Package 'oncoscanR'

April 8, 2026

Type Package

Title Secondary analyses of CNV data (HRD and more)

Version 1.13.0

Description The software uses the copy number segments from a text file and identifies all chromosome arms that are globally altered and computes various genome-wide scores. The following HRD scores (characteristic of BRCA-mutated cancers) are included: LST, HR-LOH, nLST and gLOH. the package is tailored for the ThermoFisher Oncoscan assay analyzed with their Chromosome Alteration Suite (ChAS) but can be adapted to any input.

License MIT + file LICENSE

Date 2025-06-27

Encoding UTF-8

VignetteBuilder knitr

biocViews CopyNumberVariation, Microarray, Software

Imports readr, S4Vectors, methods, utils

RoxygenNote 7.3.2

Depends R (>= 4.2), IRanges (>= 2.30.0), GenomicRanges (>= 1.48.0), magrittr

Suggests testthat (>= 3.1.4), jsonlite, knitr, rmarkdown, BiocStyle

URL https://github.com/yannchristinat/oncoscanR

BugReports https://github.com/yannchristinat/oncoscanR/issues

LazyData true

git_url https://git.bioconductor.org/packages/oncoscanR

git_branch devel

git_last_commit 1b56ce7

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-04-07

Author Yann Christinat [aut, cre],
Geneva University Hospitals [aut, cph]

Maintainer Yann Christinat <yann.christinat@hcuge.ch>

Contents

`oncoscanR-package` *oncoscanR: an R package to perform secondary analysis of Copy Number Variation data.*

Description

Allows computation of different homologous recombination deficiency (HRD) scores to identify PARP inhibitors responders. The package also allows for computation of the tandem duplication plus score (TDplus; hallmark of CDK12-mutated tumors) and the identification of arm-level alterations (e.g. gain of chromosome arm 1p).

Details

The package is tailored for the ThermoFisher Oncoscan assay analyzed with their Chromosome Alteration Suite (ChAS) but can be adapted to any input.

To learn more about oncoscanR, see the vignette using `browseVignettes(package = "oncoscanR")`.

Author(s)

Maintainer: Yann Christinat <yann.christinat@hcuge.ch>

Authors:

- Geneva University Hospitals [copyright holder]

See Also

Useful links:

- https://github.com/yannchristinat/oncoscanR
- Report bugs at https://github.com/yannchristinat/oncoscanR/issues

---

`adjust_loh` *Trim LOH segments with respect to loss segments.*

Description

Trim LOH segments with respect to loss segments.

Usage

```
adjust_loh(segments)
```

Arguments

`segments`	A `GRanges` object containing the segments, their copy number and copy number types.

Details

LOH segments completely contained within (or equal to) a copy loss segment are deleted. LOH segments partially overlapping (on one end only) with a copy loss segment are trimmed to remove the overlap or split into several segments.

Value

A `GRanges` object containing the cleaned segments, their copy number and copy number types.

Examples

```
segs.adj <- adjust_loh(segs.chas_example)
```

`armlevel_alt` *Get all globally-altered chromosome arms.*

Description

Get all globally-altered chromosome arms.

Usage

```
armlevel_alt(segments, kit.coverage, threshold = 0.9)
```

Arguments

`segments`	A `GRanges` object containing the segments.
`kit.coverage`	A `GRanges` object containing the regions covered on each chromosome arm.
`threshold`	The minimum percentage of the arm to be considered as globally altered. Defaults to 80%.

Details

By default uses the sum of all alterations and set the arm as globally altered if \>80% of the arm is altered. Does not account for alteration type and copy number. Will run the function `trim_to_coverage` on the segments.

Value

A list of globally-altered chromosome arms with the percentage of arm altered.

Examples

```
arms <- armlevel_alt(segs.chas_example, oncoscan_na33.cov, 0.9)
```

`cntypes` *Accepted types of CN for the segments - 'Gain': 1-2 extra copies - 'Weak amplification': 3-7 extra copies - 'Strong amplification': 8 or more extra copies - 'Heterozygote loss': Loss of one copy out of two - 'Homozygote loss': Loss of all copies - 'LOH': copy-neutral loss of one parental allele*

Description

Accepted types of CN for the segments - 'Gain': 1-2 extra copies - 'Weak amplification': 3-7 extra copies - 'Strong amplification': 8 or more extra copies - 'Heterozygote loss': Loss of one copy out of two - 'Homozygote loss': Loss of all copies - 'LOH': copy-neutral loss of one parental allele

Usage

```
cntypes
```

Format

A list of strings containing all CN types

Source

cntypes <- list(LOH='LOH', Gain='Gain', Loss='Loss')

---

get_amp_segments *Return all segments with an amplification (5 or more copies).*

Description

Return all segments with an amplification (5 or more copies).

Usage

```
get_amp_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
segs.amp <- get_amp_segments(segs.chas_example)
```

get_gain_segments *Return all segments with gain of copies.*

Description

Return all segments with gain of copies.

Usage

```
get_gain_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
segs.gain <- get_gain_segments(segs.chas_example)
```

get_hetloss_segments *Return all segments with heterozygous loss.*

Description

Return all segments with heterozygous loss.

Usage

```
get_hetloss_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
segs.hetloss <- get_hetloss_segments(segs.chas_example)
```

get_homloss_segments *Return all segments with homozygous loss.*

Description

Return all segments with homozygous loss.

Usage

```
get_homloss_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
get_homloss_segments <- get_homloss_segments(segs.chas_example)
```

get_loh_segments *Return all segments of type LOH, independently of the copy number.*

Description

Return all segments of type LOH, independently of the copy number.

