

# Package ‘ino’

July 16, 2022

**Title** Initialization of Numerical Optimization

**Version** 0.1.0

**Description** Implementation of several initialization strategies for the numerical optimization of real-valued functions, in particular likelihood functions of statistical models.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.2.0

**Imports** ggplot2, rlang, RprobitB, mvtnorm, evd, crayon, progress, dplyr, foreach, doSNOW

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), purrr, pracma

**Config/testthat/edition** 3

**URL** <https://github.com/loelschlaeger/ino>

**BugReports** <https://github.com/loelschlaeger/ino/issues>

**VignetteBuilder** knitr

**Depends** R (>= 4.0.0)

**NeedsCompilation** no

**Author** Lennart Oelschläger [aut, cre]  
(<https://orcid.org/0000-0001-5421-9313>),  
Marius Ötting [aut] (<https://orcid.org/0000-0002-9373-0365>)

**Maintainer** Lennart Oelschläger <oelschlaeger.lennart@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-07-16 08:40:02 UTC

## R topics documented:

clear_ino . . . . .	2
earthquakes . . . . .	3
fixed_initialization . . . . .	3
merge_ino . . . . .	4
overview_optima . . . . .	4
plot.ino . . . . .	5
random_initialization . . . . .	5
setup_ino . . . . .	6
set_optimizer . . . . .	8
set_optimizer_nlm . . . . .	9
set_optimizer_optim . . . . .	10
standardize_initialization . . . . .	10
subset_initialization . . . . .	12
summary.ino . . . . .	13
<b>Index</b>	<b>14</b>

---

clear_ino	<i>Clear initialization runs</i>
-----------	----------------------------------

---

### Description

This function clears initialization runs saved in an ino object.

### Usage

```
clear_ino(x, which = "all")
```

### Arguments

x	An object of class ino.
which	Either "all" to clear all initialization runs, or alternatively a numeric vector of row numbers in x\$runs\$table.

### Value

The updated ino object.

---

earthquakes

*Earthquake data*

---

### Description

This data set includes the number of yearly measured earthquakes from 1900 to 2006.

### Usage

```
data(earthquakes)
```

### Format

The data set is a `data.frame` with two integer columns, `year` for the year and `obs` for the number of measured earthquakes.

### Source

The data was obtained from <http://hmms-for-time-series.de/second/data/earthquakes.txt> on 2022-03-25.

---

fixed\_initialization

*Fixed initialization*

---

### Description

This function is an implementation of the fixed initialization strategy.

### Usage

```
fixed_initialization(  
  x,  
  at,  
  ncores = getOption("ino_ncores"),  
  verbose = getOption("ino_progress")  
)
```

### Arguments

<code>x</code>	An object of class <code>ino</code> .
<code>at</code>	A vector containing the (fixed) initial values.
<code>ncores</code>	This function is parallelized, set the number of cores here.
<code>verbose</code>	Set to <code>TRUE</code> ( <code>FALSE</code> ) to print (hide) progress.

**Details**

For more details see the help vignette: `vignette("workflow", package = "ino")`

**Value**

The updated ino object.

---

merge_ino	<i>Merge initialization runs</i>
-----------	----------------------------------

---

**Description**

This function merges multiple ino objects.

**Usage**

```
merge_ino(...)
```

**Arguments**

... Arbitrary many ino objects, of which the initialization results are merged into the first object, which is then returned.

**Value**

The updated ino object.

---

overview_optima	<i>Overview of optima</i>
-----------------	---------------------------

---

**Description**

This function provides an overview of the identified optima.

**Usage**

```
overview_optima(x, digits = 2, plot = FALSE)
```

**Arguments**

x An object of class ino.  
 digits The number of digits of the optima values.  
 plot Set to TRUE for a visualization.

**Value**

Either a `data.frame` (if `plot = FALSE`), or otherwise a `ggplot` object.

---

`plot.ino`*Visualization of initialization*

---

**Description**

This function plots one or multiple summary statistics on the initialization runs in an `ino` object.

