

Package ‘MEIGOR’

November 14, 2025

Type Package

Title MEIGOR - MEtaheuristics for bioinformatics Global Optimization

Version 1.45.0

Date 2020-06-09

Maintainer Jose A. Egea <josea.egea@gmail.com>

Description MEIGOR provides a comprehensive environment for performing global optimization tasks in bioinformatics and systems biology.

It leverages advanced metaheuristic algorithms to efficiently search the solution space and is specifically tailored to handle the complexity and high-dimensionality of biological datasets.

This package supports various optimization routines and is integrated with Bioconductor's infrastructure for a seamless analysis workflow.

License GPL-3

Encoding UTF-8

LazyData false

biocViews SystemsBiology, Optimization, Software

Depends R (>= 4.0), Rsolnp, snowfall, deSolve, CNORode

Suggests CellNOptR, knitr, BiocStyle

VignetteBuilder knitr

NeedsCompilation no

git_url <https://git.bioconductor.org/packages/MEIGOR>

git_branch devel

git_last_commit 2caec25

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-13

Author Jose A. Egea [aut, cre],
David Henriques [aut],
Alexandre Fdez. Villaverde [aut],
Thomas Cokelaer [aut]

Contents

MEIGOR-package	3
essR-package	4
BayesFit	5
CeSSR	5
CeVNSR	8
cnolist	9
cnolist_cellnopt	9
cur_params	10
dhc	13
essR	13
essR_multistart	18
eucl_dist	18
ith_essR	19
ith_VNSR	19
MEIGO	19
model	20
model_cellnopt	20
nls_fobj	20
optim_fobj	21
paramsOpt	21
runBayesFit	21
rvnds_hamming	23
rvnds_local	25
solnp_eq	25
solnp_fobj	25
solnp_ineq	26
ssm_beyond	26
ssm_defaults	26
ssm_evalfc	27
ssm_isdif2	27
ssm_localsolver	27
ssm_optset	27
ssm_penalty_function	28
ssm_round_int	28
vns_defaults	28
vns_optset	29

Index

30

MEIGOR-package*Global optimization Toolbox*

Description

A global optimization package containing several algorithms such as a scatter search implementation and variable neighborhood search (plus cooperative multicore/multimachine implementations of these) and dynamic hill climbing.

Details

Package:	MEIGOR
Type:	Package
Version:	0.99.0
Date:	2012-09-10
License:	GPLv3
LazyLoad:	yes

Author(s)

Jose Egea David Henriques Thomas Cokelaer Alejandro F. Villaverde Julio R. Banga Julio Saez-Rodriguez

Maintainer: Jose Egea <josea.egea@upct.es>

References

Egea, J.A., Maria, R., Banga, J.R. (2010) An evolutionary method for complex-process optimization. *Computers & Operations Research* 37(2): 315-324.

Egea, J.A., Balsa-Canto, E., Garcia, M.S.G., Banga, J.R. (2009) Dynamic optimization of nonlinear processes with an enhanced scatter search method. *Industrial & Engineering Chemistry Research* 49(9): 4388-4401.

See Also

[essR](#) [essR](#)

Description

essR attempts to solve problems of the form:

```

min F(x)  subject to:
x
  ceq(x) = 0 (equality constraints)
  c_L <=
```

See Also

[essR](#)

BayesFit	<i>BayesFit</i>
----------	-----------------

Description

BayesFit

Details

Check `essR` documentation for more information about the input arguments.

