Package 'circumstance'

January 26, 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

```
continue(object, ...)

## S4 method for signature 'mif2List'
continue(object, ...)
```

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.

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, ...)

## S4 method for signature 'ANY,missing'
mif2(data, starts, ...)

## S4 method for signature 'pompList,missing'
mif2(data, starts, ...)

## S4 method for signature 'pfilterList,missing'
mif2(data, starts, ...)

## S4 method for signature 'mif2List,missing'
mif2(data, starts, ...)
```

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](#)

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, ...)

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

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

## 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

See Also

pomp::pfilter.

Examples

```
library(circumstance)
library(doFuture)
library(doRNG)
registerDoFuture()
registerDoRNG()

ou2() -> ou2

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

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

Index