Package ‘fitode’

Type Package
Title Tools for Ordinary Differential Equations Model Fitting
Version 0.1.1
Description Methods and functions for fitting ordinary differential equations (ODE) model in ‘R’. Sensitivity equations are used to compute the gradients of ODE trajectories with respect to underlying parameters, which in turn allows for more stable fitting. Other fitting methods, such as MCMC (Markov chain Monte Carlo), are also available.
License GPL (>= 2)
Depends R (>= 4.0), bbmle
Imports deSolve, Deriv, MASS, numDeriv, mvtnorm, coda, methods
Suggests knitr, ggplot2, testthat (>= 3.0.0)
VignetteBuilder knitr
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
Config/testthat/edition 3
NeedsCompilation no
Author Sang Woo Park [aut, cre] (<https://orcid.org/0000-0003-2202-3361>), Ben Bolker [aut] (<https://orcid.org/0000-0002-2127-0443>)
Maintainer Sang Woo Park <swp2@princeton.edu>
Repository CRAN
Date/Publication 2022-10-29 08:50:05 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>blowfly</td>
<td>2</td>
</tr>
<tr>
<td>coef,fitode-method</td>
<td>3</td>
</tr>
<tr>
<td>coef,fitodeMCMC-method</td>
<td>3</td>
</tr>
<tr>
<td>confint,fitode-method</td>
<td>4</td>
</tr>
<tr>
<td>confint,fitodeMCMC-method</td>
<td>5</td>
</tr>
</tbody>
</table>
blowfly

Nicholson's blowfly data

Description
...

Usage

blowfly

Format

A data frame containing 361 rows comprising:

eggs  number of eggs
nonemerging  ?
emerging  ?
**coef,fitode-method**

*Extract model coefficients from fitode objects*

**Description**

Extracts estimated parameters (either on response scales or link scales)

**Usage**

```r
## S4 method for signature 'fitode'
coef(object, type = c("response", "links"))
```

**Arguments**

- `object`: fitode object
- `type`: type of coefficients. The default (`type=response`) is on the response scale; this is the scale on which the model parameters are defined. Alternatively, `type=link` can be used to obtain parameters on the estimated scale.

**Value**

The estimated coefficients of the fitode object

---

**coef,fitodeMCMC-method**

*Extract model coefficients from fitodeMCMC objects*

**Description**

Extracts estimated parameters (median of the marginal posterior distributions)

**Usage**

```r
## S4 method for signature 'fitodeMCMC'
coef(object)
```

**Arguments**

- `object`: fitodeMCMC object

**Value**

The estimated median coefficients of the fitodeMCMC object
Calculate confidence intervals from fitode objects for model parameters and their transformations

Description

Calculate confidence intervals for model parameters and their transformations using (1) delta method, (2) profile likelihood, and (3) importance sampling.

Usage

```r
## S4 method for signature 'fitode'
confint(  
  object,  
  parm,  
  level = 0.95,  
  method = c("delta", "profile", "impsamp", "wmvrnorm"),  
  nsim = 1000,  
  seed,  
  ...  
)
```

Arguments

- `object`: fitode object
- `parm`: character vector specifying model parameters or list of formulas specifying transformations
- `level`: the confidence level required
- `method`: method for calculating confidence intervals
- `nsim`: number of simulations to be used for importance sampling
- `seed`: seed
- `...`: extra arguments passed to profiling method

Value

The confidence intervals for model parameters and their transformations of the fitode object
confint, fitodeMCMC-method

Calculate credible intervals from fitodeMCMC objects for model parameters and their transformations

Description

Calculate credible intervals for model parameters and their transformations from posterior samples.

Usage

```r
## S4 method for signature 'fitodeMCMC'
confint(object, parm, level = 0.95)
```

Arguments

- `object`: fitodeMCMC object
- `parm`: character vector specifying model parameters or list of formulas specifying transformations
- `level`: the credible level required

Value

The credible intervals of the fitodeMCMC object

fitode

Fit ordinary differential equations model

Description

This function fits ordinary differential equations models to a uni- or multi-variate time series by maximum likelihood. It relies on sensitivity equations to compute gradients of the estimated trajectory with respect to model parameters. This allows one to use gradient-based optimization algorithms, which can provide more robust estimation.

