# Package ‘bioOED’

October 12, 2022

**Type** Package

**Title** Sensitivity Analysis and Optimum Experiment Design for Microbial Inactivation

**Version** 0.2.1

**Date** 2019-08-07

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

**License** GPL-3

**LazyData** TRUE

**biocViews**

**Imports** FME (>= 1.3.2), bioinactivation (>= 1.1.2), corrplot(>= 0.73), dplyr (>= 0.5.0), ggplot2 (>= 2.1.0), stats (>= 3.3.2), graphics(>= 3.3.2), MEIGOR (>= 1.0.0), rlang (>= 0.1.6), tidyr (>= 0.7.2), tidyverse(>= 1.2.1), tidyselect(>= 0.2.5)

**Suggests** knitr (>= 1.9), testthat (>= 0.9.1), rmarkdown (>= 1.12)

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**NeedsCompilation** no

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

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

**Repository** CRAN

**Date/Publication** 2019-08-07 15:00:02 UTC

## 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_isothermal_FIM` *Fisher Information Matrix for isothermal experiments*

---

### Description

Fisher Information Matrix for isothermal experiments

### Usage

```
calculate_isothermal_FIM(model, exp_design, pars)
```

### Arguments

| | |
|---|---|
| `model` | character defining the inactivation model according to the rules in the `bioinactivation` package. |
| `exp_design` | data.frame with two columns named `times` and `temperature` describing the experiment design. |
| `pars` | list defining the model parameters according to the rules defined in the `bioinactivation` package. |

### Examples

```
library("dplyr")
time_profile <- seq(0, 50, length = 20)
Temp_profile <- seq(52.5,60, length = 3)

exp_design <- expand.grid(time_profile,Temp_profile) %>%
  rename(times = Var1, temperature = Var2)

pars <- list(temp_crit = 55,
             n = 1.5,
             k_b = 0.1)

calculate_isothermal_FIM("Peleg", exp_design, pars )
```

## calculate_limit *"Detection" limit for each model*

### Description

Calculation of the detection limit depending on the model.

### Usage

```
calculate_limit(model, pars, limit, temp_range)
```

### Arguments

| | |
|---|---|
| `model` | character string defining the inactivation model to use. |
| `pars` | list defining the model parameters according to the rules defined in the bioinactivation package. |
| `limit` | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |
| `temp_range` | Numeric vector that defines the range of possible temperatures |

### Value

Numerical value that indicates the limit of detection

## 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_Amod_iso` *Objective function for A modified-optimal OED with detection limit*

### Description

Points outside of the allowable area are moved back in time to the detection limit

### Usage

```
criterium_Amod_iso(x, model, pars, limit)
```

### Arguments

`x` a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points.

`model` character string defining the inactivation model to use.

`pars` list defining the model parameters according to the rules defined in the bioinactivation package.

`limit` numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit.

### Value

Numeric value of the objective function for criterium A modified, which is a determinant of the FIM.

## `criterium_A_iso` *Objective function for A-optimal OED with detection limit*

### Description

Points outside of the allowable area are moved back in time to the detection limit

### Usage

```
criterium_A_iso(x, model, pars, limit)
```

### Arguments

| | |
|---|---|
| `x` | a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points. |
| `model` | character string defining the inactivation model to use. |
| `pars` | list defining the model parameters according to the rules defined in the bioinactivation package. |
| `limit` | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |

### Value

Numeric value of the objective function for criterium A, which is a determinant of the FIM.

## `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_D_iso` *Objective function for D-optimal OED with detection limit*

### Description

Points outside of the allowable area are moved back in time to the detection limit

### Usage

```
criterium_D_iso(x, model, pars, limit)
```

### Arguments

`x` a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points.

`model` character string defining the inactivation model to use.

`pars` list defining the model parameters according to the rules defined in the bioinactivation package.

`limit` numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit.

### Value

Numeric value of the objective function for criterium D, which is a determinant of the FIM.

## `criterium_Emod_iso` *Objective function for E modified-optimal OED with detection limit*

### Description

Points outside of the allowable area are moved back in time to the detection limit

### Usage

```
criterium_Emod_iso(x, model, pars, limit)
```

### Arguments

`x` a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points.

