

# Package: ROptimus (via r-universe)

September 2, 2024

**Title** A Parallel General-Purpose Adaptive Optimisation Engine

**Version** 3.0.0

**Description** A general-purpose optimisation engine that supports i) Monte Carlo optimisation with Metropolis criterion [Metropolis et al. (1953) <doi:10.1063/1.1699114>, Hastings (1970) <doi:10.1093/biomet/57.1.97>] and Acceptance Ratio Simulated Annealing [Kirkpatrick et al. (1983) <doi:10.1126/science.220.4598.671>, Černý (1985) <doi:10.1007/BF00940812>] on multiple cores, and ii) Acceptance Ratio Replica Exchange Monte Carlo Optimisation. In each case, the system pseudo-temperature is dynamically adjusted such that the observed acceptance ratio is kept near to the desired (fixed or changing) acceptance ratio.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Imports** doParallel (>= 1.0.11), foreach (>= 1.4.4), iterators (>= 1.0.9), parallel (>= 3.4.2)

**Depends** R (>= 3.1.0)

**LazyData** true

**URL** <https://github.com/SahakyanLab/ROptimus>

**NeedsCompilation** no

**Repository** <https://sahakyanlab.r-universe.dev>

**RemoteUrl** <https://github.com/sahakyanlab/roptimus>

**RemoteRef** HEAD

**RemoteSha** d1f24ef6610c850b3dc024d344216a43f1d84fdb

## Contents

|                           |   |
|---------------------------|---|
| ex.m.fun . . . . .        | 2 |
| ex.r.fun . . . . .        | 2 |
| ex.u.fun . . . . .        | 3 |
| IJ_ORIG . . . . .         | 3 |
| Optimus . . . . .         | 4 |
| OptimusExamples . . . . . | 6 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>8</b> |
|--------------|----------|

---

|          |                                |
|----------|--------------------------------|
| ex.m.fun | <i>Tutorial 5 m() function</i> |
|----------|--------------------------------|

---

### Description

The m() function supplied to Optimus() for reproducing Tutorial 5. This is for the small executable example.

### Usage

ex.m.fun

### Format

Function

---

|          |                                |
|----------|--------------------------------|
| ex.r.fun | <i>Tutorial 5 r() function</i> |
|----------|--------------------------------|

---

### Description

The r() function supplied to Optimus() for reproducing Tutorial 5. This is for the small executable example.

### Usage

ex.r.fun

### Format

Function

---

`ex.u.fun`*Tutorial 5 u() function*

---

**Description**

The `u()` function supplied to `Optimus()` for reproducing Tutorial 5. This is for the small executable example.

**Usage**`ex.u.fun`**Format**

Function

---

`IJ_ORIG`*Tutorial 5 Genomic Contact Data*

---

**Description**

A dataset containing 734 genomic contact pairs for reproducing Tutorial 5. Here, we take a set of 734 pairs of *i* and *j* 40-kilobase DNA regions that are known to be in contact inside a cell. Binning the DNA (e.g., a single chromosome) into 40-kb regions, each region is represented as a single integer that is equal to its end position divided by the length of the region, which is 40 kb. For instance, the 1st region, with start and end positions at the 1st and 40000th nucleotides, respectively, is denoted as 1 (40000th base / 40000 bases = 1). This simplifies the notation for a contact between two regions to a pair of positive integers as in used in this dataset.

**Usage**`IJ_ORIG`**Format**

A data frame with 734 rows and 2 variables:

**i** Integer representing 40-kb contact region partner of *j*

**j** Integer representing 40-kb contact region partner of *i*

---

 Optimus

*Acceptance Ratio Simulated Annealing and Acceptance Ratio Replica Exchange Monte Carlo Optimisation Engine*


---

## Description

Acceptance Ratio Simulated Annealing and Acceptance Ratio Replica Exchange Monte Carlo Optimisation Engine

## Usage

```

Optimus(
  NUMITER = 1e+06,
  STATWINDOW = 70,
  T.INI = 1e-05,
  T.ADJSTEP = 5e-09,
  TSCLnum = 2,
  T.SCALING = 3,
  T.MIN = 5e-09,
  T.DELTA = 2,
  DUMP.FREQ = 10000,
  LIVEPLOT = TRUE,
  LIVEPLOT.FREQ = 1e+05,
  PDFheight = 29,
  PDFwidth = 20,
  NCPU = 4,
  LONG = TRUE,
  SEED = 840,
  OPTNAME = "",
  DATA = NULL,
  K.INITIAL = 0,
  rDEF,
  mDEF,
  uDEF,
  EXCHANGE.FREQ = 1000,
  ACCRATIO = c(90, 50, 5, 1),
  CYCLES = 10,
  ACCRATIO.IN = 90,
  ACCRATIO.FIN = 0.5,
  OPT.TYPE = "SA",
  DIR,
  starcore = NULL
)

```

## Arguments

NUMITER            Number of model optimisation steps.

