Package ‘EKMCPP’

October 12, 2022

Type Package
Title MCMC Procedures for Estimating Enzyme Kinetics Constants
Version 1.1.2
Author Hyukpyo Hong, Boseung Choi, Jae Kyoung Kim
Maintainer Hyukpyo Hong <hphong@kaist.ac.kr>
Description Functions for estimating catalytic constant and Michaelis-Menten constant for enzyme kinetics model using Metropolis-Hasting algorithm within Gibbs sampler based on the Bayesian framework.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports numDeriv
Depends R (>= 3.5.0)
Suggests rmarkdown, knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2021-08-20 06:20:02 UTC

R topics documented:
catalytic_est .................................................... 2
main_est ....................................................... 3
MM_catal_est ................................................... 5
MM_est ........................................................ 7
timeseries_data_example ................................... 9
Index 10

1
catalytic_est  

Function for estimating the catalytic constant

Description

The function estimates catalytic constant using progress-curve data, enzyme concentrations, substrate concentrations, and the Michaelis-Meten constant.

Usage

catalytic_est(
    method,       
    timespan,     
    products,     
    enz,          
    subs,         
    K_M,          
    catal_m,      
    catal_v,      
    nrepeat,      
    jump,         
    burn,         
    volume,       
    t_unit,       
    c_unit
)

Arguments

method  
This determines which model, the sQSSA or tQSSA model, is used for the estimation. Specifically, the input for method is TRUE (FALSE); then the tQSSA (sQSSA) model is used.

timespan  
time points when the concentrations of products were measured.

products  
measured concentrations of products.

enz  
initial enzyme concentrations.

subs  
initial substrate concentrations.

K_M  
true value of the Michaelis-Menten constant.

catal_m  
prior mean of gamma prior for the catalytic constant k_cat.

catal_v  
prior variance of gamma prior for the catalytic constant k_cat.

nrepeat  
number of effective iteration, i.e., posterior samples.

jump  
length of distance between sampling, i.e., thinning rate.

burn  
length of burn-in period.

volume  
the volume of a system. It is used to scale the product concentration. FALSE input provides automatic scaling.
The function `catalytic_est` generates a set of Monte Carlo simulation samples from posterior distribution of the catalytic constant of enzyme kinetics model. Because the function estimates only the catalytic constant, the true value of the Michaelis-Menten constant should be given. Authors' recommendation: "Do not use this function directly. Do use the function `main_est()` to estimate the parameter so that the main function calls this function".

### Value
A vector containing posterior samples of the estimated parameter: the catalytic constant.

### Examples
```r
## Not run:
data("timeseries_data_example")
timespan1=timeseries_data_example[,c(1,3,5,7)]
products1=timeseries_data_example[,c(2,4,6,8)]
catalytic_result <- catalytic_est(method=TRUE,timespan=timespan1,
products=products1,enz = c(4.4, 4.4, 440, 440), subs=c(4.4, 4.4, 4.4, 4.4),
K_M=44, catal_m = 1, catal_v = 1000, jump = 10, burn = 1000, nrepeat = 1000,
volume = FALSE, t_unit = "sec", c_unit = "mM")
## End(Not run)
```

---

main_est

The function estimates either the catalytic constant, the Michaelis-Menten constant, or both simultaneously using progress-curve data, initial enzyme concentrations, and initial substrate concentrations.