`model` character string defining the inactivation model to use.

`pars` list defining the model parameters according to the rules defined in the bioinactivation package.

`limit` numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit.

### Value

Numeric value of the objective function for criterium E modified, which is a determinant of the FIM.

---

## criterium_E_iso *Objective function for E-optimal OED with detection limit*

---

### Description

Points outside of the allowable area are moved back in time to the detection limit

### Usage

```
criterium_E_iso(x, model, pars, limit)
```

### Arguments

| | |
|---|---|
| `x` | a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points. |
| `model` | character string defining the inactivation model to use. |
| `pars` | list defining the model parameters according to the rules defined in the bioinactivation package. |
| `limit` | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |

### Value

Numeric value of the objective function for criterium E, which is a determinant of the FIM.

---

## 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. |

## detection_bigelow *Detection limit of the Bigelow model*

### Description

Calculation of the detection limit for the Bigelow model

### Usage

```
detection_bigelow(pars, temperature, limit)
```

### Arguments

`pars` list defining the model parameters according to the rules defined in the bioinactivation package.

`temperature` numerical value that describes the temperature at which the detection limit will be calculated

`limit` numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit.

### Value

Numerical value that indicates the limit of detection for that temperature for the Bigelow model

### Examples

```
pars <- list(temp_ref = 55,
        z = 5.18 ,
        D_R = 12.10 )
detection_bigelow( pars, temperature = 57, limit=7)
```

## detection_mafart *Detection limit of the Mafart model*

### Description

Calculation of the detection limit for the Mafart model

### Usage

```
detection_mafart(pars, temperature, limit)
```

### Arguments

| | |
|---|---|
| pars | list defining the model parameters according to the rules defined in the bioinactivation package. |
| temperature | numerical value that describes the temperature at which the detection limit will be calculated |
| limit | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |

### Value

Numerical value that indicates the limit of detection for that temperature for the Mafart model

### Examples

```
pars <- list(temp_ref = 55,
        z = 5.18 ,
        p = 0.99 ,
        delta_ref = 11.96)
detection_mafart( pars, temperature = 57, limit=7)
```

## detection_peleg *Detection limit of the Peleg model*

### Description

Calculation of the detection limit for the Peleg model

### Usage

```
detection_peleg(pars, temperature, limit)
```

### Arguments

| | |
|---|---|
| pars | list defining the model parameters according to the rules defined in the bioinactivation package. |
| temperature | numerical value that describes the temperature at which the detection limit will be calculated |
| limit | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |

### Value

Numerical value that indicates the limit of detection for that temperature for the Peleg model

### Examples

```
pars <- list(temp_crit = 56.95,
        k_b = 0.58 ,
        n = 1 )
detection_peleg( pars, temperature = 57, limit=7)
```

## detFIM *Objective function for D-optimal OED*

### Description

Objective function for D-optimal OED

### Usage

```
detFIM(x, model, pars)
```

### Arguments

| | |
|---|---|
| `x` | a numeric vector of length `n` defining the design matrix. The first n/2 elements are the time points and the last n/2 are the temperatures of these points. |
| `model` | character string defining the inactivation model to use. |
| `pars` | list defining the model parameters according to the rules defined in the bioinactivation package. |

### Value

Numeric value of the objective function for criterium D, which is a determinant of the FIM.

### Examples

```
pars <- list(temp_crit = 55,
        n = 1.5,
        k_b = 0.1)
detFIM(x = c(10,15, 20, 25), "Peleg", pars)
```

## get_detection *Calculate detection limit*

### Description

Calculation of the detection limit depending on the model.