**Usage**

```
## S3 method for class 'ino'  
plot(x, var = ".time", by = ".strategy", type = "boxplot", ...)
```

**Arguments**

<code>x</code>	An object of class <code>ino</code> .
<code>var</code>	The name of the statistic to be plotted.
<code>by</code>	Plots the summary statistic <code>var</code> for all groups listed in <code>by</code> .
<code>type</code>	Governs the type of plot. Either <code>"boxplot"</code> , <code>"histogram"</code> , or <code>"barplot"</code> .
<code>...</code>	Additional arguments to be passed to the plotting function.

**Value**

An `ggplot` object.

---

`random_initialization` *Random initialization*

---

**Description**

This function is an implementation of the random initialization strategy.

**Usage**

```
random_initialization(  
  x,  
  sampler = stats::rnorm,  
  ...,  
  ncores = getOption("ino_ncores"),  
  verbose = getOption("ino_progress")  
)
```

**Arguments**

x	An object of class <code>ino</code> .
sampler	The sampler for random initial values. Can be any function that <ul style="list-style-type: none"> <li>• has as first argument an integer, say <code>npar</code>,</li> <li>• and returns a numeric vector of length <code>npar</code>.</li> </ul> Per default, <code>sampler = stats::rnorm</code> , i.e. independent draws from a standard normal distribution as initial value.
...	Additional argument to <code>sampler</code> (optional).
ncores	This function is parallelized, set the number of cores here.
verbose	Set to <code>TRUE</code> ( <code>FALSE</code> ) to print (hide) progress.

**Details**

For more details see the help vignette: `vignette("workflow", package = "ino")`

**Value**

The updated `ino` object.

---

setup\_ino

*Setup*

---

**Description**

Use this function to specify the numerical optimization problem. The function returns an object of class `ino` that contains all specifications.

**Usage**

```
setup_ino(
  f,
  npar,
  ...,
  opt = set_optimizer_nlm(),
  mpvs = character(0),
  verbose = getOption("ino_progress"),
  skip_test = FALSE
)
```

**Arguments**

f	An object of class function, the function to be optimized.
npar	The length of the first argument of f, i.e. the argument over which f is optimized.
...	Additional and named arguments to be passed to f (optional).
opt	The output of <code>set_optimizer</code> , which is an object of class optimizer. Per default, <code>opt = set_optimizer_nlm()</code> , which specifies the <code>nlm</code> optimizer. Can also be a list of different optimizer objects, see the details.
mpvs	A character vector of the argument names with multiple parameter values, see the details. Per default, <code>mpvs = character(0)</code> .
verbose	Set to TRUE (FALSE) to print (hide) the test results of the setup at the console.
skip_test	Set to TRUE to skip the specification tests.

**Details****Specifying a function:**

One function f must be specified per ino object. The function is optimized over its first argument, which should be a numeric vector of length npar, followed by any other arguments specified via the ... argument.

**Specifying an optimizer:**

The numerical optimizer must be specified via the opt argument as the output of `set_optimizer`. You can specify multiple optimizer for comparison, see below.

**Specifying multiple parameter values:**

You can specify multiple values for each ... parameter. Such arguments must be in list format, where each list element must be a valid parameter value. The names of these arguments must be added to the mpvs input to make clear that you want to iterate over them.

**Specifying multiple optimizer:**

Specifying multiple optimizer is analogue to specifying multiple parameter values: Submit a list of optimizer objects (i.e. outputs of `set_optimizer`) to the opt argument.

**An example:**

Say that f is a likelihood function of npar parameters that has the additional argument data. Say furthermore that you want to conduct a simulation experiment of the initialization effect for f for two different data sets. And last, say that you want to compare the `nlm` optimizer and the `optim` optimizer. Then, specify

```
setup_ino(
  f = f,
  npar = npar,
  data = list("data1" = <data set 1>,
             "data2" = <data set 2>),
  opt = list("nlm" = set_optimizer_nlm(),
            "optim" = set_optimizer_optim()),
  mpvs = c("data", "opt")
)
```

**Value**

An object of class `ino`.

**See Also**

`set_optimizer()` to specify an optimizer.