Usage

```r
fitode(
    model,
    data,
    start,
    tcol = "times",
    method = "BFGS",
    optimizer = "optim",
    link,
```
fixed = list(),
prior = list(),
prior.density = TRUE,
control = list(maxit = 1e+05),
solver.opts = list(method = "rk4"),
solver = ode,
skip.hessian = FALSE,
force.hessian = FALSE,
use.ginv = TRUE,
quietly = FALSE,
...
)

Arguments

model     odemodel object
data       data frame with a time column and observation columns
start      named vector of starting parameter values
tcol       (character) time column
method     optimization method
optimizer  optimizer
link       named vector or list of link functions for model parameters
fixed      named vector or list of model parameters to fix and their values
prior      list of formulas specifying prior distributions
prior.density (logical) should priors represent probability distributions?
control    see optim
 solver.opts options for ode integration. See ode
 solver     ode solver
skip.hessian skip hessian calculation
force.hessian (logical) calculate the hessian numerically instead of taking the jacobian of the
              gradients based on sensitivity equations
use.ginv   (logical) use generalized inverse (ginv) to compute approximate vcov
quietly    suppress progress messages?
...
mle2 arguments

Value

An object of class “fitode” as described in fitode-class.

See Also

mle2
fitode-class

Class “fitode”. Result of ode fitting based on Maximum Likelihood Estimation

Description

Class “fitode”. Result of ode fitting based on Maximum Likelihood Estimation

Slots

- **call** (language) The call to `fitode`
- **model** odemodel object
- **data** the time series data
- **coef** estimated parameters
- **vcov** estimated variance-covariance matrix
- **min** minimum negative log-likelihood
- **mle2** mle2 object
- **link** list of link functions for model parameters
- **fixed** list of fixed parameters
- **prior** list of priors

See Also

- `mle2-class`

fitodeMCMC

Fit ordinary differential equations model using MCMC

Description

This function fits ordinary differential equations models to a uni- or multi-variate time series by MCMC using the Metropolis-Hastings update rule. It searches through the parameter space on link scales, which can provide more efficient posterior sampling.

Usage

```r
fitodeMCMC(
    model, 
    data, 
    start, 
    tcol = "times", 
    proposal.vcov, 
    prior = list(), 
)```
chains = 1,
iter = 2000,
burnin = iter/2,
thin = 1,
refresh = max(iter/10, 1),
prior.only = FALSE,
link,
fixed = list(),
solver.opts = list(method = "rk4"),
solver = ode,
...)

Arguments

model  
ode model

data  
data frame with time column and observation column

start  
named vector of starting parameter values

tcol  
time column

proposal.vcov  
variance-covariance matrix of a multivariate normal proposal distribution

prior  
list of formulas specifying prior distributions

chains  
(numeric) number of chains

iter  
(numeric) number of iterations per chain

burnin  
(numeric) number of burnin iterations

thin  
(numeric) thinning interval between consecutive observations

refresh  
(numeric) refresh interval

prior.only  
(logical) sample from prior distribution only?

link  
named vector or list of link functions for model parameters

fixed  
named vector or list of model parameters to fix and their values

solver.opts  
options for ode integration. See ode

solver  
ode solver

...  
additional arguments (unused)

Value

An object of class “fitodeMCMC” as described in fitodeMCMC-class.
fitodeMCMC-class

Class "fitodeMCMC". Result of ode fitting based on Markov Chain Monte Carlo (MCMC)

Description

Class "fitodeMCMC". Result of ode fitting based on Markov Chain Monte Carlo (MCMC)

Slots

call (language) The call to \texttt{fitodeMCMC}
model odemodel object
data the time series data
coeff estimated parameters (posterior median)
vcov estimated variance-covariance matrix
mcmc mcmc.list object containing posterior samples
lp mcmc.list object containing log-posterior values of posterior samples
link list of link functions for model parameters
fixed list of fixed parameters
prior list of priors
details a list containing miscellaneous objects for internal uses

initialize,odemodel-method

Constructor method of "odemodel" class

Description

Constructor method of "odemodel" class

Usage

```r
## S4 method for signature 'odemodel'
initialize(
  .Object,
  name,
  model,
  observation,
  initial,
  par,
  link,
  diffnames,
  keep_sensitivity = TRUE,
  call
)
```
Arguments

.Object object
name name of the model
model ode model
observation observation model
initial initial values
par model parameters
link link functions for parameters (log links are used as default)
diffnames optional character vector specifying the names of a variable for which the consecutive difference needs to be calculated
keep_sensitivity (logical) maintain the Jacobian as a part of the model object?
call original function call

Value

An object of class “odemodel” as described in odemodel-class.

Examples

SI_model <- odemodel(
  name = "SI",
  model = list(
    S ~ - beta*S*I/N,
    I ~ beta*S*I/N - gamma*I
  ),
  observation = list(
    susceptible ~ dnorm(mean=S, sd=sigma1),
    infected ~ dnorm(mean=I, sd=sigma2)
  ),
  initial = list(
    S ~ N * (1 - i0),
    I ~ N * i0
  ),
  par = c("beta", "gamma", "N", "i0", "sigma1", "sigma2"),
  link = c(i0="logit")
)

logLik, fitode-method

Extract log-likelihood

Description

Extract log-likelihood of a fit
**loglik.ode-class**

**Usage**

