

# Package ‘mplusParallel.automation’

September 20, 2023

**Title** Parallel Processing Automation for 'Mplus'

**Type** Package

**Version** 0.0.1.1

**Description** Offers automation tools to parallelize 'Mplus' operations when using 'R' for data generation. It facilitates streamlined integration between 'Mplus' and 'R', allowing users to run and manage multiple 'Mplus' models simultaneously and efficiently in 'R'.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** dplyr, furrr, future, parallel

**Depends** MplusAutomation

**Suggests** knitr, mvtnorm, rmarkdown

**NeedsCompilation** no

**Author** Jake Plantz [aut, cre]

**Maintainer** Jake Plantz <Jakewplantz@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-09-20 13:10:05 UTC

## R topics documented:

Base_AutomationFunc_single . . . . .	2
mplusParallel_automation . . . . .	2
removeParFolders . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

Base\_AutomationFunc\_single

*Print base automation functions for single or multi conditions*

---

### Description

This function is used to print the mplus\_automation code used in the mplusParallel\_automation function. This is done as convenience to ease custom function generation.

### Usage

```
Base_AutomationFunc_single()
```

### Value

Invisible NULL. The function is called for its side effect of deleting folders.

### See Also

[mplusParallel\\_automation](#) for the function that creates these folders.

---

mplusParallel\_automation

*Parallel automation of running Mplus models using R.*

---

### Description

This function provides a parallelized automation for Mplus when using R as the data generation method. When data\_gen is specified, include any arguments from the function that need to be set in the global environment.

### Usage

```
mplusParallel_automation(  
  k,  
  k.start = 1,  
  data_gen = NA,  
  seed = 123,  
  ncores = "default",  
  run = TRUE,  
  useCores = TRUE,  
  cores_per_analysis = "default",  
  Par_plan = "cluster",  
  rec = FALSE,  
  results = NULL,
```

```

multi_con = FALSE,
con_index = c(),
specific_sums = NULL,
specific_params = NULL,
item = NULL,
params_ext = c("unstandardized"),
modV1s = NULL,
ops = NULL,
modV2s = NULL,
custom_auto = NULL,
retry = TRUE,
max_retry = 5,
folder = ""
)

```

### Arguments

k	Number of replications desired.
k.start	Defaults to 1. Specifies the replication to start on. Useful if the simulation stopped on a specific replication and resuming without loss of work.
data_gen	Can take either a single dataframe in the 'folder' or a data generation function. When it is a data generation function any arguments for data generation should be specified in this function.
seed	Seed defaults to 123 but can be any integer. This ensures every replication's data is generated using a different seed but is reproducible.
ncores	Defaults to the number of cores on the machine - 1.
run	Logical. Defaults to T. When T the Mplus models will be run. When F models will not be run and the output files will be read in only.
useCores	Logical. When TRUE, the mplus files will be adjusted to use the number of cores on the machine. This can speed up simulation run times.
cores_per_analysis	Applies when useCores is TRUE. Default is ncores/2. If you experience issues or crashes due to memory or core use, set this lower. In testing the default will use most of a computer's CPU power but no break the simulation.
Par_plan	Plan for parallel processing. Defaults to 'cluster'. Can take any argument from the 'future' package
rec	Logical. Indicates if the files are in subdirectories.
results	Indicates which results to collect. Supports summaries, parameters, and modindices or any named list argument output by mplus automation. When using summaries, parameters, or modindices more specific output is available.
multi_con	Logical. Indicates whether multiple conditions are run in a singular instance. Default is F.
con_index	A character vector. Specifies the indices for conditions to be tracked.
specific_sums	Extracts specific output when results is 'summaries'.

<code>specific_params</code>	Extracts specific parameters when results is 'parameters'.
<code>item</code>	Extracts specific items when results is 'parameters'.
<code>params_ext</code>	When results is 'parameters', specifies parameter type for extraction. Can take any type but defaults to 'unstandardized'. If you do not desire unstandardized parameters read in an output file to determine the name of the parameters of interest and use this as the named argument.
<code>modV1s</code>	Used for specific output when results is 'mod_indicies'.
<code>ops</code>	Operator for modV1s, e.g., 'BY' for factor loadings.
<code>modV2s</code>	Second variable for modV1s.
<code>custom_auto</code>	User-defined function for running and reading in models. Only functions that return single dataframes each run are currently supported.
<code>retry</code>	Logical. Defaults to TRUE. Retries with a new seed if chi is not returned by the model.
<code>max_retry</code>	Defaults to 5. Specifies how many times a new seed should be attempted.
<code>folder</code>	User-defined path to the root folder of where your Mplus files are located.

**Value**

Function returns a dataframe of all the desired parameters for each replication.

**Examples**

```
## Not run:
# Loading the package
library(mplusParallel.automation)

# Data Generation
n_people <- 500
n_items <- 12
data <- matrix(sample(1:5, n_people * n_items, replace = TRUE), ncol = n_items)

# Writing an example input file
inp_content <- "
TITLE: TEST
DATA: FILE IS exdat.csv;
VARIABLE:
  Names ARE
  i1 i2 i3 i4 i5 i6 i7 i8 i9 i10 i11 i12;
USEVARIABLES ARE i1-i12;
ANALYSIS:
  TYPE = GENERAL;
  PROCESSORS=6;
OUTPUT:
  STANDARDIZED;
  MODINDICES (ALL);
MODEL:
  trait1 BY
  i1 (a1)
```

```
i2 (a2)
i3 (a3)
i4 (a4)
i5 (a5)
i6 (a6);
trait2 BY
i7 (a7)
i8 (a8)
i9 (a9)
i10 (a10)
i11 (a11)
i12 (a12);
i1-i12 (e);
trait1 @ 1
trait2 @ 1
"
writeLines(inp_content, "example_model_simple.inp")

# Running the function
res <- mplusParallel_automation(k=5, data_gen = data_gen,
results = 'parameters', specific_params = c('trait1.by', 'trait2.by'), folder = 'user_defined_path')

# Clean up
removeParFolders()

## End(Not run)
```

---

removeParFolders

*Remove Parallel Processing Folders from mplusParallel\_automation*

---

## Description

This function is used to delete all parallel processing folders (with names containing the word "session") that were created by the `mplusParallel_automation` function.

## Usage

```
removeParFolders(folder = "")
```

## Arguments

`folder`            The user-defined folder to search for parallel processing folders. Should be the same as the one used for `mplusParallel_automation`.

## Value

Invisible NULL. The function is called for its side effect of deleting folders.

**See Also**

[mplusParallel\\_automation](#) for the function that creates these folders.

**Examples**

```
# Assuming you have parallel processing folders in your current RStudio  
# document's directory  
removeParFolders(folder = 'user_defined_path')
```

# Index

Base\_AutomationFunc\_single, [2](#)  
mplusParallel\_automation, [2](#), [2](#), [6](#)  
removeParFolders, [5](#)