**Examples**

```
setup_ino(
  f = f_ll_hmm,
  npar = 4,
  data = earthquakes,
  N = 2,
  neg = TRUE,
  opt = set_optimizer_nlm()
)
```

---

set\_optimizer

*Specify numerical optimizer*

---

**Description**

Use this function to specify a numerical optimizer for the optimization problem.

**Usage**

```
set_optimizer(opt, f, p, v, z, ..., crit = character(0))
```

**Arguments**

<code>opt</code>	An object of class function, a numerical optimizer.
<code>f</code>	The name of the function input of <code>opt</code> .
<code>p</code>	The name of the starting parameter values input of <code>opt</code> .
<code>v</code>	The name of the optimal function value in the output list of <code>opt</code> .
<code>z</code>	The name of the optimal parameter vector in the output list of <code>opt</code> .
<code>...</code>	Additional arguments to be passed to the optimizer <code>opt</code> . Without specifications, the default values of <code>opt</code> are used.
<code>crit</code>	The names of additional elements in the output of <code>opt</code> to be saved after the optimization.



**Details**

Numerical optimizer specified for the `opt` argument must fulfill the following requirements:

- it must have an input `f` for the function to be optimized,
- it must have an input `p` for the starting parameter values,
- it must have a `...` argument for additional parameters to `f`,
- the output must be a named list, including the optimal function value (named `v`) and parameter vector (named `z`).

**Value**

An object of class `optimizer`, which can be passed to `setup_ino`.

**See Also**

`set_optimizer_nlm()` and `set_optimizer_optim()`, two wrappers for the `nlm` and `optim` optimizer.

**Examples**

```
set_optimizer(
  opt = pracma::nelder_mead,
  f = "fn",
  p = "x0",
  v = "fmin",
  z = "xmin",
  tol = 1e-6,
  crit = c("xmin", "fcount")
)
```

---

set\_optimizer\_nlm      *Specify nlm optimizer*

---

**Description**

Specify `nlm` optimizer

**Usage**

```
set_optimizer_nlm(..., crit = c("code", "iterations"))
```

**Arguments**

`...`      Additional arguments to be passed to the optimizer `opt`. Without specifications, the default values of `opt` are used.

`crit`      The names of additional elements in the output of `opt` to be saved after the optimization.

**Value**

An object of class optimizer, which can be passed to [setup\\_ino](#).

**See Also**

[set\\_optimizer\(\)](#) for specifying a different optimizer.

---

set_optimizer_optim	<i>Specify <a href="#">optim</a> optimizer</i>
---------------------	--

---

**Description**

Specify [optim](#) optimizer

**Usage**

```
set_optimizer_optim(..., crit = c("convergence"))
```

**Arguments**

...	Additional arguments to be passed to the optimizer opt. Without specifications, the default values of opt are used.
crit	The names of additional elements in the output of opt to be saved after the optimization.

**Value**

An object of class optimizer, which can be passed to [setup\\_ino](#).

**See Also**

[set\\_optimizer\(\)](#) for specifying a different optimizer.

---

standardize_initialization	<i>Standardize initialization</i>
----------------------------	-----------------------------------

---

**Description**

This function is an implementation of the standardize initialization strategy.

**Usage**

```
standardize_initialization(  
  x,  
  arg = "data",  
  by_col = TRUE,  
  center = TRUE,  
  scale = TRUE,  
  col_ign = NULL,  
  initialization = random_initialization(),  
  ncores = getOption("ino_ncores"),  
  verbose = getOption("ino_progress")  
)
```

**Arguments**

x	An object of class <code>ino</code> .
arg	A character, the name(s) of the argument(s) to be standardized. The argument must be of class <code>matrix</code> or <code>data.frame</code> . Per default, <code>arg = "data"</code> .
by_col	A boolean, set to <code>TRUE</code> (the default) to standardize column-wise, set to <code>FALSE</code> to standardize by rows.
center	A boolean, passed to <code>scale</code> .
scale	A boolean, passed to <code>scale</code> .
col_ign	A numeric vector of column indices (or row indices if <code>by_col = FALSE</code> ) that are ignored when standardizing.
initialization	An object of class <code>ino_call</code> which determines the initialization for the standardized <code>ino</code> object. The <code>ino_call</code> is generated by any of the strategy functions (any function with the name <code>*_initialization</code> ), when the <code>x</code> is unspecified. Per default, <code>initialization = random_initialization()</code> , i.e. random initialization.
ncores	This function is parallelized, set the number of cores here.
verbose	Set to <code>TRUE</code> ( <code>FALSE</code> ) to print (hide) progress.