```r
## S4 method for signature 'fitode'
logLik(object)
```

**Arguments**

- `object` fitode object

**Value**

The log-likelihood of the fitode object

---

**loglik.ode-class**

*Class representing log-likelihood models used to fit ode models*

**Description**

Class representing log-likelihood models used to fit ode models

**Slots**

- `name` name of the distribution
- `expr` an expression specifying the model
- `observation` observation variable name
- `mean` mean variable name
- `par` additional parameter names
- `grad` the gradient with respect to the parameters

---

**plot,fitode,missing-method**

*Plot a fitode object*

**Description**

Plot a fitode object
Usage

```
## S4 method for signature 'fitode,missing'
plot(
  x,
  level,
  data,
  which,
  method = c("delta", "impsamp", "wmvrnorm"),
  onepage = TRUE,
  xlim,
  ylim,
  xlabs,
  ylabs,
  col.traj = "black",
  lty.traj = 1,
  col.conf = "black",
  lty.conf = 4,
  add = FALSE,
  nsim = 1000,
  ...
)
```

Arguments

- `x` fitode object
- `level` the confidence level required
- `data` (FIXME)
- `which` which to plot
- `method` confidence interval method
- `onepage` (logical) print all figures on one page?
- `xlim` x coordinates range
- `ylim` y coordinates range
- `xlabs` a label for the x axis
- `ylabs` a label for the y axis
- `col.traj` colour of the estimated trajectory
- `lty.traj` line type of the estimated trajectory
- `col.conf` colour of the confidence intervals
- `lty.conf` line type of the confidence intervals
- `add` add to another plot?
- `nsim` number of simulations for mvrnorm, wmvrnorm methods
- `...` additional arguments to be passed on to the plot function

Value

No return value, called for side effects
Description

Plot a fitodeMCMC object

Usage

```r
## S4 method for signature 'fitodeMCMC,missing'
plot(
  x,
  level,
  data,
  which,
  onepage = TRUE,
  xlim,
  ylim,
  xlabs,
  ylabs,
  col.traj = "black",
  lty.traj = 1,
  col.conf = "black",
  lty.conf = 4,
  add = FALSE,
  ...
)
```

Arguments

- `x` : fitodeMCMC object
- `level` : the confidence level required
- `data` : (FIXME)
- `which` : which to plot
- `onepage` : (logical) print all figures on one page?
- `xlim` : x coordinates range
- `ylim` : y coordinates range
- `xlabs` : a label for the x axis
- `ylabs` : a label for the y axis
- `col.traj` : colour of the estimated trajectory
- `lty.traj` : line type of the estimated trajectory
- `col.conf` : colour of the confidence intervals
plot_internal

lty.conf  line type of the confidence intervals
add  add to another plot?
...  additional arguments to be passed on to the plot function

Value

No return value, called for side effects

plot_internal  Internal function for plotting methods

Description

Internal function for plotting methods

Usage

plot_internal(
  pred,
  data,
  onepage = TRUE,
  xlim,
  ylim,
  xlabs,
  ylabs,
  col.traj = "black",
  lty.traj = 1,
  col.conf = "black",
  lty.conf = 4,
  add = FALSE,
...  
)

Arguments

pred  prediction objects
data  observed data
onepage  (logical) print all figures on one page?
xlim  x coordinates range
ylim  y coordinates range
xlabs  a label for the x axis
ylabs  a label for the y axis
col.traj  colour of the estimated trajectory
lty.traj  line type of the estimated trajectory
predict, fitode-method

Description

Computes estimated trajectories and their confidence intervals (using either the delta method or importance sampling).

Usage

