

Advanced User Manual

Nicholas Andre G. Johnson and Aleksandr B. Sahakyan

16 August 2018

This section will document all possible input arguments to Optimus and will describe the output format of Optimus. The code snippet below is the first part of the definition of Optimus which lists all input arguments. We will refer to input arguments without default values as mandatory input arguments and those with default values as optional input arguments (with the exception of K.INITIAL, which will be considered a mandatory input argument).

```
Optimus <- function(NUMITER      = 1000000,
                     STATWINDOW  = 70,
                     T.INI       = 0.00001,
                     T.ADJSTEP   = 0.000000005,
                     TSCLnum     = 2,
                     T.SCALING   = 3,
                     T.MIN       = 0.000000005,
                     T.DELTA     = 2,
                     DUMP.FREQ   = 10000,
                     LIVEPLOT    = TRUE,
                     LIVEPLOT.FREQ = 100000,
                     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"
                     ){...}
```

Mandatory Input Arguments

All mandatory input arguments have necessarily already been defined in the Tutorials. However, we will reiterate these definitions in this section. Optimus has four mandatory inputs, mDEF, uDEF, rDEF (referred to as $m()$, $u()$ and $r()$ respectively in the Tutorials) and K.INITIAL.

mDEF

mDEF must be of type closure. mDEF should be a function designed to operate on the whole set of parameter snapshot K and return the corresponding observable object O . Please note, that the size of K and O are not necessarily to match, depending on the nature of the model used. mDEF must necessarily take K and DATA as input variables, and it must necessarily operate on K to produce O (operating on DATA in optional, see Tutorial 3 for an illustration).

uDEF

uDEF must be of type closure. uDEF should be a function designed to evaluate the performance of a given snapshot of coefficients K . uDEF should necessarily take as inputs O (the output of mDEF) and the variable DATA. The output of uDEF should have two components, Q holding a single number of the quality of the K coefficients, and E holding a (pseudo)energy for the given snapshot K . It is important that the returned (pseudo)energy value is lower for better performance/version of K , never vice-versa. The Q component of the uDEF function output is only used for plotting the optimisation process, and, if desired, can just repeat the value of the E component.

rDEF

rDEF must be of type closure. rDEF should be a function that defines a rule by which the K coefficient vector is to be altered from one step to another. rDEF must accept K as an input and return an object equivalent to K , but with some alteration(s).

K.INITIAL

K.INITIAL is an object of any type which stores the initial values for the parameter(s) to be optimised. The only requirement for K is that it should be something alterable via rDEF and something that influences the outcome of mDEF.

Optional Input Arguments

NUMITER

NUMITER is a variable of type double that is the number of iterations (or steps) of the optimisation process per core. It has a default value of 1 000 000.

STATWINDOW

STATWINDOW is a variable of type double that is the number of iterations executed between subsequent temperature adjustments executed by the Temperature Control Unit (STATWINDOW is also the number of iterations used to calculate the observed acceptance ratio). It has a default value of 70.

T.INI

T.INI is a variable of type double that represents the initial system pseudo-temperature at the beginning of the optimisation procedure. It has a default value of 0.00001.

T.ADJSTEP

T.ADJSTEP is a variable of type double that represents the baseline temperature change step-size for temperature auto-adjustment. It has a default value of 0.000000005.

TSCLnum

TSCLnum is a variable of type double that indicates the maximum number of STATWINDOWS for which the observed acceptance ratio can sequentially be below or above the ideal acceptance ratio before T.ADJSTEP is scaled by T.SCALING. It has a default value of 2.

T.SCALING

T.SCALING is a variable of type double that represents the value by which T.ADJSTEP is scaled in accordance with the condition specified by TSCLnum. It has a default value of 3.

T.MIN

T.MIN is a variable of type double and is the value that the system pseudo-temperature is automatically set to if at any time, the Temperature Control Unit attempts to make the pseudo-temperature a negative value. It has a default value of 0.000000005.

T.DELTA

T.DELTA is a variable of type double. If after a STATWINDOW, the observed acceptance ratio is within T.DELTA of the ideal acceptance ratio, the Temperature Control Unit will make no change to the system pseudo-temperature. It has a default value of 2.

DUMP.FREQ

DUMP.FREQ is a variable of type double. It is the frequency in steps of printing the best found model to the working directory. It has a default value of 10 000.

LIVEPLOT

LIVEPLOT is a variable of type logical that indicates whether the optimisation process will be plotted in a pdf file in the working directory. It has a default value of TRUE.

