

# Package ‘MaxWiK’

May 7, 2026

**Title** Machine Learning Method Based on Isolation Kernel Mean Embedding

**Version** 1.0.6

**Description** Incorporates Approximate Bayesian Computation to get a posterior distribution and to select a model optimal parameter for an observation point. Additionally, the meta-sampling heuristic algorithm is realized for parameter estimation, which requires no model runs and is dimension-independent. A sampling scheme is also presented that allows model runs and uses the meta-sampling for point generation. A predictor is realized as the meta-sampling for the model output. All the algorithms leverage a machine learning method utilizing the maxima weighted Isolation Kernel approach, or 'MaxWiK'. The method involves transforming raw data to a Hilbert space (mapping) and measuring the similarity between simulated points and the maxima weighted Isolation Kernel mapping corresponding to the observation point. Comprehensive details of the methodology can be found in the papers Iurii Nagornov (2024) <[doi:10.1007/978-3-031-66431-1\\_16](https://doi.org/10.1007/978-3-031-66431-1_16)> and Iurii Nagornov (2023) <[doi:10.1007/978-3-031-29168-5\\_18](https://doi.org/10.1007/978-3-031-29168-5_18)>.

**License** GPL (>= 3)

**Depends** R (>= 3.3.0)

**Imports** methods, stats, utils, scales, parallel, abc, ggplot2

**Suggests** rmarkdown, knitr

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**LazyData** true

**NeedsCompilation** no

**Author** Yuri Nagornov [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0002-7935-6776>>)

**Maintainer** Yuri Nagornov <nagornov.yuri@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-07-07 05:30:02 UTC

## Contents

apply_range . . . . .	2
Data.2D . . . . .	3
MaxWiK.ggplot.density . . . . .	3
MaxWiK_templates . . . . .	4
meta_sampling . . . . .	5
read_file . . . . .	8
read_hyperparameters . . . . .	9
restrict_data . . . . .	10
sampler_MaxWiK . . . . .	10

<b>Index</b>	<b>13</b>
--------------	-----------

---

apply_range	<i>Function to restrict values of the data according with the range for each dimension</i>
-------------	--------------------------------------------------------------------------------------------

---

### Description

Function to restrict values of the data according with the range for each dimension

### Usage

```
apply_range(diapason, input.data)
```

### Arguments

diapason	Vector of min and max values or data frame with two rows (min and max) for each dimension of input data
input.data	Data frame of input where values will be corrected

### Value

The same data frame with corrected values according to the diapason

### Examples

```
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the templates and vignettes for usage.
```

---

Data.2D

*List of the objects for the 2D example of the MaxWiK methods usage*

---

### Description

A list containing input and output data for 2D example for Approximate Bayesian Computation, including sampling scheme, meta-sampling, and prediction. To understand all details of the dataset, please, be kind to see vignette of the package.

### Usage

Data.2D

### Format

A list of:

**X** Input data frame of the model

**Y** Output data frame of the model

**observation** Data frame with observation info

**ABC** List of hyperparameters, the matrix of Voronoi sites, posteriori distribution, and results of MaxWiK algorithm

**metasampling** List of results of meta-sampling algorithm, and the network of points during meta-sampling

**sampling** List of object which are necessary for sampling algorithm like function for simulation, parameters of the model, MSE (mean squared error), and X12 - generated points

**predictor** List of object which are necessary for predictor algorithm like posteriori.MaxWiK, result of the algorithm, and network of points during meta-sampling

---

MaxWiK.ggplot.density *Density plot*

---

### Description

Density plot

**Usage**

```

MaxWiK.ggplot.density(
  title = "",
  datafr1,
  datafr2,
  var.df,
  obs.true = NULL,
  best.sim = NULL,
  clr = c("#a9b322", "#f9b3a2", "red", "blue"),
  alpha = c(0.1, 0.4),
  lw = c(0.7, 0.7),
  lt = c("dashed", "dotted")
)

```

**Arguments**

title	Title of the plot
datafr1	data frame 1
datafr2	data frame 2
var.df	Variables to show
obs.true	True observation if so, NULL by default
best.sim	The best point from a simulation if so, NULL by default
clr	Colors to plot, by default it is c( "#a9b322", "#f9b3a2", 'red', 'blue' )
alpha	Transparency values for density plots
lw	Line widths
lt	Line types

**Value**

Make and return the ggplot object of the densities of the data frames

**Examples**

```

MaxWiK::MaxWiK_templates(dir = tempdir()) # See the templates and vignettes for usage.
# Function 'MaxWiK.ggplot.density()' is used in the MaxWiK.ABC.R and
# MaxWiK.Predictor.R templates.

