Package ‘mrgsim.parallel’

July 6, 2020

Type Package
Title Simulate with 'mrgsolve' in Parallel
Version 0.1.1
Author Kyle Baron
Maintainer Kyle Baron <kylebtwin@imap.cc>
Description Simulation from an 'mrgsolve'
<https://cran.r-project.org/package=mrgsolve> model using a parallel backend.
Input data sets are split (chunked) and simulated in parallel using
mclapply() or future_lapply()
<https://cran.r-project.org/package=future.apply>.
License GPL (>= 2)
Imports parallel, dplyr, future.apply
Depends mrgsolve, R (>= 3.5.0)
Suggests testthat
Encoding UTF-8
LazyData true
URL https://github.com/kylebaron/mrgsim.parallel
BugReports https://github.com/kylebaron/mrgsim.parallel/issues
RoxygenNote 7.1.1
Language en-US
NeedsCompilation no
Repository CRAN
Date/Publication 2020-07-06 15:30:02 UTC

R topics documented:

chunk_data_frame .................................................. 2
mrgsim.parallel .................................................... 2
parallel_mrgsim_d .................................................. 3
parallel_mrgsim_ei .................................................. 4
chunk_data_frame  Chunk a data frame

Description
Use chunk_by_id to split up a data set by the ID column; use chunk_by_row split a data set by rows.

Usage
chunk_by_id(data, nchunk, id_col = "ID", mark = NULL)
chunk_by_row(data, nchunk, mark = NULL)

Arguments
- data: a data frame
- nchunk: number of chunks
- id_col: character specifying the column containing the ID for chunking
- mark: when populated as a character label, adds a column to the chunked data frames with that name and with value the integer group number

Value
A list of data frames

Examples
x <- expand.grid(ID = 1:10, B = rev(1:10))
chunk_by_id(x, 3)
chunk_by_row(x, nchunk = 4)

mrgsim.parallel  Simulate with 'mrgsolve' in Parallel

Description
Simulate with 'mrgsolve' in Parallel

Resources
- GitHub: https://github.com/kylebaron/mrgsim.parallel
- Docs: https://kylebaron.github.io/mrgsim.parallel
**Package options**

- mrgsim.parallel mc.able: if TRUE, multicore will be used if appropriate.

---

**parallel_mrgsim_d**

*Simulate a data set in parallel*

**Description**

Use `future_mrgsim_d` to simulate with the `future` package. Use `mc_mrgsim_d` to simulate with `parallel::mclapply`.

**Usage**

```r
future_mrgsim_d(
  mod,
  data,
  nchunk = 4,
  ...
)
```

```r
mc_mrgsim_d(
  mod,
  data,
  nchunk = 4,
  ...
)
```

```r
fu_mrgsim_d(
  mod,
  data,
  nchunk = 4,
  ...
)
```
.parallel = TRUE
)

fu_mrgsim_d0(..., .dry = TRUE)

Arguments

mod mrgsolve model object see mrgsolve::mrgmod

data data set to simulate; see mrgsolve::data_set

nchunk number of chunks in which to split the data set

... passed to mrgsim_d

.as_list if TRUE a list is return; otherwise (default) a data frame

.p post processing function executed on the worker; arguments should be (1) the
simulated output (2) the model object

.dry if TRUE neither the simulation nor the post processing will be done

.seed passed to future_lapply as future.seed

.parallel if FALSE, the simulation will not be parallelized; this is intended for debugging
and testing use only

Value

A data frame or list of simulated data

See Also

future_mrgsim_ei

Examples

mod <- mrgsolve::house()

data <- mrgsolve::expand.ev(amt = seq(10))

out <- future_mrgsim_d(mod, data, nchunk = 2)
parallel_mrgsim_ei

Usage

future_mrgsim_ei(
  mod,
  events,
  idata,
  nchunk = 4,
  ...,
  .as_list = FALSE,
  .p = NULL,
  .dry = FALSE,
  .seed = TRUE,
  .parallel = TRUE
)

fu_mrgsim_ei(
  mod,
  events,
  idata,
  nchunk = 4,
  ...,
  .as_list = FALSE,
  .p = NULL,
  .dry = FALSE,
  .seed = TRUE,
  .parallel = TRUE
)

fu_mrgsim_ei0(..., .dry = TRUE)

mc_mrgsim_ei(
  mod,
  events,
  idata,
  nchunk = 4,
  ...,
  .as_list = FALSE,
  .p = NULL,
  .dry = FALSE,
  .seed = NULL,
  .parallel = TRUE
)

Arguments

mod        mrgsolve model object see mrgsolve::mrgmod
events     an event object from mrgsolve; see mrgsolve::ev
idata      an idata set of parameters, one per simulation unit (individual); see mrgsolve::idata_set
`parallel_mrgsim_ei`

- `nchunk`: number of chunks in which to split the data set
- `...`: passed to `mrgsim_d`
- `.as_list`: if TRUE a list is return; otherwise (default) a data frame
- `.p`: post processing function executed on the worker; arguments should be (1) the simulated output (2) the model object
- `.dry`: if TRUE neither the simulation nor the post processing will be done
- `.seed`: passed to `future_lapply` as `future.seed`
- `.parallel`: if FALSE, the simulation will not be parallelized; this is intended for debugging and testing use only

**Value**

A data frame or list of simulated data

**See Also**

`future_mrgsim_ei`

**Examples**

```r
mod <- mrgsolve::house()

events <- mrgsolve::ev(amt = 100)

idata <- data.frame(CL = runif(10, 0.5, 1.5))

out <- future_mrgsim_ei(mod, events, idata)
```
Index

chunk_by_id, 2
chunk_by_id(chunk_data_frame), 2
chunk_by_row, 2
chunk_by_row(chunk_data_frame), 2
chunk_data_frame, 2

fu_mrgsim_d(parallel_mrgsim_d), 3
fu_mrgsim_d0(parallel_mrgsim_d), 3
fu_mrgsim_ei(parallel_mrgsim_ei), 4
fu_mrgsim_ei0(parallel_mrgsim_ei), 4
future_lapply, 4, 6
future_mrgsim_d, 3
future_mrgsim_d(parallel_mrgsim_d), 3
future_mrgsim_ei, 4, 6
future_mrgsim_ei(parallel_mrgsim_ei), 4

mc_mrgsim_d, 3
mc_mrgsim_d(parallel_mrgsim_d), 3
mc_mrgsim_ei, 4
mc_mrgsim_ei(parallel_mrgsim_ei), 4
mrgsim.parallel, 2
mrgsim_d, 4, 6
mrgsolve::data_set, 4
mrgsolve::ev, 5
mrgsolve::idata_set, 5
mrgsolve::mrgmod, 4, 5

parallel_mrgsim_d, 3
parallel_mrgsim_ei, 4