LIVEPLOT.FREQ

LIVEPLOT.FREQ is a variable of type double that indicates the frequency in steps of printing the optimisation process in a pdf (this variable is only relevant if LIVEPLOT = TRUE). It has a default value of 100 000.

PDFheight

PDFheight is a variable of type double that indicates the height of the PDF that is produced (if LIVEPLOT = TRUE). It has a default value of 29.

PDFwidth

PDFwidth is a variable of type double that indicates the width of the PDF that is produced (if LIVEPLOT = TRUE). It has a default value of 20.

NCPU

NCPU is a variable of type double that indicates the number of Optimisation replicas to execute. If calling the Replica Exchange Version of Optimus, NCPU must be greater than 1. NCPU has a default value of 4.

LONG

LONG is a variable of type logical. If LONG = TRUE, a memory-friendly version of Optimus will be activated (in anticipation of a long simulation) and only data from the optimal explored parameter configuration and the last 10 000 optimisation iterations will be stored. LONG has a default value of TRUE.

SEED

SEED is a variable of type double which sets the seed for the random number generator. It has a default value of 840.

OPTNAME

OPTNAME is a variable of type character that can be thought of as the name of the optimisation process. OPTNAME is used when creating the file names of the Optimus output. OPTNAME = "" is the default value.

DATA

DATA is a variable of type list holding any additional data that must be accessed by mDEF and uDEF. The default value for DATA is NULL.

OPT.TYPE

OPT.TYPE is a variable of type character which specifies the mode of Optimus to execute and should always be equal to "SA" or "RE". If equal to "SA" (for Simulated Annealing), the Acceptance Ratio Annealing version of Optimus will be executed. If equal to "RE" (for Replica Exchange), the Replica Exchange version of Optimus will be executed. The default value of OPT.TYPE is "SA".

Acceptance Ratio Annealing Specific Optional Inputs

The below optional input arguments only impact the Acceptance Ratio Annealing mode of Optimus.

CYCLES

CYCLES is a variable of type double that specifies the number of annealing cycles to execute per core during the optimisation process. It has a default value of 10.

ACCRATIO.IN

ACCRATIO.IN is a variable of type double that specifies the initial target acceptance ratio for the annealing schedule in each annealing cycle. It has a default value of 90.

ACCRATIO.FIN

ACCRATIO.FIN is a variable of type double that specifies the final target acceptance ratio for the annealing schedule in each annealing cycle. It has a default value of 0.5.

Replica Exchange Specific Optional Inputs

The below optional input arguments only impact the Replica Exchange mode of Optimus.

EXCHANGE.FREQ

EXCHANGE.FREQ is a variable of type double that specifies the number of optimisation iterations to execute between subsequent parameter configuration exchanges between adjacent replicas. It has a default value of 1000.

ACCRATIO

ACCRATIO is a vector of doubles that must have length equal to the value of NCPU. ACCRATIO specifies the target acceptance ratio for each optimisation replica. It has a default value of c(90, 50, 5, 1).

Optimus Output

Optimus creates multiple output files in a user's working directory during an optimisation run. Each core used will generate 4 or 5 output files (depending on the value of LIVEPLOT):

- 1) [OPTNAME][Core number]_model_ALL
- 2) [OPTNAME][Core number]_model_K
- 3) [OPTNAME][Core number]_model_O
- 4) [OPTNAME][Core number]_model_QE
- 5) [OPTNAME][Core number]

Note that the 5th file is only produced if LIVEPLOT = TRUE. As an illustration, for the Acceptance Ratio Annealing run from Tutorial 1, Optimus generates 20 output files with the following names: poly_4_SA1_model_ALL, poly_4_SA1_model_K, poly_4_SA1_model_O, poly_4_SA1_model_QE, poly_4_SA1, poly_4_SA2_model_ALL, poly_4_SA2_model_K, poly_4_SA2_model_O, poly_4_SA2_model_QE, poly_4_SA2, poly_4_SA3_model_ALL, poly_4_SA3_model_K, poly_4_SA3_model_O, poly_4_SA3_model_QE, poly_4_SA3, poly_4_SA4_model_ALL, poly_4_SA4_model_K, poly_4_SA4_model_O, poly_4_SA4_model_QE and poly_4_SA4.

[OPTNAME][Core number]_model_ALL

This is the most important output file in that essentially all contents from the other output files is contained in this file. The *_model_ALL file is an R workspace that contains a variable OUTPUT of type list. For Acceptance Ratio Annealing, OUTPUT has 12 fields (numbered 1-12 below) while for Replica Exchange,

OUTPUT contains an additional 14 fields (numbered 13-26 below). Note that the additional fields of OUTPUT in Replica Exchange were included to facilitate the writing of the code; they can largely be ignored by the user.