```

---

MaxWiK_templates	<i>Function to copy the templates from extdata folder in the library to /Templates/ folder in the working directory</i>
------------------	-------------------------------------------------------------------------------------------------------------------------

---

**Description**

Function to copy the templates from extdata folder in the library to /Templates/ folder in the working directory

**Usage**

```
MaxWiK_templates(dir)
```

**Arguments**

dir                    Folder to where files should be save, by default dir = './'

**Value**

List of logic numbers for each copied file, TRUE - success, FALSE - not success

**Examples**

```
MaxWiK_templates( dir = tempdir() )
```

---

meta_sampling	<i>Function to get Approximate Bayesian Computation based on Maxima Weighted Isolation Kernel mapping</i>
---------------	-----------------------------------------------------------------------------------------------------------

---

**Description**

The function meta\_sampling() iteratively generates tracer based on the simple procedure:

- making a reflection of the top points from the best point,
- and then generating the point tracers between them,
- finally, the algorithm chooses again the top points and the best point (sudoku() function is used),
- repeat all the steps until condition to be TRUE:  
 $\text{abs}(\min(\text{sim\_tracers}) - \text{sim\_previous}) < \text{epsilon}$

The function MaxWiK.predictor() uses the meta-sampling for a prediction

The function get.MaxWiK() is used to get Approximate Bayesian Computation based on Maxima Weighted Isolation Kernel mapping. On given data frame of parameters, statistics of the simulations and an observation, using the internal parameters psi and t, the function get.MaxWiK() returns the estimation of a parameter corresponding to Maxima weighted Isolation Kernel ABC method.

**Usage**

```
meta_sampling(
  psi = 4,
  t = 35,
  param,
  stat.sim,
  stat.obs,
```

```

talkative = FALSE,
check_pos_def = FALSE,
n_bullets = 16,
n_best = 10,
halfwidth = 0.5,
epsilon = 0.001,
rate = 0.1,
max_iteration = 15,
save_web = TRUE,
use.iKernelABC = NULL
)

```

```

MaxWiK.predictor(
  psi = 4,
  t = 35,
  param,
  stat.sim,
  new.param,
  talkative = FALSE,
  check_pos_def = FALSE,
  n_bullets = 16,
  n_best = 10,
  halfwidth = 0.5,
  epsilon = 0.001,
  rate = 0.1,
  max_iteration = 15,
  save_web = TRUE,
  use.iKernelABC = NULL
)

```

```

get.MaxWiK(
  psi = 40,
  t = 350,
  param,
  stat.sim,
  stat.obs,
  talkative = FALSE,
  check_pos_def = TRUE,
  Matrix_Voronoi = NULL
)

```

### Arguments

psi	Integer number. Size of each Voronoi diagram or number of areas/points in the Voronoi diagrams
t	Integer number of trees in the Isolation Forest
param	or par . sim - data frame of parameters of the model
stat.sim	Summary statistics of the simulations (model output)

stat.obs	Summary statistics of the observation point
talkative	Logical parameter to print or do not print messages
check_pos_def	Logical parameter to check the Gram matrix is positive definite or do not check
n_bullets	Number of generating points between two
n_best	Number of the best points to construct the next web net
halfwidth	Parameter for the algorithm of deleting of generated points
epsilon	Criterion to stop meta-sampling
rate	Rate to renew points in the web net of generated points
max_iteration	Maximum of iterations during meta-sampling
save_web	Logical to save all the generated points (web net)
use.iKernelABC	The iKernelABC object to use for meta-sampling. By default it is NULL and is generated.
new.param	New parameter for the predictor input
Matrix_Voronoi	is a predefined matrix of information about Voronoi trees (rows - trees, columns - Voronoi points/areas IDs). By default it is NULL and is generated randomly.