### Usage
```r
main_est(
  method = TRUE,
  timeseries,
  enz,
  subs,
  K_M = FALSE,
  catal = FALSE,
  K_M_init = FALSE,
)```

---

### Description
Main function for estimating catalytic constant `k_cat` and Michaelis-Menten (MM) constant `K_M`
std = FALSE,
tun = 2.4,
nrepeat = 1000,
jump = 10,
burn = 1000,
catal_m = 1,
catal_v = 1e+06,
K_M_m = FALSE,
K_M_v = FALSE,
volume = FALSE,
t_unit,
c_unit
)

Arguments

method
This determines which model, the sQSSA or tQSSA model, is used for the estimation. Specifically, the input for method is TRUE (FALSE); then the tQSSA (sQSSA) model is used. Its default value is TRUE.

timeseries
Data frame containing the time points and measured concentrations of products. Every two columns represent the time points when the concentrations of the products were measured and the corresponding measured concentrations.

enz
initial enzyme concentrations

subs
initial substrate concentrations

K_M
true value of the Michaelis-Menten constant. Specify this object if the true value is known. Its default value is FALSE.

catal
true value of the catalytic constant. Specify this object if the true value is known. Its default value is FALSE.

K_M_init
initial value of K_M constant for the Metropolis-Hastings algorithm. If the input is FALSE then it is determined by max(subs). Its default value is FALSE.

std
standard deviation of proposal distribution. If the input is FALSE then it is determined by using the hessian of log posterior distribution. Its default value is FALSE.

tun
tuning constant for the Metropolis-Hastings algorithm when std is FALSE (i.e., hessian of the log posterior distribution is used). Its default value is 2.4.

nrepeat
number of effective iteration, i.e., posterior samples. Its default value is 1,000.

jump
length of distance between sampling, i.e., thinning rate. Its default value is 10.

burn
length of burn-in period. Its default value is 1,000.

catal_m
prior mean of gamma prior for the catalytic constant k_cat. Its default value is 1.

catal_v
prior variance of gamma prior for the catalytic constant k_cat. Its default value is 1e+06.

K_M_m
prior mean of gamma prior for the Michaelis-Menten constant K_M. If the input is FALSE then it is determined by max(subs). Its default value is FALSE.
**K_M_v**

prior variance of gamma prior for the Michaelis-Menten constant K_M. If the input is FALSE then it is determined by max(subs)^2*1000. Its default value is FALSE.

**volume**

the volume of a system. It is used to scale the product concentration. FALSE input provides automatic scaling. Its default value is FALSE.

**t_unit**

the unit of time points. It can be an arbitrary string.

**c_unit**

the unit of concentrations. It can be an arbitrary string.

### Details

The function main_est generates a set of Markov Chain Monte Carlo (MCMC) simulation samples from the posterior distribution of the catalytic constant or (and) the Michaelis-Menten constant of enzyme kinetics model. Users should input initial enzyme concentrations, substrate concentrations, and progress-curve data. Prior information for both parameters can be given. The Gibbs sampling and Metropolis Hastings algorithms are used to sample the parameters. Parameters for the MCMC such as tuning parameter for proposal distribution, prior parameters, and the iteration number can be specified by users. This function use one of catalytic_est(), MM_est(), MM_catal_est() to generate the samples depending on parameter(s) to be estimated.

### Value

A vector (or matrix) containing posterior samples of the estimated parameter(s).

### Examples

```r
## Not run:
data("timeseries_data_example")
result <- main_est(method=TRUE, timeseries = timeseries_data_example,
enz = c(4.4, 4.4, 440, 440), subs=c(4.4, 4.4, 4.4, 4.4), K_M_init = 1e+1,
std=1e+1, tun = 3.5, jump=10, burn=1000, nrepeat=1000,
catal_m=1, catal_v=100, K_M_m=1, K_M_v=1e+4, volume = FALSE,
t_unit = "sec", c_unit = "mM")
## End(Not run)
```

---

**MM_catal_est**

Function for estimating both of the Michaelis-Menten constant and catalytic constant simultaneously

### Description

The function estimates both of the catalytic and the Michaelis-Meten constants simultaneously using progress-curve data, enzyme concentrations, and substrate concentrations.
Usage

```r
MM_catal_est(
  method,
  timespan,
  products,
  enz,
  subs,
  K_M_init,
  std,
  tun,
  nrepeat,
  jump,
  burn,
  catal_m,
  catal_v,
  K_M_m,
  K_M_v,
  volume,
  t_unit,
  c_unit
)
```

Arguments

- **method**: This determines which model, the sQSSA or tQSSA model, is used for the estimation. Specifically, the input for method is TRUE (FALSE); then the tQSSA (sQSSA) model is used.
- **timespan**: time points when the concentrations of products were measured.
- **products**: measured concentrations of products.
- **enz**: initial enzyme concentrations.
- **subs**: initial substrate concentrations.
- **K_M_init**: initial value of K_M constant for the Metropolis-Hastings algorithm. If the input is FALSE then it is determined by max(subs).
- **std**: standard deviation of proposal distribution. If the input is FALSE then it is determined by using the hessian of log posterior distribution.
- **tun**: tuning constant for the Metropolis-Hastings algorithm when std is FALSE (i.e., hessian of the log posterior distribution is used).
- **nrepeat**: number of effective iteration, i.e., posterior samples.
- **jump**: length of distance between sampling, i.e., thinning rate.
- **burn**: length of burn-in period.
- **catal_m**: prior mean of gamma prior for the catalytic constant k_cat.
- **catal_v**: prior variance of gamma prior for the catalytic constant k_cat.
- **K_M_m**: prior mean of gamma prior for the Michaelis-Menten constant K_M. If the input is FALSE then it is determined by max(subs).
**Details**

The function MM_catal_est generates a set of Markov Chain Monte Carlo simulation samples from the posterior distribution of $K_M$ and catalytic constant of enzyme kinetics model. Authors' recommendation: "Do not use this function directly. Do use the function main_est() to estimate the parameters so that the main function calls this function"

**Value**

A matrix containing posterior samples of the estimated parameters: the catalytic constant and the Michaelis-Menten constant.

**Examples**