|               |  |
|---------------|--|
| STATWINDOW    | Number of last ongoing iterations to calculate acceptance ratio for temperature auto-adjustment.   |
| T.INI         | Initial temperature (K) for Metropolis criterion.  |
| T.ADJSTEP     | Temperature change step-size for temperature auto-adjustment based on the actual acceptance ratio.   |
| T.SCLnum      | Cutoff for one of the NumofAccRatSMIdeal and NumofAccRatGRIdeal numbers after which the adjustment step is multiplied by T.SCALING.  |
| T.SCALING     | See above.   |
| T.MIN         | Value to which the pseudo-temperature is set when the  |
| T.DELTA       | Minimum value by which acceptance ratio in a STATWINDOW must differ from the ideal acceptance ratio for the temperature control unit to make a temperature adjustment.                       |
| DUMP.FREQ     | The frequency (in steps) of writing the found ongoing best model.  |
| LIVEPLOT      | Plotting the optimisation process in a pdf file.   |
| LIVEPLOT.FREQ | Frequency (in steps) of plotting the results.  |
| PDFheight     | Plot height in inches.   |
| PDFwidth      | Plot width in inches.  |
| NCPU          | Number of CPU cores to use, by running more replicas of the optimisation. The usage of more than 1 cores will attempt to load the foreach and doParallel libraries in the case of SA Optimus |
| LONG          | If TRUE, it means that a long simulation is expected to be done, hence the memory-friendly mode will be activated.   |
| SEED          | Setting the seed for the random number generator.  |
| OPTNAME       | The name of the optimisation process.  |
| DATA          | A list that holds any supplementary data that functions mDEF or uDEF need to access.   |
| K.INITIAL     | The initial parameter configuration from which the optimisation process will begin.  |
| rDEF          | Function that defines a rule by which the parameters(K) are randomly altered.  |
| mDEF          | Model function that operates on the parameters to be optimized (K) and returns an observable object O.   |
| uDEF          | Function that evaluates the performance of a given set of parameters K.  |
| EXCHANGE.FREQ | Frequency of exchanges (NUMITER should be divisible by this number, for RE Optimus).   |
| ACCRATIO      | Vector of Acceptance Ratios for each replica (length of ACCRATIO must be equal to NCPU, for RE Optimus).   |
| CYCLES        | Number of annealing cycles (NUMITER should be divisible by this number, for SA Optimus).   |
| ACCRATIO.IN   | Initial acceptance ratio (%) at the beginning of each constituent annealing cycles (for SA Optimus).   |

|              |   |
|--------------|---|
| ACCRATIO.FIN | Final acceptance ratio (%) at the end of each constituent annealing cycles (for SA Optimus).  |
| OPT.TYPE     | String specifying which optimisation protocol to use. Enter "SA" for Simulated Annealing or "RE" for Replica Exchange (default value is "SA") |
| DIR          | String specifying which optimisation protocol to use.   |
| starcore     | Experimental variable of type list, holding some parameters for in-lab starcore use only.   |

**Value**

A probabilistic optimal parameter configuration K.

**Examples**

```
K <- IJ_ORIG
K$j <- sample(x=K$j, size=nrow(K), replace=FALSE)
out.dir <- tempdir()

Optimus(NCPU=1, OPTNAME="IJ.NEW.OPTI.SA", NUMITER=500, CYCLES=2, DIR=out.dir,
        DUMP.FREQ=10, LONG=FALSE, OPT.TYPE="SA", K.INITIAL=K,
        rDEF=ex.r.fun, mDEF=ex.m.fun, uDEF=ex.u.fun,
        DATA=list(IJ_ORIG=IJ_ORIG, gaplimit=50, numContacts=nrow(IJ_ORIG)))
```

---

OptimusExamples

*Generate script for reproducing a tutorial*

---

**Description**

Generate script for reproducing a tutorial

**Usage**

```
OptimusExamples(
  dir,
  example = 1,
  method = "SA",
  file_name = "example.R",
  mopac = NULL,
  vignette = TRUE
)
```

**Arguments**

|         |   |
|---------|---|
| dir     | String specifying directory where to save the script and vignette.                  |
| example | Integer specifying tutorial or example number based on manual (default value is 1). |

|           |  |
|-----------|--|
| method    | String specifying which optimisation protocol to use. Enter "SA" for Simulated Annealing or "RE" for Replica Exchange (default value is "SA").                       |
| file_name | String specifying file name of script (default value is example.R).  |
| mopac     | String specifying the path of the MOPAC (Molecular Orbital PACkage) executable for tutorial 3 (Geometry Optimisation of Vitamin C Molecule) (default value is NULL). |
| vignette  | If TRUE, add tutorial vignette to directory (default value is TRUE).   |

**Value**

A script (and optionally a vignette) for reproducing a tutorial.

**Examples**

```
out.dir <- tempdir()
OptimusExamples(dir=out.dir, example=1)
```

# Index

## \* datasets

ex.m.fun, 2

ex.r.fun, 2

ex.u.fun, 3

IJ\_ORIG, 3

ex.m.fun, 2

ex.r.fun, 2

ex.u.fun, 3

IJ\_ORIG, 3

Optimus, 4

OptimusExamples, 6