Package 'bioOED'

March 13, 2017

Type Package

Title Sensitivity Analysis and Optimum Experiment Design for Microbial Inactivation

Version 0.1.1

Date 2017-03-12

Description Extends the bioinactivation package with functions for Sensitivity Analysis and Optimum Experiment Design.

License GPL-3

LazyData TRUE

Imports FME (>= 1.3.2), bioinactivation (>= 1.1.2), corrplot(>= 0.73), dplyr (>= 0.5.0), ggplot2 (>= 2.1.0), MEIGOR (>= 1.0.0), stats (>= 3.3.2), graphics(>= 3.3.2)

Suggests knitr (>= 1.9), testthat (>= 0.9.1), rmarkdown

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Alberto Garre [aut, cre],
Jose Lucas Penalver [aut],
Pablo S. Fernandez [aut],
Jose A. Egea [aut]

Maintainer Alberto Garre <garre.alberto@gmail.com>

Repository CRAN

Date/Publication 2017-03-13 08:45:26

R topics documented:

`calculate_FIM` *Calculation of Fisher Information Matrix*

Description

The sensitivities at the different `times` are calculated by linear interpolation of the results provided in `sensitivities`.

Usage

```
calculate_FIM(sensitivities, times)
```

Arguments

`sensitivities`	data.frame of class `sensFun` as returned by `sensitivity_inactivation`.
`times`	Numeric vector of time points where observations will be taken.

Value

Matrix with the estimation of the Fisher Information Matrix.

calculate_pars_correlation *Correlation Between Model Parameters Sensitivities*

Description

Correlation Between Model Parameters Sensitivities

Usage

```
calculate_pars_correlation(inactivation_model, parms, temp_profile, parms_fix,
  n_times = 100, sensvar = "logN")
```

Arguments

inactivation_model	Character defining the inactivation model to use.
parms	Numeric vector with the nominal values of the model parameters.
temp_profile	Data frame describing the environmental conditions.
parms_fix	Nominal value of the parameters not considered for the sensitivity.
n_times	Numeric value specifying the nombers of time points where the sensitivity functions will be calculated. 100 by default.
sensvar	The output variable for which the sensitivity will be estimated. "logN" by default.

Examples

```
parms_fix <- c(temp_ref = 57.5)
parms <- c(delta_ref = 3.9, z = 4.2, p = 1, N0 = 1e6)
temp_profile <- data.frame(time = c(0, 60), temperature = c(30, 60))
correlations <- calculate_pars_correlation("Mafart", parms,
                                           temp_profile, parms_fix)
plot(correlations)
```

criterium_D *D Optimality Criterium*

Description

D Optimality Criterium

Usage

```
criterium_D(FIM)
```

Arguments

FIM	Matrix with the values of the Fisher Information Matrix

---

criterium_modE *Modified-E Optimality Criterium*

Description

Modified-E Optimality Criterium

Usage

```
criterium_modE(FIM, eig_tol = 1e-10)
```

Arguments

FIM	Matrix with the values of the Fisher Information Matrix
eig_tol	Tolerance for the eigen values. If any eigen value is lower than this value, the FIM is singular and a high value (1e20) is returned. 1e-10 by default.

---

inactivation_OED *Optimum Experimental Design of Microbial Inactivation*

Description

Performs an optimum experimental design for the settings selected. The OED is based on the FIM, estimated using the local sensitivity functions provided by sensitivity_inactivation.