### Value

The function `meta_sampling()` returns the list of the next objects:

- `input.parameters` that is the list of all the input parameters for Isolation Kernel ABC method;
- `iteration` that is iteration value when algorithm stopped;
- `network` that is network points when algorithm stopped;
- `par.best` that is data frame of one point that is the best from all the generated tracer points;
- `sim.best` that is numeric value of the similarity of the best tracer point;
- `iKernelABC` that is result of the function `get.MaxWiK()` given on input parameters;
- `spiderweb` that is the list of all the networks during the meta-sampling.

The function `MaxWiK.predictor()` returns the list of the next objects:

- `input.parameters` that is the list of all the input parameters for Isolation Kernel ABC method;
- `iteration` that is iteration value when algorithm stopped;
- `network` that is network points when algorithm stopped;
- `prediction.best` that is data frame of one point that is the best from all the generated tracer points;
- `sim.best` that is numeric value of the similarity of the best tracer point;
- `iKernelABC` that is result of the function `get.MaxWiK()` given on input parameters;
- `spiderweb` that is the list of all the networks during the meta-sampling.

The function `get.MaxWiK()` returns the list of :

- `kernel_mean_embedding` is a maxima weighted kernel mean embedding (mapping) related to the observation point;

- parameters\_Matrix\_Voronoi is a matrix of information about Voronoi trees (rows - trees, columns - Voronoi points/areas IDs) for parameters data set;
- parameters\_Matrix\_iKernel is a matrix of all points of PARAMETERS in a Hilbert space (rows - points, columns - isolation trees);
- Hilbert\_weights is a weights in Hilbert space to get maxima weighted kernel mean embedding for parameters\_Matrix\_iKernel;
- Matrix\_iKernel is a matrix of all points of simulations in a Hilbert space (rows - points, columns - isolation trees);
- iFeature\_point is a feature embedding mapping for the OBSERVATION point;
- similarity is a vector of similarities between the simulation points and observation point;
- Matrix\_Voronoi is a matrix of information about Voronoi trees (rows - trees, columns - Voronoi points/areas IDs);
- t is a number of trees in the Isolation Forest;
- psi is a number of areas/points in the Voronoi diagrams

### Functions

- meta\_sampling(): The function to get the best value of parameter corresponding to Maxima Weighted Isolation Kernel mapping which is related to an observation point
- MaxWiK.predictor(): The function to get the prediction of output based on a new parameter and MaxWiK

### Examples

```
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the template 'MaxWiK.ABC.R' and
# vignettes for usage.
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the template 'MaxWiK.Predictor.R'
# and vignettes for usage.
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the template 'MaxWiK.ABC.R' and
# vignettes for usage.
```

---

read\_file

*Function to read file*

---

### Description

Function to read file

### Usage

```
read_file(file_name = "", stringsAsFactors = FALSE, header = TRUE)
```

**Arguments**

file\_name      Name of file to read  
stringsAsFactors      Parameter for read.table function, by default stringsAsFactors = FALSE  
header      Logical type to read or do not read head of a file

**Value**

data.frame of data from a file

**Examples**

NULL

---

*read\_hyperparameters*      *Function to read hyperparameters and their values from the file*

---

**Description**

Function to read hyperparameters and their values from the file

**Usage**

```
read_hyperparameters(input)
```

**Arguments**

input      File name to input

**Value**

Parameters and their values

**Examples**

```
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the templates and vignettes for usage.
```

---

restrict_data	<i>Function to restrict data in the size to accelerate the calculations</i>
---------------	-----------------------------------------------------------------------------

---

**Description**

restrict\_data() is based on rejection ABC method to restrict original dataset

**Usage**

```
restrict_data(par.sim, stat.sim, stat.obs, size = 300)
```

**Arguments**

par.sim	Data frame of parameters
stat.sim	Data frame of outputs of simulations
stat.obs	Data frame of observation point
size	Integer number of points to leave from original dataset