Value

<code>f_mean</code>	Vector with size of <code>n_iter+1</code> containing the mean value of the objective function in each iteration.
<code>f_sd</code>	Vector with size of <code>n_iter+</code>

```

dim2 = 33;      bal2 = 0;      n2_2 = 0;
dim3 = 46;      bal3 = 0;      n2_3 = 2;
dim4 = 56;      bal4 = 0;      n2_4 = 4;
dim5 = 72;      bal5 = 0.25;  n2_5 = 7;
dim6 = 72;      bal6 = 0.25;  n2_6 = 10;
dim7 = 88;      bal7 = 0.25;  n2_7 = 15;
dim8 = 101;     bal8 = 0.5;   n2_8 = 20;
dim9 = 111;     bal9 = 0.25;  n2_9 = 50;

```

```

opts$ce_isparallel=TRUE;

#Call the solver
Results<-MEIGO(problem,opts,algorithm="CeSSR")
sfStop()

```

CeVNSR

Global optimization algorithm for MINLPs based on VNS using a Co-operative Strategy

Description

Value

f_mean	Vector with size of n_iter+1 containing the mean value of the objective function in each iteration.
f_sd	Vector with size of n_iter+1 containing the standard deviation value of the objective function in each iteration.
fbest	Vector with size of n_iter+1 containing the best value of the objective function in each iteration.
iteration_res	A list containing the results from every VNS instance initialized. It follows the format: results\$iteration_res[[iteration+1]][[thread_number]].

cur_params

*Current values of all model parameters***Description**

For a given set of values for the parameters to be estimated, this method returns an array containing the actual (not log-transformed) values of all model parameters, not just those to be estimated, in the same order as specified in the model. This is helpful when simulating the model at a given position in parameter space.

range [0,1], optional. Temperature for thermodynamic integration support. Used to scale likelihood when calculating the posterior value. Defaults to 1, i.e. no effect.

output

List of output values with entries as explained below. num_estimate: int. Number of parameters to estimate. estimate_idx: list of int. Indices of parameters to estimate in the model's full parameter list. initial_values: list of float. Starting values for parameters to estimate, taken from the parameters' nominal values in the model or explicitly specified in 'options'. initial_position: list

estimated, regardless.

Value

A list of the values of all model parameters.

Examples

```
data("simpleExample", package="MEIGOR")
initial_pars = createLBodeContPars(model, LB_n=1, LB_k
```

dhc	<i>Local search algorithm within eSS</i>
-----	--

Description

Local search algorithm within eSS

Note

For internal use of MEIGOR

problem\$f_0: Function values of initial point(s) (optional). These values MUST correspond to feasible points.

NOTE: The dimension of f_0 and x_0 may be different. For example, if we want to introduce 5 initial points but we only know the values for 3 of them, x_0 would have 5 rows whereas f_0 would have only 3 elements. In this example, it is mandatory that the first 3 rows of x_0 correspond to the values of f_0 .

opts\$local_balance: Balances between quality (=0) and diversity (=1) for choosing initial points for the local search (default 0.5).

opts\$local_finish: Applies local search to the best solution found once the optimization is finished (same values as opts.local.solver).

opts\$local_bestx: When activated (i.e. =1) only applies local search to the best solution found to date, ignoring filters (Default=0).

References

If you use essR and publish the results, please cite the following papers:

Egea, J.A., Maria, R., Banga, J.R. (2010) An evolutionary method for complex-process optimization. *Computers & Operations Research* 37(2): 315-324.

Egea, J.A., Balsa-Canto, E., Garcia, M.S.G., Bang

```

#===== END OF PROBLEM SPECIFICATIONS =====

Results<-essR(problem,opts);

#3 Constrained problem with equality constraints

ex3<-function(x,k1,k2,k3,k4){
  f=-x[4];
  #Equality constraints
  g<-rep(0,5);
  g[1]=x[4]-x[3]+x[2]-x[1]+k4*x[4]*x[6];
  g[2]=x[1]-1+k1*x[1]*x[5];
  g[3]=x[2]-x[1]+k2*x[2]*x[6];
  g[4]=x[3]+x[1]-1+k3*x[3]*x[5];

  #Inequality constraint
  g[5]=x[5]^0.5+x[6]^0.
```

```
g[2] = x[2]^2 + 2.0*x[3]^2 + x[1]^2 + 2.0*x[4]^2 - x[2] - x[4];
g[3] = 2.0*x[2]^2 + x[3]^2 + x[1]^2 + 2.0*x[2] - x[3] - x[4];
return(list(F=F, g=g));
}

# global optimum
#x*=[2.23607, 0, 1, 0];
#f(x*)=-40.9575;

#===== PROBLEM SPECIFICATIONS =====
problem<-list(f="ex4", x_L=rep(0,4), x_U=rep(10,4), x_0=c(3,4,5,1
```

