Package 'circumstance'

July 17, 2023

R topics documented:

`circumstance-package` *circumstance package*

Description

circumstance provides tools for parallelizing certain **pomp** calculations.

`continue` *Continue an iterative calculation*

Description

Continue an iterative computation where it left off.

Usage

```
## S4 method for signature 'mif2List'
continue(object, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pmcmcList'
continue(object, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)
```

Arguments

object	the result of an iterative **pomp** computation
...	additional arguments will be passed to the underlying method. This allows one to modify parameters used in the original computations.
seed	seed for the parallel random-number generator. Setting seed=TRUE passes the current seed to the parallel RNG. See %dofuture% for more information.
chunk.size	average number of elements processed per chunk. See %dofuture% for more information.
scheduling	average number of chunks that each worker processes. See %dofuture% for more information. This is ignored unless chunk.size=NULL.

See Also

mif2

---

mif2 *Parallel iterated filtering*

Description

Runs multiple instances of mif2 using foreach.

Usage

```
## S4 method for signature 'ANY,data.frame'
mif2(data, starts, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'ANY,missing'
mif2(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pompList,missing'
mif2(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pfilterList,missing'
mif2(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'mif2List,missing'
mif2(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)
```

Arguments

data	passed to pomp::mif2
starts	data frame containing parameters at which to begin iterated filtering
...	all additional arguments are passed to pomp::mif2

seed	seed for the parallel random-number generator. Setting seed=TRUE passes the current seed to the parallel RNG. See %dofuture% for more information.
chunk.size	average number of elements processed per chunk. See %dofuture% for more information.
scheduling	average number of chunks that each worker processes. See %dofuture% for more information. This is ignored unless chunk.size=NULL.

See Also

pomp::mif2.

---

pfilter *Parallel particle filter computations*

Description

Runs multiple instances of pfilter using foreach.

Usage

```
## S4 method for signature 'ANY,numeric'
pfilter(data, Nrep, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'ANY,missing'
pfilter(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pompList,numeric'
pfilter(data, Nrep, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pompList,missing'
pfilter(data, Nrep, ...)
```

Arguments

data	passed to pomp::pfilter
Nrep	number of replicate particle filter computations to run. By default, Nrep = 1.
...	all additional arguments are passed to pomp::pfilter
seed	seed for the parallel random-number generator. Setting seed=TRUE passes the current seed to the parallel RNG. See %dofuture% for more information.
chunk.size	average number of elements processed per chunk. See %dofuture% for more information.
scheduling	average number of chunks that each worker processes. See %dofuture% for more information. This is ignored unless chunk.size=NULL.

See Also

pomp::pfilter.

Examples

```
library(circumstance)
library(doFuture)

ou2() -> ou2

plan(sequential)
system.time(ou2 |> pfilter(Np=10000,Nrep=6) -> pfs)

plan(multicore)
system.time(ou2 |> pfilter(Np=10000,Nrep=6) -> pfs)
```

---

plot_matrix *A scatterplot matrix with densities on the diagonal.*

Description

A special scatterplot matrix.

Usage

```
plot_matrix(data, ...)

## S3 method for class 'list'
plot_matrix(
  data,
  marg.exp = 0.02,
  labels = names(data),
  alpha = 1,
  pch = 16,
  size = unit(0.03, "npc"),
  ...
)

## S3 method for class 'data.frame'
plot_matrix(
  data,
  marg.exp = 0.02,
  labels = names(data),
  alpha = 1,
  pch = 16,
  size = unit(0.03, "npc"),
  ...
```

```
)

## S3 method for class 'plotmatrix'
print(x, newpage = is.null(vp), vp = NULL, ...)
```

Arguments

`data`	Data to plot.
`...`	optional arguments, passed to `hist`.
`marg.exp`	Fraction by which to expand the plot at the margins.
`labels`	Names of variables plotted.
`alpha, pch, size`	Refer to the plotted points in the scatterplots.
`x`	`plot_matrix` object to display.
`newpage`	logical; if `TRUE`, `grid.newpage()` will be called before the graphics are drawn.
`vp`	viewport to use. See `viewport`.

Examples

```
# requires dplyr
library(dplyr)

data.frame(
  a=rexp(n=1000,rate=1/3),
  b=rnorm(1000)
) |>
  mutate(
    c=a+b^2,
    d=a-b^3
  ) -> x

print(plot_matrix(x,alpha=0.2))

g <- plot_matrix(
  x[-2],
  labels=c(
    expression(alpha),
    expression(beta),
    expression(phi)
  ),
  alpha=0.3
)
print(g)

print(plot_matrix(as.list(x),alpha=0.2,breaks="scott"))
```

pmcmc *Particle Markov chain Monte Carlo in parallel*

Description

Runs multiple instances of pmcmc using foreach.

Usage

```
## S4 method for signature 'ANY,data.frame'
pmcmc(data, starts, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'ANY,missing'
pmcmc(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pompList,missing'
pmcmc(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pfilterList,missing'
pmcmc(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

## S4 method for signature 'pmcmcList,missing'
pmcmc(data, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)
```

Arguments

data	passed to pomp::pmcmc
starts	data frame containing parameters at which to begin iterated filtering
...	all additional arguments are passed to pomp::pmcmc
seed	seed for the parallel random-number generator. Setting seed=TRUE passes the current seed to the parallel RNG. See %dofuture% for more information.
chunk.size	average number of elements processed per chunk. See %dofuture% for more information.
scheduling	average number of chunks that each worker processes. See %dofuture% for more information. This is ignored unless chunk.size=NULL.

See Also

pomp::pmcmc.

Index