLY in the stratum of release, i.e. the diagonal case.

`u2` A numeric vector of the number of unmarked fish captured in each stratum. These will be expanded by the capture efficiency to estimate the population size in each stratum.

`sampfrac` A numeric vector with entries between 0 and 1 indicating what fraction of the stratum was sampled. For example, if strata are calendar weeks, and sampling occurred only on 3 of the 7 days, then the value of `sampfrac` for that stratum would be 3/7.

`jump.after` A numeric vector with elements belonging to `time`. In some cases, the spline fitting the population numbers should be allowed to jump. For example, the population size could take a jump when hatchery released fish suddenly arrive at the trap. The jumps occur AFTER the strata listed in this argument.

`bad.n1` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of marked fish releases should be discarded. The values of `n1` will be set to NA for these strata.

`bad.m2` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of recovered marked fish should be ignored. For example, poor handling is suspected to induce handling induced mortality in the marked fish and so only very few are recovered. The values of `m2` in the entire row will be set to NA for these strata.

`bad.u2` A numeric vector with elements belonging to `time`. In some cases, something goes wrong in the stratum, and the number of unmarked fish captred should be ignored. The values of `u2` in the entire row will be set to NA for these strata.

`logitP.cov` A numeric matrix for covariates to fit the logit(catchability). Default is a single intercept, i.e. all strata have the same mean logit(catchability).

`logitP.fixed` A numeric vector (could be null) of the time strata where the logit(P) whould be fixed. Typically, this is used when the capture rates for some strata are 0 and logit(P) is set to -10 for these strata. The fixed values are given in `logitP.fixed.values`

`logitP.fixed.values`
A numerical vector (could be null) of the fixed values for logit(P) at strata given by logitP.fixed. Typically this is used when certain strata have a 0 capture rate and the fixed value is set to -10 which on the logit scale gives p[i] essentially 0. Don't specify values such as -50 because numerical problems could occur in WinBugs/OpenBugs.

`marked_available_n`
Information, usually from prior studies, on the fraction of marks that will be available. The *_n and *_x are used to create a "binomial" distribution for information on the marked availability. For example, if *_n=66 and *_x=40, then you estimate that about 40/66=61%.

`marked_available_x`
: See marked_available_n

The following parameters are usually not set by the user.

`n.chains` : Number of parallel MCMC chains to fit.

`n.iter` : Total number of MCMC iterations in each chain.

`n.burnin` : Number of burn-in iterations.

`n.sims` : Number of simulated values to keeps for posterior distribution.

`prior.muTT` : Information on prior movement. For example c(1,4,3,2) would have a prior of .1, .4, .3, and .2. A prior of c(10,40,30,20) would have same values but stronger information.

`mean.muTT` : The values of prior.muTT are converted into the mean of muTT using the make.muTT.prior function.

`sd.muTT` : The values of prior.muTT are converted into the sd of muTT using the make.muTT.prior function.

`tauU.alpha` : One of the parameters along with `tauU.beta` for the prior for the variance of the random noise for the smoothing spline.

`tauU.beta` : One of the parameters along with `tauU.alpha` for the prior for the variance of the random noise for the smoothing spline.

`taueU.alpha` : One of the parameters along with `taueU.beta` for the prior for the variance of noise around the spline.

`taueU.beta` : One of the parameters along with `taueU.alpha` for the prior for the variance of noise around the spline.

`Delta.max` : Maximum transition time for marked fish, i.e. all fish assumed to have moved by Delta.max unit of time

`tauTT.alpha` : One of the parameters along with `tauTT.beta` for the prior on 1/var of logit continuation ratio for travel times

`tauTT.beta` : One of the parameters along with `tauTT.alpha` for the prior on 1/var of logit continuation ratio for travel times

`mu_xiP` : One of the parameters for the prior for the mean of the logit(catchability) across strata

`ma.p.alpha` : One of the parameters for the prior for the marked_availability = beta(ma.p.alpha, ma.p.beta)

`ma.p.beta` : One of the parameters for the prior for the marked_availability = beta(ma.p.alpha, ma.p.beta)

`tau_xiP` : One of the parameter for the prior for the mean of the logit(catchability) across strata

`tauP.alpha` : One of the parameters for the prior for the variance in logit(catchability) among strata