Usage

```
inactivation_OED(inactivation_model, parms, temp_profile, parms_fix, n_points,
  criteria = "D", n_times = 100, sensvar = "logN",
  optim_algorithm = "global", opts_global = NULL)
```

Arguments

inactivation_model	Character string defining the inacivation model.
parms	Named numeric vector defining the model parameters. They must be named according to the needs of predict_inactivation.
temp_profile	Data frame defining the temperature profile. It must contain a column named time and a column named temperature.
parms_fix	Named numeric vector defining the model parameters to be omitted during the calculation of the local sensitivities.

n_points	Number of measurements which will be taken during the experiment.
criteria	Character defining the criteria for the OED. Either D (default) or E-mod.
n_times	Integer defining th enumber of discrete time points used for the interpolation of the local sensitivities.
sensvar	Character defining the variable to use for the OED. Either logN (default) or N.
optim_algorithm	Character defining the type of algorithm to use for the optimization. Either global (default) or local.
opts_global	List defining the options for the global optimization algorithm (see MEIGO). By default, global solver with a maximum of 50000 function evaluations and printout on every step.

Value

A list of class OEDinactivation with the following items:

- optim: Objetc returned by the optimization function.
- model: Inactivation model used for the calculations.
- parms: Nominal model parameters.
- parms_fix: Model parameters not considered for the sensitivity calculation.
- criteria: Criteria used for the OED.
- sensvar: Variable used for the OED.
- optim_algorithm: Type of optimization algorithm.
- optim_times: Optimum measurement times calculated.
- penalty: Logical indicating whether penalty function was used.
- temp_profile: Temperature profile of the experiment.

Examples

```
## Definition of input variables

parms_fix <- c(temp_ref = 57.5)
parms <- c(delta_ref = 3.9,
           z = 4.2,
           p = 1,
           N0 = 1e6
           )

temp_profile <- data.frame(time = c(0, 60), temperature = c(30, 60))

n_points <- 5

## OED with local optimization

set.seed(191210)

local_OED <- inactivation_OED("Mafart", parms, temp_profile, parms_fix,
```

```
                      n_points, criteria = "E-mod", sensvar = "logN",
                      optim_algorithm = "local")

print(local_OED$optim_times)
plot(local_OED)
```

`inactivation_OED_penalty` *Optimum Experimental Design of Microbial Inactivation with Penalty*

Description

Performs an optimum experimental design for the settings selected including a function which penalties points too close. The OED is based on the FIM, estimated using the local sensitivity functions provided by `sensitivity_inactivation`.

Usage

```
inactivation_OED_penalty(inactivation_model, parms, temp_profile, parms_fix,
  n_points, time_min, criteria = "D", n_times = 100, sensvar = "logN",
  optim_algorithm = "global", opts_global = NULL)
```

Arguments

`inactivation_model`	Character string defining the inacivation model.
`parms`	Named numeric vector defining the model parameters. They must be named according to the needs of `predict_inactivation`.
`temp_profile`	Data frame defining the temperature profile. It must contain a column named `time` and a column named `temperature`.
`parms_fix`	Named numeric vector defining the model parameters to be omitted during the calculation of the local sensitivities.
`n_points`	Number of measurements which will be taken during the experiment.
`time_min`	Numeric value indicating the minimum space between measurements.
`criteria`	Character defining the criteria for the OED. Either `D` (default) or `E-mod`.
`n_times`	Integer defining th enumber of discrete time points used for the interpolation of the local sensitivities.
`sensvar`	Character defining the variable to use for the OED. Either `logN` (default) or `N`.
`optim_algorithm`	Character defining the type of algorithm to use for the optimization. Either `global` (default) or `local`.
`opts_global`	List defining the options for the global optimization algorithm (see `MEIGO`). By default, global solver with a maximum of 50000 function evaluations and print-out on every step.

Value

A list of class `OEDinactivation` with the following items:

- optim: Objetc returned by the optimization function.
- model: Inactivation model used for the calculations.
- parms: Nominal model parameters.
- parms_fix: Model parameters not considered for the sensitivity calculation.
- criteria: Criteria used for the OED.
- sensvar: Variable used for the OED.
- optim_algorithm: Type of optimization algorithm.
- optim_times: Optimum measurement times calculated.
- penalty: Logical indicating whether penalty function was used.
- temp_profile: Temperature profile of the experiment.