- 1) K.stored
- 2) O.stored
- 3) STEP
- 4) PROB.VEC
- 5) T.DE.FACTO
- 6) IDEAL.ACC.VEC
- 7) ACC.VEC.DE.FACTO
- 8) STEP4ACC.VEC.DE.FACTO
- 9) ENERGY.DE.FACTO
- 10) Q.STRG
- 11) ACCEPTANCE
- 12) STEP.STORED
- 13) E.stored
- 14) E.old
- 15) Q.old
- 16) K
- 17) T
- 18) Step.stored
- 19) ENERGY.TRIAL.VEC
- 20) STEP.add
- 21) NumofAccRatSMideal
- 22) NumofAccRatGRideal
- 23) t.adjstep
- 24) AccR.category
- 25) new.T.INI
- 26) instanceOFswitch

K.stored holds the optimal parameter configuration found by the given processor. O.stored holds the object O generated by mDEF from the optimal parameter configuration K.stored. STEP is a double that holds the current optimisation iteration number. PROB.VEC is a vector that holds the acceptance probability for each optimisation step. T.DE.FACTO is a vector that holds the system pseudo-temperature during each optimisation iteration. IDEAL.ACC.VEC is a vector that holds the target acceptance ratio for each optimisation iteration in the case of Acceptance Ratio Annealing. In the case of Replica Exchange, IDEAL.ACC.VEC holds the same value as the input variable ACCRATIO. ACC.VEC.DE.FACTO is a vector that holds the observed acceptance ratio at the end of each STATWINDOW. STEP4ACC.VEC.DE.FACTO is a vector that holds the optimisation step numbers that correspond to the end of a STATWINDOW. ENERGY.DE.FACTO is a vector that holds the actual system energy E at each optimisation step. Q.STRG is a vector that holds the system quality Q at each optimisation step. ACCEPTANCE is a vector of binary variables whose i^{th} entry is 1 if the candidate parameter configuration was accepted on the i^{th} optimisation step and 0 otherwise. STEP.STORED is a vector storing the number of each optimisation step.

E.stored is a double that stores the energy E associated with the optimal parameter configuration K.stored. E.old is a double that stores the energy E associated with the most recently considered parameter configuration. Q.old is a double that stores the quality Q associated with the most recently considered parameter configuration. K hold the most recently considered parameter configuration. T is a double that holds the current system temperature. Step.stored is a double that holds the optimisation step number on which the optimal parameter configuration K.stored was discovered. ENERGY.TRIAL.VEC is a vector that holds the energy E of the candidate parameter configuration at every optimisation step. STEP.add is a double that facilitates indexing into output vectors. NumofAccRatSMideal is a double that represents the number of times the observed acceptance ratio has sequentially been smaller than the target acceptance ratio. NumofAccRatGRideal is a double that represents the number of times the observed acceptance ratio has been sequentially greater

than the target acceptance ratio. `t.adjuststep` is a double holding the current value by which the system pseudo-temperature is increased or decreased each `STATWINDOW` by the Temperature Control Unit. `AccR.category` is a character that indicates whether the observed acceptance ratio was above or below the target acceptance ratio during the previous `STATWINDOW`. `new.T.INI` is a double that stores an estimate for the ideal initial system pseudo-temperature deduced by the Temperature Control Unit. Finally, `instanceOFswitch` is a double that tracks the number of times the observed acceptance ratio has transitioned from being less than the target ratio to greater than the target ratio or vice versa.

`[OPTNAME][Core number]__model_K`

The `*__model_K` file is an R workspace that contains a variable `K.stored` of the same type as `K.INITIAL`. It holds the optimal parameter configuration found by the given processor.

`[OPTNAME][Core number]__model_O`

The `*__model_O` file is an R workspace that contains a variable `O.stored` that holds the object O generated by `mDEF` from the optimal parameter configuration `K.stored`.

`[OPTNAME][Core number]__model_QE`

The `*__model_QE` file is a text file that stores the values of E and Q that are produced from the optimal parameter configuration `K.stored` and, in the case of the Replica Exchange Version of Optimus, stores the target acceptance ratio associated with the given replica.

`[OPTNAME][Core number]`

The `*` file is a pdf file that includes 5 plots:

- 1) Acceptance probability as a function of optimisation iteration.
- 2) System psuedo-temperature as a function of optimisation iteration.
- 3) Observed (red solid line) and target (black dashed line) acceptance ratio as a function of optimisation iteration.
- 4) Energy E as a function of optimisation iteration.
- 5) Quality Q as a function of optimisation iteration.