```r
## S4 method for signature 'fitode'
predict(
  object,
  level,
  times,
  method = c("delta", "impsamp", "wmvrnorm"),
  nsim = 1000
)
```

Arguments

- `object`: fitode object
- `level`: the confidence level required
- `times`: time vector to predict over. Default is set to the time frame of the data.
- `method`: confidence interval method. Default is set to Delta method.
- `nsim`: number of simulations for mvrnorm, wmvrnorm methods

Value

The estimated trajectories and their confidence intervals of the fitode object
**predict,fitodeMCMC-method**

*Prediction function for fitodeMCMC objects*

**Description**
Computes estimated trajectories and their credible intervals. The estimated trajectories are obtained by taking the median trajectories from the posterior samples.

**Usage**
```r
## S4 method for signature 'fitodeMCMC'
predict(object, level, times, simplify = TRUE)
```

**Arguments**
- **object**: fitodeMCMC object
- **level**: the credible level required
- **times**: time vector to predict over. Default is set to the time frame of the data.
- **simplify**: (logical) simplify output to return estimated trajectories and their credible intervals? If simplify=FALSE, all posterior trajectories will be returned

**Value**
Estimated trajectories and their credible intervals of the fitodeMCMC object

---

**prior.ode-class**
*Class representing prior models used to fit ode models*

**Description**
Class representing prior models used to fit ode models

**Slots**
- **name**: name of the distribution
- **expr**: an expression specifying the model
- **observation**: observation variable name
- **par**: additional parameter names
- **keep_grad**: keep gradient?
- **grad**: the gradient with respect to the parameters
profile, fitode-method  Profile fitode objects

Description
Profile fitode objects

Usage
## S4 method for signature 'fitode'
profile(fitted, which = 1:p, alpha = 0.05, trace = FALSE, ...)

Arguments
- `fitted`: fitted model object
- `which`: which parameter(s) to profile? (integer value)
- `alpha`: critical level
- `trace`: trace progress of computations?
- `...`: additional arguments passed to mle2 profiling method

Value
The log-likelihood profile of the fitode object

SierraLeone2014  Data from 2014 Sierra Leone Ebola epidemic

Description
Ebola case reports ...

Usage
SierraLeone2014

Format
A data frame with 67 rows comprising:
- `times`: decimal dates (year + fraction of year)
- `confirmed`: confirmed cases
**simulate, fitode-method**

*simulate fitode objects*

**Description**

Simulate fitode objects.

**Usage**

```r
## S4 method for signature 'fitode'
simulate(object, nsim = 1, seed = NULL, times, parms, y, observation = TRUE)
```

**Arguments**

- `object`: fitode object
- `nsim`: number of simulations
- `seed`: random-number seed
- `times`: time vector
- `parms`: named vector of parameter values
- `y`: initial values
- `observation`: (logical) propagate observation error?

**Value**

The numerical simulation of the object.

**simulate, odemodel-method**

*simulate model objects*

**Description**

Simulate model objects.

**Usage**

```r
## S4 method for signature 'odemodel'
simulate(
  object,
  nsim = 1,
  seed = NULL,
  times,
  parms,
)
simulate_internal

y,
solver.opts = list(method = "rk4"),
solver = ode,
observation = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>odemodel object</td>
</tr>
<tr>
<td>nsim</td>
<td>number of simulations</td>
</tr>
<tr>
<td>seed</td>
<td>random-number seed</td>
</tr>
<tr>
<td>times</td>
<td>time vector</td>
</tr>
<tr>
<td>parms</td>
<td>named vector of parameter values</td>
</tr>
<tr>
<td>y</td>
<td>initial values</td>
</tr>
<tr>
<td>solver.opts</td>
<td>options for ode solver</td>
</tr>
<tr>
<td>solver</td>
<td>ode solver (must take y, times, func, and parms arguments)</td>
</tr>
<tr>
<td>observation</td>
<td>(logical) propagate observation error?</td>
</tr>
</tbody>
</table>

Value

The numerical simulation of the object

simulate_internal Internal function for simulation models

Description

Simulates deterministic trajectories and propagates observation error

Usage

simulate_internal(
  model,
  times,
  parms,
  y,
  solver.opts = list(method = "rk4"),
  solver = ode,
  observation = TRUE,
  nsim = 1,
  seed = NULL
)
**Arguments**

- **model**: odemodel object
- **times**: time vector
- **parms**: named vector of parameter values
- **y**: initial values
- **solver.opts**: options for ode solver
- **solver**: ode solver (must take y, times, func, and parms arguments)
- **observation**: (logical) propagate observation error?
- **nsim**: number of simulations
- **seed**: seed

---

**stdEr, fitode-method**

Extract standard error from fitode objects

**Description**

Calculates standard error by taking the square root of the diagonal matrix

**Usage**

```r
## S4 method for signature 'fitode'
stdEr(x, type = c("response", "links"))
```

**Arguments**

- **x**: fitode object
- **type**: type of standard error. The default (type=response) is on the response scale; this is the scale on which the model parameters are defined. Alternatively, type=link can be used to obtain standard errors on the estimated scale.