Examples

```
## Definition of input variables

parms_fix <- c(temp_ref = 57.5)
parms <- c(delta_ref = 3.9,
           z = 4.2,
           p = 1,
           N0 = 1e6
)

temp_profile <- data.frame(time = c(0, 60), temperature = c(30, 60))

n_points <- 5
time_min <- 10

## OED with local optimization

set.seed(0123182)

local_OED <- inactivation_OED_penalty("Mafart", parms, temp_profile, parms_fix,
                              n_points, criteria = "E-mod", sensvar = "logN",
                              optim_algorithm = "local", time_min = time_min)

print(local_OED$optim_times)
plot(local_OED)

## OED with global optimization

opts_global <- list(maxeval=1000,  local_solver=0,
                    local_finish="DHC", local_iterprint=1)

global_OED <- inactivation_OED_penalty("Mafart", parms, temp_profile, parms_fix,
                               n_points, criteria = "E-mod", opts_global = opts_global,
```

```
                                time_min = time_min)

print(global_OED$optim_times)
plot(global_OED)
```

`inactivation_sens_handler` *Handler for the calculation of sensitivities of inactivation models*

Description

Handler for the calculation of sensitivities of inactivation models

Usage

```
inactivation_sens_handler(model_parms, inactivation_model, times, temp_profile,
  parms_fix)
```

Arguments

`model_parms`	A named vector or list with the values of the model parameters. See the documentation of bioinactivation::predict_inactivation.
`inactivation_model`	A character defining the inactivation model to use. See the documentation of bioinactivation::predict_inactivation.
`times`	A numeric vector describing the points where the solution will be calculated. See the documentation of bioinactivation::predict_inactivation.
`temp_profile`	A data frame describing the temperature profile. See the documentation of bioinactivation::predict_inactivation.
`parms_fix`	A named vector or list with the values of the known model parameters. See the documentation of bioinactivation::predict_inactivation.

`objective_D` *Objective Function for the D Criterium*

Description

Objective Function for the D Criterium

Usage

```
objective_D(times, sensitivities)
```

Arguments

times	A numeric vector of points where the FIM will be calculated.
sensitivities	An object returned by sensitivity_inactivation.

---

objective_D_penalty *Objective Function for the D Criterium with Penalty*

Description

Objective Function for the D Criterium with Penalty

Usage

```
objective_D_penalty(times, sensitivities, time_min)
```

Arguments

times	Numeric vector of points where the FIM is calculated.
sensitivities	An object returned by sensitivity_inactivation.
time_min	Numeric defining the minimum time between measurements.

---

objective_Emod *Objective Function for the modified-E Criterium*

Description

Objective Function for the modified-E Criterium

Usage

```
objective_Emod(times, sensitivities)
```

Arguments

times	A numeric vector of points where the FIM will be calculated.
sensitivities	An object returned by sensitivity_inactivation.

objective_Emod_penalty *Objective Function for the modified-E Criterium with Penalty*

Description

Objective Function for the modified-E Criterium with Penalty

Usage

```
objective_Emod_penalty(times, sensitivities, time_min)
```

Arguments

times	Numeric vector of points where the FIM is calculated.
sensitivities	An object returned by sensitivity_inactivation.
time_min	Numeric defining the minimum time between measurements.

optimize_refTemp *Optimization of the Reference Temperature*

Description

Finds the optimum value of the reference temperature which minimizes the correlation between sensitivty functions of the model parameters.

Usage

```
optimize_refTemp(temp_ref0, lower, upper, inactivation_model, parms,
  temp_profile, parms_fix, n_times = 100)
```

Arguments

temp_ref0	Initial value of the reference temperature to use for the optimization.
lower	Lower bound for the reference temperature.
upper	Upper bound for the reference temperature.
inactivation_model	Character identifying the inactivation model to use for the calculation.
parms	Numeric vector with the nominal values of the model parameters.
temp_profile	Data frame describing the environmental conditions.
parms_fix	Nominal value of the parameters not considered for the sensitivity.
n_times	Numeric value specifying the nombers of time points where the sensitivity functions will be calculated. 100 by default.

