

Package ‘circumstance’

April 25, 2024

Contents

circumstance-package	1
continue	2
mif2	2
pfilter	3
plot_matrix	4
pmcmc	6

Index	8
--------------	----------

circumstance-package	<i>circumstance package</i>
----------------------	-----------------------------

Description

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

Author(s)

Maintainer: Aaron A. King <kingaa@umich.edu>

See Also

Useful links:

- <https://kingaa.github.io/circumstance/>
- Report bugs at <https://github.com/kingaa/circumstance/issues/>

continue	<i>Continue an iterative calculation</i>
----------	--

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	<i>Parallel iterated filtering</i>
------	------------------------------------

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 'pomplList,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 <code>pomp::mif2</code>
starts	data frame containing parameters at which to begin iterated filtering
...	all additional arguments are passed to <code>pomp::mif2</code>
seed	seed for the parallel random-number generator. Setting seed=TRUE passes the current seed to the parallel RNG. See <code>%dofuture%</code> for more information.
chunk.size	average number of elements processed per chunk. See <code>%dofuture%</code> for more information.
scheduling	average number of chunks that each worker processes. See <code>%dofuture%</code> 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 'pomplList,numeric'
pfilter(data, Nrep, ..., seed = TRUE, chunk.size = NULL, scheduling = 1)

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

```

Arguments

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

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

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

See Also

`pomp : pmcmc`.

Index

`%dofuture%`, [2–4](#), [7](#)

`circumstance` (`circumstance-package`), [1](#)
`circumstance`, `package`
 (`circumstance-package`), [1](#)
`circumstance-package`, [1](#)
`continue`, [2](#)
`continue`, `mif2List`-method (`continue`), [2](#)
`continue`, `pmcmcList`-method (`continue`), [2](#)

`hist`, [5](#)

`mif2`, [2](#), [2](#)
`mif2`, `ANY`, `data.frame`-method (`mif2`), [2](#)
`mif2`, `ANY`, `missing`-method (`mif2`), [2](#)
`mif2`, `mif2List`, `missing`-method (`mif2`), [2](#)
`mif2`, `pfilterList`, `missing`-method (`mif2`),
 [2](#)
`mif2`, `pomplList`, `missing`-method (`mif2`), [2](#)

`pfilter`, [3](#)
`pfilter`, `ANY`, `missing`-method (`pfilter`), [3](#)
`pfilter`, `ANY`, `numeric`-method (`pfilter`), [3](#)
`pfilter`, `pomplList`, `missing`-method
 (`pfilter`), [3](#)
`pfilter`, `pomplList`, `numeric`-method
 (`pfilter`), [3](#)

`plot_matrix`, [4](#)
`pmcmc`, [6](#)
`pmcmc`, `ANY`, `data.frame`-method (`pmcmc`), [6](#)
`pmcmc`, `ANY`, `missing`-method (`pmcmc`), [6](#)
`pmcmc`, `pfilterList`, `missing`-method
 (`pmcmc`), [6](#)
`pmcmc`, `pmcmcList`, `missing`-method (`pmcmc`),
 [6](#)
`pmcmc`, `pomplList`, `missing`-method (`pmcmc`), [6](#)
`pomp`: `:mif2`, [3](#)
`pomp`: `:pfilter`, [4](#)
`pomp`: `:pmcmc`, [7](#)
`print.plotmatrix` (`plot_matrix`), [4](#)

`viewport`, [5](#)