**Details**

For more details see the help vignette: `vignette("workflow", package = "ino")`

**Value**

The updated `ino` object.

---

 subset\_initialization *Subset initialization*


---

### Description

This function is an implementation of the subset initialization strategy.

### Usage

```
subset_initialization(
  x,
  arg = "data",
  how = "random",
  prop = 0.5,
  by_row = TRUE,
  col_ign = NULL,
  kmeans_arg = list(centers = 2),
  initialization = random_initialization(),
  ncores = getOption("ino_ncores"),
  verbose = getOption("ino_progress")
)
```

### Arguments

x	An object of class <code>ino</code> .
arg	A character, the name of the argument to be subsetted. The argument must be of class <code>matrix</code> or <code>data.frame</code> . Per default, <code>arg = "data"</code> .
how	A character, specifying how to select the subset. Can be one of "random" (default), "first", and "kmeans".
prop	A numeric between 0 and 1, specifying the proportion of the subset.
by_row	A boolean, set to <code>TRUE</code> (the default) to subset by row, set to <code>FALSE</code> to subset by column.
col_ign	A numeric vector of column indices (or row indices if <code>by_row = FALSE</code> ) that are ignored when clustering. Only relevant if <code>how = "kmeans"</code> .
kmeans_arg	A list of additional arguments for <code>kmeans</code> . Per default, <code>kmeans_arg = list(centers = 2)</code> , which sets the number of clusters to 2. Only relevant if <code>how = "kmeans"</code> .
initialization	An object of class <code>ino_call</code> which determines the initialization for the standardized <code>ino</code> object. The <code>ino_call</code> is generated by any of the strategy functions, when the <code>x</code> is unspecified. Per default, <code>initialization = random_initialization()</code> , i.e. random initialization.
ncores	This function is parallelized, set the number of cores here.
verbose	Set to <code>TRUE</code> ( <code>FALSE</code> ) to print (hide) progress.

**Details**

For more details see the help vignette: `vignette("workflow", package = "ino")`

**Value**

The updated ino object.

---

summary.ino

*Summary of initialization*


---

**Description**

This function gives an overview of the initialization runs in an ino object.

**Usage**

```
## S3 method for class 'ino'
summary(object, group = NULL, ...)
```

**Arguments**

object	An object of class ino.
group	A character vector for grouping the optimization results, or NULL (default) for no grouping.
...	Named functions for computing statistics.

**Details**

The following values are available for each ino object:

- `.strategy`, the name of the initialization strategy,
- `.optimizer`, the name of the optimizer (if more than one),
- `.time`, the optimization time,
- `.optimum`, the function value at the optimum.

**Value**

A data.frame.

# Index

- \* **dataset**
  - earthquakes, 3
- \* **evaluation**
  - overview\_optima, 4
  - plot.ino, 5
  - summary.ino, 13
- \* **specification**
  - clear\_ino, 2
  - merge\_ino, 4
  - set\_optimizer, 8
  - set\_optimizer\_nlm, 9
  - set\_optimizer\_optim, 10
  - setup\_ino, 6
- \* **strategy**
  - fixed\_initialization, 3
  - random\_initialization, 5
  - standardize\_initialization, 10
  - subset\_initialization, 12

clear\_ino, 2

earthquakes, 3

fixed\_initialization, 3

kmeans, 12

merge\_ino, 4

nlm, 7, 9

optim, 7, 9, 10

overview\_optima, 4

plot.ino, 5

random\_initialization, 5

scale, 11

set\_optimizer, 7, 8

set\_optimizer(), 8, 10

set\_optimizer\_nlm, 9

set\_optimizer\_nlm(), 9

set\_optimizer\_optim, 10

set\_optimizer\_optim(), 9

setup\_ino, 6, 9, 10

standardize\_initialization, 10

subset\_initialization, 12

summary.ino, 13