Details

The optimization is made using the optim function. The target for the optimization is the maximization of the determinant of the correlation matrix between parameter sensitivities. The Brent method is used, as it is the recommended one for unidimensional optimization. The parameters z and D/delta cannot be fixed.

Value

The object returned by optim.

Examples

```
parms <- c(delta_ref = 3.9, z = 4.2, p = 1, N0 = 1e6)
temp_profile <- data.frame(time = c(0, 60), temperature = c(30, 60))

optimize_refTemp(57, 50, 70, "Mafart", parms, temp_profile, c())
```

---

penalty_function *Penalty Function for OED*

Description

Penalty Function for OED

Usage

```
penalty_function(time_points, time_min, a = 1e+15, b = 2e+15)
```

Arguments

time_points	Numeric vector of time points for the measurements.
time_min	Numeric defining the minimum time between measurements.
a	Numeric defining the shape of the penalty function. 1e15 by default.
b	Numeric defining the shape of the penalty function. 2e15 by default.

plot.OEDinactivation *Plot of OEDinactivation*

Description

Plot of OEDinactivation

Usage

```
## S3 method for class 'OEDinactivation'
plot(x, y = NULL, ...)
```

Arguments

x	An instance of OEDinactivation
y	Ignored
...	Ignored

plot.parCorrelation *Correlation Plot of Parameter Sensitivities*

Description

Makes a correlation plot of the sensitivities between model parameters.

Usage

```
## S3 method for class 'parCorrelation'
plot(x, y = NULL, ...)
```

Arguments

x	Instance of parCorrelation
y	Ignored
...	Ignored

refTemp_optim_handler *Hanlder for the Optimization of Reference Temperature*

Description

Hanlder for the Optimization of Reference Temperature

Usage

```
refTemp_optim_handler(temp_ref, inactivation_model, parms, temp_profile,
  parms_fix, n_times, temp_ref0)
```

Arguments

temp_ref	New value of the reference temperature.
inactivation_model	Character identifying the inactivation model to use for the calculation.
parms	Numeric vector with the nominal values of the model parameters.
temp_profile	Data frame describing the environmental conditions.
parms_fix	Nominal value of the parameters not considered for the sensitivity.
n_times	Numeric value specifying the nombers of time points where the sensitivity functions will be calculated. 100 by default.
temp_ref0	Initial value of the reference temperature.

sensitivity_inactivation *Local sensitivities of microbial inactivation*

Description

Calculates the local sensitivity function of a microbial inactivation process. These are estimated using finite differences, through the function sensFun from the FME package.

Usage

```
sensitivity_inactivation(inactivation_model, parms, temp_profile, parms_fix,
  n_times = 100, varscale = 1, parscale = 1, sensvar = "logN", ...)
```

Arguments

inactivation_model	Character defining the inactivation model to use.
parms	Numeric vector with the nominal values of the model parameters.
temp_profile	Data frame describing the environmental conditions.
parms_fix	Nominal value of the parameters not considered for the sensitivity.
n_times	Numeric value specifying the nombers of time points where the sensitivity functions will be calculated. 100 by default.
varscale	The scaling factor for sensitivity variables. NULL indicates that the variable value is used. 1 by default.
parscale	The scaling factor for parameters. NULL indicates that the parameter value is used. 1 by default.
sensvar	The output variable for which the sensitivity will be estimated. "logN" by default.
...	Additional arguments passed to sensFun

Value

A data.frame of class sensFun.

See Also

sensFun

Examples

```
parms_fix <- c(temp_ref = 57.5)
parms <- c(delta_ref = 3.9,
           z = 4.2,
           p = 1,
           N0 = 1e6
)

temp_profile <- data.frame(time = c(0, 60), temperature = c(30, 60)
)

sensitivity <- sensitivity_inactivation("Mafart", parms,
                                temp_profile, parms_fix)

plot(sensitivity)
```

Index