# Package 'NGScopy'

April 12, 2018

**Type** Package

**Version** 1.12.0

**Date** 2015-10-07 11:40:39 EDT

**Title** NGScopy: Detection of Copy Number Variations in Next Generation Sequencing sequencing

**Description** NGScopy provides a quantitative caller for detecting copy number variations in next generation sequencing (NGS), including whole genome sequencing (WGS), whole exome sequencing (WES) and targeted panel sequencing (TPS). The caller can be parallelized by chromosomes to use multiple processors/cores on one computer.

**License** GPL (>=2)

**LazyLoad** TRUE

**NeedsCompilation** no

**Repository** Bioconductor

**Depends** R (>= 3.1.0)

**Imports** methods, parallel, Xmisc (>= 0.2.1), rbamtools (>= 2.6.0), changepoint (>= 2.1.1)

**Suggests** RUnit, NGScopyData, GenomicRanges

**Collate** 'NGScopy-package.R' 'interface-to-cran-changepoint.R' 'ngscopy-internal.R' 'ngscopy.R' 'ngscopy-test.R' 'archive.R'

**Maintainer** Xiaobei Zhao <xiaobei@binf.ku.dk>

**biocViews** CopyNumberVariation, DNASeq, TargetedResequencing, ExomeSeq, WholeGenome, Sequencing

**Author** Xiaobei Zhao [aut, cre, cph]

## R topics documented:

---

`NGScopy-package` *NGScopy: Detection of copy number variations in next generation sequencing*

---

## Description

NGScopy provides a quantitative caller for detecting copy number variations in next generation sequencing (NGS), including whole genome sequencing (WGS), whole exome sequencing (WES) and targeted panel sequencing (TPS). The caller can be parallelized by chromosomes to use multiple processors/cores on one computer.

## Author(s)

Xiaobei Zhao

## References

Zhao et al (2014), Targeted Sequencing in Non-Small Cell Lung Cancer (NSCLC) Using the University of North Carolina (UNC) Sequencing Assay Captures Most Previously Described Genetic Aberrations in NSCLC, In preparation

---

`df.to.gr` *Convert a data.frame to a GRanges object*

---

## Description

Convert a data.frame to a GRanges object

## Usage

```
df.to.gr(x, which.chr = "chr", which.start = "start", which.end = "end",
  which.width = "width", which.name = "name", chrlength = NULL,
  start0 = TRUE)
```

## Arguments

| | |
|---|---|
| `x` | data.frame or matrix |
| `which.chr` | character, the column name of ‘chr‘ |
| `which.start` | character, the column name of ‘start‘ |
| `which.end` | character, the column name of ‘end‘ |
| `which.width` | character, the column name of ‘width‘ |
| `which.name` | character, the column name of ‘name‘ |
| `chrlength` | numeric, the lengths of the chromosomes |
| `start0` | logical, wheter the ‘start‘ is 0-based. |

## Value

a GRanges object

## Author(s)

Xiaobei Zhao

## Examples

```
## Not run:
x <- data.frame(
  chr=c('chr1','chr2'),start=0,end=100,
  name=paste0('ID',1:2),score=1:2
  )
df.to.gr(x)
df.to.gr(x,chrlength=c(chr1=1000,chr2=1200))

## End(Not run)
```

---

`help_segmtype` *Get help for segmentation functions*

---

## Description

Get help for segmentation functions given the type of segmentation.

## Usage

```
help_segmtype(segtype)
```

## Arguments

| | |
|---|---|
| `segtype` | character, the type of segmentation. See `parse_segmtype()` |

## Author(s)

Xiaobei Zhao

## Examples

```
## Not run:
help_segmtype(parse_segmtype()[1])

## End(Not run)
```

NGScopy-class *Detection of copy number variations in next generation sequencing (NGS)*

## Description

NGScopy is a reference class to detect copy number variations by "restriction-imposed windowing" in next generation sequencing (NGS).

## Details

Detection of copy number variations in next generation sequencing (NGS)

## Fields

`inFpathN` character, the file path to the control (normal) sample

`inFpathT` character, the file path to the case (tumor) sample

`outFpre` character, the file path of the directory for output.

`libsizeN` numeric, the library size of the control (normal) sample.

`libsizeT` numeric, the library size of the case (tumor) sample.

`mindepth` numeric, the minimal depth of reads per window.

`minsize` numeric, the minimal size of a window.

`regions` data.frame of three columns (chr/start/end), the regions to study. It follows the BED format: zero-based, half-open; (start,end].

`segtype` character, the type of change to capture during segmentation, mean and/or variance, normal or nonparametric distributions. A character vector with a single or multiple values from c("mean.norm","meanvar.norm","mean.cusum","var.css"). see `changepoint`.

`dsN` integer, the downsampling factor of the control (normal) sample.

`dsT` integer, the downsampling factor of the case (tumor) sample.

`MoreArgs.cn` list, a list of arguments for method 'calc_cn'. See `set_MoreArgs.cn`.

`MoreArgs.segm` list, a list of arguments for method 'calc_segm'. See `set_MoreArgs.segm`.

