Package 'ParDNAcopy'

January 20, 2025

Type Package

Title Parallel implementation of the ``segment'' function of package ``DNAcopy''

Version 2.0

Date 2014-01-01

Author Alex Krasnitz, Guoli Sun

Maintainer Guoli Sun <guolisun87@gmail.com>

Description Parallelized version of the ``segment'' function from Bioconductor package ``DNAcopy'', utilizing multi-core computation on host CPU

License GPL-2

Depends DNAcopy, parallel

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-30 07:53:34

Contents

parSegment 1

Index 3

`parSegment` *Parallel implementation of* `segment` *function of* `DNAcopy`

Description

There are three key differences between this function and the original `segment` function of package `DNAcopy`. First, the execution can be parallelized, either by using multiple cores of the present host or by invoking a grid engine to run on multiple hosts. Secondly, random number generator may be re-initialized, with the same seed, for each sample. Finally, there is a "skinny" option for the value, i.e., a `DNAcopy` object with no `data` item.

Usage

```
parSegment(CNAobj, ranseed = NULL, distrib = c("vanilla", "Rparallel"),
njobs = 1, out = c("full", "skinny"), ...)
```

Arguments

CNAobj	An object of class CNA, usually a value produced by the CNA function of DNAcopy
ranseed	A single integer to seed the random number generator.
distrib	One of "vanilla" (default) and "Rparallel" to choose a parallelization option: no parallelization ("vanilla"), parallelization on multiple cores of the local host ("Rparallel").
njobs	An integer specifying the desired number of parallel jobs.
out	One of "full" (default) or "skinny" to specify the form of the value, an object of class DNAcopy, with the data item present ("full") or not ("skinny").
...	Arguments other than x to be passed on to the segment function of DNAcopy.

Value

An object of class DNAcopy. If out == "skinny" the data item of the value will not be returned in order to reduce the memory use.

Author(s)

Alex Krasnitz

See Also

Package DNAcopy.

Examples

```
data(coriell)
#prepare data for segmentation
CNA.object <- CNA(genomdat=coriell[,c(4,5)],coriell$Chromosome,coriell$Position,
data.type="logratio",sampleid=dimnames(coriell)[[2]][4:5])
#equivalent to "segment" of DNAcopy
parseg<-parSegment(CNA.object,undo.splits="sdundo")
#Random number generator to be re-seeded for each sample
parsegrep<-parSegment(CNA.object,ranseed=123,undo.splits="sdundo")
#multi-core execution but the result should not change
parsegrep1<-parSegment(CNA.object,ranseed=123,distrib="Rparallel",njobs=2,
undo.splits="sdundo")
```

Index

parSegment, 1