**Value**

restrict\_data() returns the list of:  
 par.sim - restricted parameters which are close to observation point  
 stat.sim - restricted stat.sim which are close to observation point

**Examples**

```
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the templates and vignettes for usage.
```

---

sampler_MaxWiK	<i>Function to generate parameters and simulate a model based on MaxWiK algorithm</i>
----------------	---------------------------------------------------------------------------------------

---

**Description**

Function to generate parameters and simulate a model based on MaxWiK algorithm

**Usage**

```
sampler_MaxWiK(
  stat.obs,
  stat.sim,
  par.sim,
  model,
  arg0 = list(),
  size = 500,
```

```

    psi_t,
    epsilon,
    nmax = 100,
    include_top = FALSE,
    slowly = FALSE,
    rate = 0.2,
    n_simulation_stop = NA,
    check_err = TRUE,
    include_web_rings = TRUE,
    number_of_nodes_in_ring = 2
  )

sampler_MaxWiK_parallel(
  stat.obs,
  stat.sim,
  par.sim,
  model,
  arg0 = list(),
  size = 500,
  psi_t,
  epsilon,
  nmax = 100,
  include_top = FALSE,
  slowly = FALSE,
  rate = 0.2,
  n_simulation_stop = NA,
  check_err = TRUE,
  include_web_rings = TRUE,
  number_of_nodes_in_ring = 2,
  cores = 4
)

```

### Arguments

stat.obs	Summary statistics of the observation point
stat.sim	Summary statistics of the simulations (model output)
par.sim	Data frame of parameters of the model
model	Function to get output of simulation during sampling
arg0	List with arguments for a model function, so that arg0 is NOT changed during sampling
size	Number of points in the simulation based on MaxWiK algorithm
psi_t	Vector of psi and t hyperparameters.
epsilon	Criterion to stop simulation when $MSE_{current} - MSE_{previous} < \epsilon$
nmax	Maximal number of iterations
include_top	Logical to include top points (network) from spider_web() function to simulate or do not

slowly	Logical for two algorithms: slow and fast seekers in sampling
rate	Rate value in the range $[0, 1]$ to define the rate of changing in the original top of sampled points for slow scheme (if slowly = TRUE)
n_simulation_stop	Maximal number of simulations to stop sampling. If n_simulation_stop = NA then there is no restriction (by default)
check_err	Logical parameter to check epsilon or do not
include_web_rings	Logical to include or do not include the cobweb rings to the simulations
number_of_nodes_in_ring	Number of points/nodes between two points in the web ring. By default number_of_nodes_in_ring = 2
cores	Number of cores for parallel calculations of a model (4 by default)

### Value

sampler\_MaxWiK() returns the list:

- results: results of all the simulations;
- best: the best value of parameter;
- MSE\_min: minimum of MSE;
- number\_of\_iterations: number of iterations;
- time: time of sampling in seconds,
- n\_simulations: the total number of simulations.

sampler\_MaxWiK\_parallel() returns the same output as in sampler\_MaxWiK().

### Functions

- sampler\_MaxWiK\_parallel(): Function to generate parameters and simulate a model based on MaxWiK algorithm

### Examples

```
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the template 'MaxWiK.Sampling.R'
# and vignettes for usage.
MaxWiK::MaxWiK_templates(dir = tempdir()) # See the template 'MaxWiK.Sampling.R'
# and vignettes for usage. For parallel implementation
# change the function 'sampler_MaxWiK()' to 'sampler_MaxWiK_parallel()'.
```

# Index

## \* datasets

Data.2D, [3](#)

apply\_range, [2](#)

Data.2D, [3](#)

get.MaxWiK (meta\_sampling), [5](#)

MaxWiK.ggplot.density, [3](#)

MaxWiK.predictor (meta\_sampling), [5](#)

MaxWiK\_templates, [4](#)

meta\_sampling, [5](#)

read\_file, [8](#)

read\_hyperparameters, [9](#)

restrict\_data, [10](#)

sampler\_MaxWiK, [10](#)

sampler\_MaxWiK\_parallel  
(sampler\_MaxWiK), [10](#)