`tauP.beta` : One of the parameters for the prior for the variance in logit(catchability) among strata

run.prob	Numeric vector indicating percentiles of run timing should be computed.
debug	Logical flag indicating if a debugging run should be made. In the debugging run, the number of samples in the posterior is reduced considerably for a quick turn around.
debug2	Logical flag indicated if additional debugging information is produced. Normally the functions will halt at browser() calls to allow the user to peek into the internal variables. Not useful except to package developers.
engine	Which MCMC sampler to use. JAGS=default, OpenBugs=alternate. Case not important.
InitialSeed	Numeric value used to initialize the random numbers used in the MCMC iterations.

Value

An MCMC object with samples from the posterior distribution. A series of graphs and text file are also created in the working directory.

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

References

Refer to the Trinity River Restoration Project report by Schwarz, C.J. et al. (2009) available at http://www.stat.sfu.ca/~cschwarz/Consulting/Trinity/Phase2. Please contact <cschwarz@stat.sfu.ca> for more details.

Examples

```
##---- See the demo files for examples of how to use this package
##
##     demo("demo-TSPNDE",     package='BTSPAS', ask=FALSE)  # the simplest usage
##
```

TimeToTargetRunSize *Computes and plots posterior distribution of time to get target run size. For example, the time to reach a cumulative run of 10,000 fish.*

Description

Takes a sim.list object from the MCMC runs, computes the posterior distribution of the time to the target runsize, plots the posterior

Usage

```
TimeToTargetRunSize( U, time, targetU, file_prefix, ci_prob=0.95)
```

Arguments

U	Elements of sim.list from MCMC object for U - the estimate runsize in each stratum
time	Vector of stratum time indices
targetU	The targeted cumulative run size. E.g. 10,000
file_prefix	Character string giving prefix for plot. A plot will be produced of the posterior in the filename paste(file_prefix,"-target.pdf",sep="")).
ci_prob	What size of credible interval should be computed?

Value

A list with a sample of the posterior (index), quantiles (quantiles), mean (mean), median(median), and standard deviation (sd), and target value (targetU)

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

Examples

```
## Not run:
# Compute the posterior of time to reach 10,000 fish. Results contains the MCMC object
#
results$TimeToTargetRunSize <- TimeToTargetRunSize(
        U=results$sims.list$U,
        time=results$data$time,
        targetU=10000,
        file_prefix = 'Time10000')


## End(Not run)
```

---

trace_plot *Creates trace plots of specified parameters showing the multiple chains and the value of Rhat*

Description

Takes the MCMC object returned from a split and produces trace_plots for the listed parameters. It shows a separate line on the plot for each chain and also shows the value of Rhat

Usage

```
trace_plot(title=" ", results=NULL, parms_to_plot=NULL, panels=c(1,1),
   mai=if(prod(panels)>1){c(.4,.4,.4,.4)} else {c(1.02,0.82,0.82,0.42)},
   cex=if(prod(panels)>1){.5} else {1}
   )
```

Arguments

title	A character string used for a title on reports and graphs
results	The MCMC object containing the results from the call to WinBugs/OpenBugs
parms_to_plot	A character vector of names of parameters to plot. These must match exactly to the parameter names used in the simulation.
panels	How many plots to put on a page. It used split.screen to format the plots.
mai	Margins (inches) from the 4 borders. See par().
cex	Character expansion factor. See par() for more details.

Value

Nothing returned. Creates the plots

Author(s)

Bonner, S.J. <s.bonner@stat.ubc.ca> and Schwarz, C. J. <cschwarz@stat.sfu.ca>

Examples

```
## Not run:
# Create trace plots of the logitP parameters
#
# extract the names of the variables from the MCMC object
varnames <- names(results$sims.array[1,1,])
# get the parms that start with logitP
parm.names <- varnames[grep("^logitP", varnames)]
# create a pdf file of the plots
pdf(file="trace-logitP.pdf",sep=""))
trace_plot(title=title, results=results,
    parms_to_plot=parm.names, panels=c(3,2))
dev.off()

# Or if you want an interactive display
par(ask=TRUE)
trace_plot(title=title, results=results,
    parms_to_plot=parm.names, panels=c(2,1))

## End(Not run)
```

Index