`pcThreads` integer, the number of processors performing the parallel computing.

`auto.save` logical, whether to save (any completed results) automatically.

`auto.load` logical, whether to load (any previously completed results) automatically.

`force.rerun` character, the names of methods to rerun regardless of any previous runs, default to c().

`out` list, the output.

## Methods

`calc_cn()` Calculate the relative copy number ratios (CNRs) per window, using the depth in the control (normal) sample as denominator and the case (tumor) sample as numerator.

`calc_segm()` Calculate the segment and detect the change points.

`get_cn()` Get the copy number object.

`get_cnr()` Get the copy number ratios.

`get_data.cn(as.granges = FALSE)` Get the data.frame of copy number.

`get_dataN()` Get the data of the normal per window.

`get_data.segm(as.granges = FALSE)` Get the data.frame of segmentation.

`get_dataT()` Get the data of the normal per window.

`get_depthN()` Get the depth of the normal per window.

`get_depthT()` Get the depth of the tumor per window.

`get_dsN()` Get dsN

`get_dsT()` Get dsT

`get_inFpathN()` Get inFpathN

`get_inFpathT()` Get inFpathT

`get_libsizeN()` Get libsizeN

`get_libsizeT()` Get libsizeT

`get_mindepth()` Get mindepth

`get_minsize()` Get minsize

`get_MoreArgs.cn()` Get MoreArgs.cn

`get_MoreArgs.segm()` Get MoreArgs.segm

`get_outFpre()` Get outFpre

`get_pcThreads()` Get pcThreads

`get_pos()` Get the position (midpoint) per window.

`get_reflength()` Get reference genome length in the normal sample.

`get_refname()` Get reference genome name in the normal sample.

`get_regions()` Get regions.

`get_segm()` Get the segmentation object.

`get_segmtype()` Get segtype, segmentation type(s).

`get_size()` Get the size per window.

`get_windows()` Get the windows.

`loadme()` Load a previously saved output.

`load_normal(normalDpath)` Load a previously saved output of the normal. normalDpath: the path to the .RData file for the output of the normal.

`plot_out(pdfFpath, width, height, scales, xlim, ylim, xlab, ylab, ..., MoreArgs.plotcn, MoreArgs`
Plot the output and save to a pdf file. pdfFpath: a file path (relative to outFpre) for the pdf output; width,height: see 'grDevices::pdf'; scales: are scales shared across all chromossomes (i.e. x coordinates reflect the range of genomic coordinates per chromosome, y coordinates reflect the range of of CNRs per chromosome), given no specific 'xlim' and 'ylim' (the default, "fixed"), or do they vary across x coordinates ("free_x"), y coordinates ("free_y"), or both ("free"); xlim,ylim,xlab,ylab,...: see 'graphics::plot'; MoreArgs.plotcn: additional arguments as in 'graphics::points'; MoreArgs.plotseg: additional arguments as in 'graphics::segments'.

`proc_cn()` Process the output of coy number object and return as a data.frame.

`proc_normal()` Process the normal sample: make the windows and count the reads per window.

`proc_segm()` Process the output of segmentation object and return as a data.frame.

`proc_tumor()` Process the tumor sample: count the reads per window.

`save_normal()` Get the output of the normal for later usage.

`set_ds(dsN, dsT)` Set downsampling factors. See 'set_dsN', 'set_dsT'.

`set_dsN(dsN)` Set downsampling factor of the control (normal). dsN: numeric, the library size of the control (normal) sample.

`set_dsT(dsT)` Set downsampling factor of the case (tumor). dsT: numeric, the library size of the case (tumor) sample.

`set_force.rerun(force.rerun)` Set force.rerun. Reset it with missing input.

`set_inFpathN(inFpathN)` Set a control (normal) sample. inFpathN: The file path to the control (normal) sample.

`set_inFpathT(inFpathT)` Set a case (tumor) sample. inFpathT: The file path to the case (tumor) sample.

`set_libsize(libsizeN, libsizeT)` Set library sizes. See 'set_libsizeN', 'set_libsizeT'.

`set_libsizeN(libsizeN)` Set library size of the control (normal). libsizeN: numeric, the library size of the control (normal) sample.

`set_libsizeT(libsizeT)` Set library size of the case (tumor). libsizeT: numeric, the library size of the case (tumor) sample.

`set_mindepth(mindepth)` Set the minimal depth per window. mindepth: the minimal depth of reads per window in the control (normal) sample.

`set_minsize(minsize)` Set the minimal size per window. minsize: the minimal size of a window in the control (normal) sample.

`set_MoreArgs.cn(...)` Set MoreArgs.cn. ..., (pseudocount: the pseudocounts added to the observed depth per window, default to 1; logr: logical, whether to make log2 transformation of the ratios, default to TRUE.)

`set_MoreArgs.segm(...)` Set MoreArgs.segm. ..., a list of other arguments to the funciton of segmentation given by segtype. See 'help_segmtype'.

`set_normal(inFpathN, mindepth, minsize)` Set a control (normal) sample and minimal depth/size per window. See 'set_inFpathN', 'set_mindepth', 'set_minsize'.