ith_essR	<i>Auxiliary function to perform parallel runs</i>
----------	--

Description

Auxiliary function to perform parallel runs

Note

For internal use

Examples

```
#global optimum

#x*=[0.0898, -0.7127];
# or
#x*=[-0.0898, 0.7127];
#
#f(x*)= -1.03163;

ex1 <- function(x){
  y<-4*x[1]*x[1]-2.1*x[1]^4+1/3*x[1]^6+x[1]*x[2]-4*x[2]*x[2]+4*x[2]^4;
  return(y)
}

#===== PROBLEM SPECIFICATIONS =====
problem
```

optim_fobj	<i>Gateway function to evaluate the objective function when the local solvers are invoked.</i>
------------	--

Description

This function is used internally by essR to evaluate the objective function when the local solvers are invoked.

Note

use_hessian: logical, optional. Whether to use the Hessian to guide the walk. Defaults to FALSE. rtol: float or list of float, optional. Relative tolerance for ode solver. atol: float or list of float, optional. Absolute tolerance for ode solver. norm_step_size: float, optional. MCMC step size. Defaults to a reasonable value. hessian_period: int, optional. Number of MCMC steps between Hessian recalculations. Defaults to a reasonable but fairly large value, as Hessian calculation is expensive. hessian_scale: float, optional. Scaling factor used in generating Hessian-guided steps. Defaults to a reasonable value. sigma_adj_interval: int, optional. How often to adjust 'output\$sig_value' while annealing to meet 'accept_rate_target'. Defaults to a reasonable value.

```

"tspan"=NULL, "step_fn"=NULL, "likelihood_fn"=NULL,
"prior_fn"=NULL, "nsteps"=NULL, "use_hessian"=FALSE,
"rtol"=NULL, "atol"=NULL, "norm_step_size"=0.75,
"hessian_period"=25000, "hessian_scale"=0.085,
"sigma_adj_interval"=NULL, "anneal_length"=NULL,
"T_init"=10, "accept_rate_target"=0.3, "sigma_max"=1,
"sigma_min"=0.25, "sigma_step"=0.125, "thermo_temp"=1, "
```

User options:

opts\$maxeval: Maximum number of function evaluations (Default 1000).

opts\$maxtime: Maximum CPU time in seconds (Default 60).

opts\$maxdist: Percentage of the problem dimension which will be perturbed in the furthest neighborhood (varies between 0 and 1, default is 0.5).

rvnds_local	<i>Local search in VNS</i>
-------------	----------------------------

Description

Local search in VNS

Note

For internal use of MEIGOR.

solnp_ineq	<i>Gateway function to evaluate the inequality constraints when solnp is invoked as local solver</i>
------------	--

Description

This function is used internally by essR to evaluate the inequality constraints when solnp is invoked as local solver

Note

`ssm_evalfc`*Gateway function to evaluate the objective function in essR*

Description

This function is used internally by essR to evaluate the objective function.

Note

ssm_penalty_function	<i>Calculates the penalized objective function in constrained problems</i>
----------------------	--

Description

This function is used internally by essR to calculate the penalized objective function in constrained problems

Note

vns_optset	<i>Set VNS options</i>
------------	------------------------

Description

Set VNS options

Note

For internal use of MEIGOR.

Index

- * **Neighbourhood**

- CeVNSR, 8

- * **cooperative**

- CeSSR, 5

- CeVNSR, 8