### Usage

```
get_detection(model, pars, temperature, limit)
```

### Arguments

| | |
|---|---|
| `model` | character string defining the inactivation model to use. |
| `pars` | list defining the model parameters according to the rules defined in the bioinactivation package. |
| `temperature` | numerical value that describes the temperature at which the detection limit will be calculated |
| `limit` | numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit. |

### Value

Numerical value that indicates the limit of detection

## get_isothermal_correlation *Parameter correlation for isothermal inactivation experiments*

### Description

Parameter correlation for isothermal inactivation experiments

### Usage

```
get_isothermal_correlation(model, exp_design, pars)
```

### Arguments

| | |
|---|---|
| `model` | character defining the inactivation model according to the rules in the `bioinactivation` package. |
| `exp_design` | data.frame with two columns named `times` and `temperature` describing the experiment design. |
| `pars` | list defining the model parameters according to the rules defined in the `bioinactivation` package. |

### Examples

```
library(tidyverse)
time_profile <- seq(0, 50, length = 20)
Temp_profile <- seq(52.5,60, length = 3)

exp_design <- expand.grid(time_profile,Temp_profile) %>%
  rename(times = Var1, temperature = Var2)

pars <- list(temp_crit = 55,
             n = 1.5,
             k_b = 0.1)

get_isothermal_correlation("Peleg", exp_design, pars )
```

---

## `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)

## Not run:
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)
```

```
## End(Not run)
```

## 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 printout on every step.

`...`
: Additional arguments passed to penalty_function.

## 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

## Not run:

## 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=500,  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)


## End(Not run)
```

## 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. |

## isothermal_OED *Optimal Experiment Design of isothermal inactivation*

### Description

Calculates an Optimal Experiment for an isothermal microbial inactivation experiment considering the maximum duration of the experiment according to the detection limit.

### Usage

```
isothermal_OED(model, pars, n_points, min_time, max_time, min_temp,
  max_temp, criterion = "D", opts = NULL)
```

### Arguments

`model` character string defining the inactivation model to use.

`pars` list defining the nominal model parameters.

`n_points` numerical stating the number of data points.

`min_time` numerical stating the lower limit for the time points.

`max_time` numerical stating the upper limit for the time points.

`min_temp` numerical stating the lower limit for the temperature.

`max_temp` numerical stating the upper limit for the temperature.

`criterion` character stating the criterion to use for the OED. function evaluations with local finish with the DHC algorithm (see help from MEIGO).

`opts` options for the MEIGO algorithm. By default, a maximum of 2000

### Value

A MEIGO object

### Examples

```
pars <- list(z = 4.2, D_R = 3.9, temp_ref = 55)
opts <- list(maxeval=200,local_finish="DHC")
## Not run:
OED <- isothermal_OED("Bigelow", pars, n_points = 5, criterion = "E-mod",
                      min_time = 0, max_time = 100, min_temp = 52.5, max_temp = 60,
                      opts = opts)
plot(OED)

## End(Not run)
```

## `isothermal_OED_limit` *OED of isothermal microbial inactivation with detection limit*

### Description

Calculates an Optimal Experiment for an isothermal microbial inactivation experiment considering the maximum duration of the experiment according to the detection limit.

### Usage

```
isothermal_OED_limit(model, pars, limit, n_points, min_time, max_time,
  min_temp, max_temp, criterion = "D", opts = NULL, x_0 = NULL)
```

### Arguments

`model` character string defining the inactivation model to use.

`pars` list defining the nominal model parameters.

`limit` numerical value describing the maximum number of log-reductions that can be identified in the experiment limit = logDL - logN0, where DL is the detection limit.

`n_points` numerical stating the number of data points.

`min_time` numerical stating the lower limit for the time points.

`max_time` numerical stating the upper limit for the time points.

`min_temp` numerical stating the lower limit for the temperature.

`max_temp` numerical stating the upper limit for the temperature.

`criterion` character string defining the criterion to use.

`opts` options for the MEIGO algorithm. By default, a maximum of 2000 function evaluations with local finish with the DHC algorithm (see help from MEIGO).

`x_0` initial point for the MEIGO algorithm. By default, it is NULL.