`set_outFpre(outFpre)` Set a directory for output. outFpre: the file path of the directory for output.

`set_pcThreads(pcThreads)` Set the number of processors. pcThreads: numeric, the number of processors performing the parallel computing. It should not exceed the system's capacity.

`set_regions(regions)` Set regions. regions: the regions in study, matrix, data.frame, character, file or connection with three columns (chr/start/end).

`set_segmtype(segtype)` Set the type of segmentation. segtype: a character vector with a single or multiple values from c("mean.norm","meanvar.norm","mean.cusum","var.css"). See 'changepoint'.

`set_tumor(inFpathT)` Set a case (tumor) sample. See 'set_inFpathT'.

`write_cn(cnFpath, ...)` Write the output of copy numbers as a data.frame to a tab separated file. cnFpath: a file path (relative to outFpre) for 'cn' output; ...: see 'Xmisc::write.data.table'.

`write_segm(segFpath, ...)` Write the output of segments as a data.frame to a tab separated file. segFpath: a file path (relative to outFpre) for 'seg' output; ...: see 'Xmisc::write.data.table'.

## Author(s)

Xiaobei Zhao

## See Also

`NGScopyData` ##

## Examples

```
require(NGScopy)
require(NGScopyData)

## Create an instance of `NGScopy' class
obj <- NGScopy$new(
  outFpre="ngscopy-case1",         # specified directory for output
  inFpathN=tps_N8.chr6()$bamFpath, # normal sample: tps_90.chr6.sort.bam
  inFpathT=tps_90.chr6()$bamFpath, # tumor sample:  tps_N8.chr6.sort.bam
  libsizeN=5777087,                # the library size of the normal sample
  libsizeT=4624267,                # the library size of the tumor sample
  mindepth=20,                     # the minimal depth of reads per window
  minsize=20000,                   # the minimal size of a window
  pcThreads=1                      # the number of processors for computing
  )

obj$show()                         # print the instance

## Not run:

## Compute the copy number and save it
## A data.frame will be saved to file `ngscopy_cn.txt' in the output directory
obj$write_cn()

## Compute the segmentation and save it
## A data.frame will be saved to file `ngscopy_segm.txt' in the output directory
obj$write_segm()

## Visualization
## A figure will be saved to file `ngscopy_out.pdf' in the output directory
obj$plot_out()

## End(Not run)
```

---

# ngscopy_cmdline_example

*Run an example of NGScopy at UNIX-like command line*

---

## Description

Run an example of NGScopy at UNIX-like command line

## Usage

```
ngscopy_cmdline_example(ifRun = FALSE, pcThreads = 1)
```

## Arguments

| | |
|---|---|
| ifRun | logical, whether to run |
| pcThreads | numeric, the number of processors performing the parallel computing. |

## Value

character, the command line.

### Author(s)

Xiaobei Zhao

### Examples

```
## To run at R prompt
ngscopy_cmdline_example()
## Not run:
ngscopy_cmdline_example(ifRun=TRUE) # Note: this would take a while. And
                                    # R will create a folder for output
                                    # at current working directory

## End(Not run)

## ## To run at Unix-like command line (not at R prompt)
## Rscript -e "require(methods);NGScopy::ngscopy_cmdline_example(TRUE)"
```

---

## ngscopy_unittest *A wrapper to run unit testing of NGScopy*

---

### Description

A wrapper to run unit testing of NGScopy

### Usage

```
ngscopy_unittest(...)
```

### Arguments

| | |
|---|---|
| `...` | additional arguments to `UnitTest$new` |

### Author(s)

Xiaobei Zhao

### Examples

```
## To run at R prompt
## Not run:
ngscopy_unittest() # Note: R will create a folder for output
                   # at current working directory

## End(Not run)

## ## To run at Unix-like command line (not at R prompt)
## Rscript -e "require(methods);NGScopy::ngscopy_unittest()"
```

## parse_segmtype *Parse the type of segmentation*

### Description

Parse the type of segmentation or return all available types with missing input.

### Usage

```
parse_segmtype(segtype)
```

### Arguments

| | |
|---|---|
| segtype | character, the type of segmentation. Return all available types if missing. |

### Value

the type of segmentation

### Author(s)

Xiaobei Zhao

### Examples

```
parse_segmtype()
```

## read_regions *Read regions from a data.frame, a file or a connection.*

### Description

Read regions from a data.frame, a file or a connection.

### Usage

```
read_regions(x)
```

### Arguments

| | |
|---|---|
| x | data.frame or character, the regions in study. A data.frame with three columns (chr/start/end), a file path or a connection with three columns (chr/start/end) without column names. Return an empty data.frame if "", "NULL" or NULL. |

### Value

data.frame of three columns (chr/start/end)

### Author(s)

Xiaobei Zhao

## Examples

```
read_regions("
chr1 0 249250621
chr2 0 243199373
chr3 0 198022430
chr4 0 191154276
chr5 0 180915260
chr6 0 171115067
")
```

# Index