**Value**

The standard error of the fitode object
\textit{stdEr.fitodeMCMC-method} \hspace{1cm} \textit{Extract standard error from fitodeMCMC objects}

\section*{Description}

Calculates standard error by taking the square root of the diagonal of the variance-covariance matrix

\section*{Usage}

\begin{verbatim}
## S4 method for signature 'fitodeMCMC'
stdEr(x)
\end{verbatim}

\section*{Arguments}

- \textbf{x} \hspace{0.5cm} fitodeMCMC object

\section*{Value}

The standard error of the fitodeMCMC object

\section*{summary.fitode-method \hspace{1cm} Summarize fitode object}

\section*{Description}

Summarize fitode objects; returns estimate, standard error, and confidence intervals

\section*{Usage}

\begin{verbatim}
## S4 method for signature 'fitode'
summary(object)
\end{verbatim}

\section*{Arguments}

- \textbf{object} \hspace{0.5cm} fitode object

\section*{Value}

The summary of the fitode object
summary, fitodeMCMC-method

Summarize fitodeMCMC object

Description

Summarize fitodeMCMC object; returns estimate, standard error, credible intervals, effective sample sizes, and gelman-rubin diagnostic

Usage

## S4 method for signature 'fitodeMCMC'
summary(object)

Arguments

object fitodeMCMC object

Value

The summary of the fitodeMCMC object

See Also

effectiveSize, gelman.diag

tumorgrowth

Tumor growth data

Description

...

Usage

tumorgrowth

Format

A data frame containing 14 rows comprising:

day
volume
update,fitode-method  

Update fitode fits

Description
Update fitode fits

Usage
## S4 method for signature 'fitode'
update(object, observation, initial, par, link, ...)

Arguments
object  fitode objects
observation  observation model
initial  initial values
par  model parameters
link  link functions for parameters (log links are used as default)
...  additional arguments to be passed to fitode

Value
An object of class “fitode” as described in fitode-class.

update,fitodeMCMC-method  

Update fitodeMCMC fits

Description
Update fitodeMCMC fits

Usage
## S4 method for signature 'fitodeMCMC'
update(object, observation, initial, par, link, ...)

Arguments
object  fitodeMCMC objects
observation  observation model
initial  initial values
par  model parameters
link  link functions for parameters (log links are used as default)
...  additional arguments to be passed to fitode
vcov, fitodeMCMC-method

Value
An object of class “fitode” as described in fitodeMCMC-class.

vcov, fitode-method

Extract variance-covariance matrix from fitode objects

Description
Extracts variance-covariance matrix (either on response scales or link scales)

Usage
### S4 method for signature 'fitode'
vcov(object, type = c("response", "links"))

Arguments
- object: fitode object
- type: type of covariance matrix. The default (type=response) is on the response scale; this is the scale on which the model parameters are defined. Alternatively, type=link can be used to obtain the covariance matrix on the estimated scale.

Value
The variance-covariance matrix of the fitode object

vcov, fitodeMCMC-method

Extract variance-covariance matrix from fitodeMCMC objects

Description
Calculates variance-covariance matrix from posterior samples

Usage
### S4 method for signature 'fitodeMCMC'
vcov(object)

Arguments
- object: fitodeMCMC object

Value
The variance-covariance matrix of the fitodeMCMC object
Index

* datasets
  blowfly, 2
  SierraLeone2014, 17
  tumorgrowth, 22

  blowfly, 2
  coef, fitode-method, 3
  coef, fitodeMCMC-method, 3
  confint, fitode-method, 4
  confint, fitodeMCMC-method, 5

  effectiveSize, 22

  fitode, 5, 7
  fitode-class, 7
  fitodeMCMC, 7, 9
  fitodeMCMC-class, 9

  gelman.diag, 22
  ginv, 6

  initialize, odemodel-method, 9

  logLik, fitode-method, 10
  loglik.ode-class, 11

  mle2, 6

  ode, 6, 8
  optim, 6

  plot, fitode, missing-method, 11
  plot, fitode-method
    (plot, fitode, missing-method), 11
  plot, fitodeMCMC, missing-method, 13
  plot, fitodeMCMC-method
    (plot, fitodeMCMC, missing-method), 13
  plot_internal, 14

  predict, fitode-method, 15
  predict, fitodeMCMC-method, 16
  prior.ode-class, 16
  profile, fitode-method, 17

  SierraLeone2014, 17
  simulate, fitode-method, 18
  simulate, odemodel-method, 18
  simulate_internal, 19
  stdEr, fitode-method, 20
  stdEr, fitodeMCMC-method, 21
  summary, fitode-method, 21
  summary, fitodeMCMC-method, 22

  tumorgrowth, 22

  update, fitode-method, 23
  update, fitodeMCMC-method, 23

  vcov, fitode-method, 24
  vcov, fitodeMCMC-method, 24