```r
## Not run:
data("timeseries_data_example")
timespan1=timeseries_data_example[,c(1,3,5,7)]
products1=timeseries_data_example[,c(2,4,6,8)]
MM_catal_result <- MM_catal_est(method=TRUE, timespan=timespan1,
products=products1, enz = c(4.4, 4.4, 440, 440), subs=c(4.4, 4.4, 4.4, 4.4),
$K_M$ $m$ $= 1$, catal_m=1, catal_v = 1000, $K_M_m$ = 1, $K_M_v$ = 100000,
std = 10, tun = 3.5, nrepeat = 1000, jump = 10, burn = 1000,
volume = FALSE, t_unit = "sec", c_unit = "mM")

## End(Not run)
```

---

**MM_est**

*Function for estimating the Michaelis-Menten constant*

**Description**

The function estimates the Michaelis-Menten constant using progress-curve data, enzyme concentrations, substrate concentrations, and the catalytic constant.

**Usage**

```r
MM_est(
  method, 
  timespan, 
  products, 
  enz, 
)```
substituting, catal, K_M_init, std, tun, nrepeat, jump, burn, K_M_m, K_M_v, volume, t_unit, c_unit)

Arguments

method  This determines which model, the sQSSA or tQSSA model, is used for the estimation. Specifically, the input for method is TRUE (FALSE); then the tQSSA (sQSSA) model is used.
timespan  time points when the concentrations of products were measured.
products  measured concentrations of products
enz  initial enzyme concentrations
subs  initial substrate concentrations
catal  true value of the catalytic constant.
K_M_init  initial value of K_M constant for the Metropolis-Hastings algorithm. If the input is FALSE then it is determined by max(subs).
std  standard deviation of proposal distribution. If the input is FALSE then it is determined by using the hessian of log posterior distribution.
tun  tuning constant for the Metropolis-Hastings algorithm when std is FALSE (i.e., hessian of the log posterior distribution is used).
nrepeat  number of effective iteration, i.e., posterior samples.
jump  length of distance between sampling, i.e., thinning rate.
burn  length of burn-in period.
K_M_m  prior mean of gamma prior for the Michaelis-Menten constant K_M. If the input is FALSE then it is determined by max(subs).
K_M_v  prior variance of gamma prior for the Michaelis-Menten constant K_M. If the input is FALSE then it is determined by max(subs)^2*1000.
volume  the volume of a system. It is used to scale the product concentration. FALSE input provides automatic scaling.
t_unit  the unit of time points. It can be an arbitrary string.
c_unit  the unit of concentrations. It can be an arbitrary string.
Details

The function MM_est generates a set of Markov Chain Monte Carlo simulation samples from posterior distribution of the Michaelis-Menten constant of enzyme kinetics model. Because the function estimates only the Michaelis-Menten constant the true value of the catalytic constant should be given. Authors' recommendation: "Do not use this function directly. Do use the function main_est() to estimate the parameter so that the main function calls this function"

Value

A vector containing posterior samples of the estimated parameter: the Michaelis-Menten constant.

Examples

```r
## Not run:
data("timeseries_data_example")
timespan1=timeseries_data_example[,c(1,3,5,7)]
products1=timeseries_data_example[,c(2,4,6,8)]
MM_result <- MM_est(method=TRUE	imespan=timespan1,products=products1,
enz = c(4.4, 4.4, 440, 440), subs=c(4.4, 4.4, 4.4, 4.4), catal = 0.051,
K_M_init = 1, K_M_m = 1, K_M_v = 100000, std = 10, tun =3.5,
nrepeat = 1000, jump = 10, burn = 1000, volume = FALSE,
t_unit = "sec", c_unit = "mM")
## End(Not run)
```

Description

An artificial data set containing the product concentration observed with the high and low enzyme concentrations. The 1st, 3rd, 5th, and 7th columns are observed times, and the 2nd, 4th, 6th, and 8th columns are product concentrations. The 2nd and 4th columns are observed with the initial enzyme concentrations of 4.4, and the 6th and 8th columns are observed with the initial enzyme concentrations of 440. The initial substrate concentrations are 4.4 for all data.

Usage

`timeseries_data_example`

Format

A data frame with 101 rows and 8 variables:

- **V1 V3 V5 V7** observed times, no unit
- **V2 V4 V6 V8** product concentrations, no unit
Index

* datasets
  timeseries_data_example, 9

catalytic_est, 2

main_est, 3
MM_catal_est, 5
MM_est, 7

timeseries_data_example, 9