### Value

A MEIGO object

### Examples

```
pars <- list(z = 4.2, D_R = 3.9, temp_ref = 55)
opts <- list(maxeval=2000,local_finish="DHC")
## Not run:
OED <- isothermal_OED_limit("Bigelow", pars, n_points = 5, criterion = "E-mod", limit = 6,
                      min_time = 0, max_time = 100, min_temp = 52.5, max_temp = 60,
                      opts = opts)
plot(OED)

## End(Not run)
```

## isothermal_sensitivities

*Local sensitivites of isothermal microbial inactivation*

### Description

Local sensitivites of isothermal microbial inactivation

### Usage

```
isothermal_sensitivities(model, exp_design, pars)
```

### Arguments

| | |
|---|---|
| model | character defining the inactivation model according to the rules in the bioinactivation package. |
| exp_design | data.frame with two columns named times and temperature describing the experiment design. |
| pars | list defining the model parameters according to the rules defined in the bioinactivation package. |

### Value

A list of class "IsoSensitivities" with 3 entries:

**model** Inactivation model.

**pars** Model parameters used for the calculations.

**sensitivities** data.frame adding columns to exp_design with the calculated sensitivities. Local sensitivities are named as the parameters, scaled sensitivities as parameter_name+_scaled.

### Examples

```
library("tidyverse")
time_profile <- seq(0, 50, length = 20)
Temp_profile <- seq(52.5,60, length = 3)

exp_design <- expand.grid(time_profile,Temp_profile) %>%
  rename(times = Var1, temperature = Var2)

pars <- list(z = 4.2, D_R = 3.9, temp_ref = 55)

my_sensitivities <- isothermal_sensitivities("Bigelow", exp_design, pars)
plot(my_sensitivities)
plot(my_sensitivities, limit = 6)
```

## 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. |
| `...` | Additional arguments passed to penalty_function. |

## 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. |
| `...` | Additional arguments passed to penalty_function. |

## `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`.

## penalty_function *Penalty Function for OED*

### Description

Penalty Function for OED

### Usage

```
penalty_function(time_points, time_min, a_penalty = 1e+15,
  b_penalty = 2e+15)
```

### Arguments

| | |
|---|---|
| time_points | Numeric vector of time points for the measurements. |
| time_min | Numeric defining the minimum time between measurements. |
| a_penalty | Numeric defining the shape of the penalty function. 1e15 by default. |
| b_penalty | Numeric defining the shape of the penalty function. 2e15 by default. |

## plot.IsoSensitivities *Plotting of IsoSensitivities objects*

### Description

Plotting of IsoSensitivities objects

### Usage

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

### Arguments

| | |
|---|---|
| x | an object of class IsoSensitivities |
| y | ignored |
| ... | ignored |
| limit | Detection limit, NULL by default (not ploted) |

## `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.OEDisothermal` *Plot of OEDisothermal object*

### Description

Plot of OEDisothermal object

### Usage

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

### Arguments

| | |
|---|---|
| `x` | an object of class IsoSensitivities |
| `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. |

## `sensitivities_Bigelow` *Local sensitivities of the Bigelow model*

### Description

Local sensitivities of the Bigelow model

### Usage

```
sensitivities_Bigelow(exp_design, pars)
```

### Arguments

| | |
|---|---|
| `exp_design` | data.frame with two columns named `times` and `temperature` describing the experiment design. |
| `pars` | list defining the model parameters according to the rules defined in the `bioinactivation` package. |

### Value

A data frame with the same number of rows as `exp_design` with additional column for local sensitivities. These are named D_R and z for local sensitivities and D_R_scaled and z_scaled for scaled locan sensitivities.

## `sensitivities_Mafart` *Local sensitivities of the Mafart model*

### Description

Local sensitivities of the Mafart model

### Usage

```
sensitivities_Mafart(exp_design, pars)
```

### Arguments

| | |
|---|---|
| `exp_design` | data.frame with two columns named `times` and `temperature` describing the experiment design. |
| `pars` | list defining the model parameters according to the rules defined in the `bioinactivation` package. |

## sensitivities_Peleg *Local sensitivities of the Peleg model*

### Description

Local sensitivities of the Peleg model

### Usage

```
sensitivities_Peleg(exp_design, pars)
```

### Arguments

`exp_design` data.frame with two columns named `times` and `temperature` describing the experiment design.

`pars` list defining the model parameters according to the rules defined in the `bioinactivation` package.

## 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