Usage

```
get_loh_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
segs.loh <- get_loh_segments(segs.chas_example)
```

get_loss_segments *Return all segments with loss of 1 or 2 copies.*

Description

Return all segments with loss of 1 or 2 copies.

Usage

```
get_loss_segments(segments)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.

Value

A GRanges object containing the selected segments, their copy number and copy number types.

Examples

```
segs.loh <- get_loh_segments(segs.chas_example)
```

get_oncoscan_coverage_from_bed *Load the oncoscan coverage BED file into a GenomicRanges object.*

Description

Load the oncoscan coverage BED file into a GenomicRanges object.

Usage

```
get_oncoscan_coverage_from_bed(filename)
```

Arguments

filename	Path to the coverage BED file.

Details

Expects the following columns from the BED file (no header): 1. Name of the chromosomal arm (e.g. "1p") 2. Start position of the arm 3. End position of the arm

Value

A GRanges object containing the regions covered on each chromosome arm.

Examples

```
oncoscan_na33.cov <- get_oncoscan_coverage_from_bed(
       system.file('extdata', 'Oncoscan.na33.r2.cov.processed.bed',
       package = 'oncoscanR'))
```

---

load_ascat *Load am ASCAT text export file.*

Description

Load am ASCAT text export file.

Usage

```
load_ascat(filename, kit.coverage)
```

Arguments

filename	Path to the ASCAT file.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm by the kit.

Details

The ASCAT file is expected to have the following column names: 'chr' (chromosome number), 'startpos' (first position of CNV segment), 'endpos' (last position of CNV segment), 'nMajor' (Number of copies of the major allele) and 'nMinor' (Number of copies of the minor allele).

The segments are attributed to each chromosome arm and split if necessary.

Value

A GRanges object containing the segments, their copy number (field cn), their copy number types (field cntype). cntype contains either 'Gain', 'Loss' or 'LOH'. If the file contains twice the same segment or does not respect the format specifications, then an error is raised. NB. If the chromosome name is in the format '1' and not 'chr1' and will be transformed if needed.

Examples

```
segs.filename <- system.file('extdata', 'ascat_example.txt',
  package = 'oncoscanR')
segs.ascat_example <- load_ascat(segs.filename, oncoscan_na33.cov)
```

load_chas *Load a ChAS text export file.*

Description

Load a ChAS text export file.

Usage

```
load_chas(filename, kit.coverage)
```

Arguments

filename	Path to the ChAS file.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm by the kit.

Details

The ChAS file is expected to have the following column names: 'CN State' (number or empty), 'Type' (expected value: 'Gain', 'Loss' or 'LOH') and 'Full Location' (in the format 'chr:start-end').

The segments are attributed to each chromosome arm and split if necessary.

Value

A GRanges object containing the segments, their copy number (field cn), their copy number types (field cntype). cntype contains either 'Gain', 'Loss' or 'LOH'. If the file contains twice the same segment or does not respect the format specifications, then an error is raised. NB. The chromosome name is in the format '1' and not 'chr1' and will be transformed if needed.

Examples

```
segs.filename <- system.file('extdata', 'chas_example.txt',
  package = 'oncoscanR')
segs.chas_example <- load_chas(segs.filename, oncoscan_na33.cov)
```

`merge_segments` *Merge segments with respect to the kit resolution and the copy number.*

Description

Merge segments with respect to the kit resolution and the copy number.

Usage

```
merge_segments(segments, kit.resolution = 300)
```

Arguments

`segments`	A `GRanges` object containing the segments, their copy number and copy number types.
`kit.resolution`	Number >0 indicating the minimum segment size detectable by the technique (in kilobases). Defaults to the Oncoscan assay resolution outside of cancer genes: 300Kb.

Details

If two segments are at a distance smaller than the resolution, then the segments are merged if the share the same `cn` value. Note that the function does not look at the copy number type or subtype but only at the actual copy number to decide whether segments can be merged.

Value

A `GRanges` object containing the cleaned segments, their copy number and copy number types.

Examples

```
segs.merged <- merge_segments(segs.chas_example)
segs.merged_50k <- merge_segments(segs.chas_example, 50)
```

`oncoscan_na33.cov` *GenomicRanges object of the chromosomal arms coverage for the oncoscan assay (based on file extdata/Oncoscan.na33.r2.cov.processed.bed).*

Description

GenomicRanges object of the chromosomal arms coverage for the oncoscan assay (based on file extdata/Oncoscan.na33.r2.cov.processed.bed).

Usage

```
oncoscan_na33.cov
```

Format

A GRanges object containing the region covered on each chromosome arm.

Source

```
oncoscan_na33.cov <- get_oncoscan_coverage_from_bed( system.file('extdata', 'Oncoscan.na33.r2.cov.pro
package = 'oncoscanR'))
```

---

prune_by_size *Remove segments smaller than the kit resolution.*

Description

Remove segments smaller than the kit resolution.

Usage

```
prune_by_size(segments, threshold = 300)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
threshold	Number indicating the minimum segment size to be kept (in kilobases). Defaults to the Oncoscan assay resolution outside of cancer genes: 300Kb.

Value

A GRanges object containing the cleaned segments, their copy number and copy number types.

Examples

```
segs.300k <- prune_by_size(segs.chas_example)
segs.50k <- prune_by_size(segs.chas_example, 50)
```

score_avgcn *Compute the average copy number variation across the genome.*

Description

Compute the average copy number variation across the genome.

Usage

```
score_avgcn(segments, kit.coverage)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.

Details

Compute the weighted average (by segment length) of the copy number variation. LOH segments and sexual chromosomes are excluded. Copy number variation is rounded to the next level (1.67 -> 1 but 2.33 -> 3).

Value

A decimal value

Examples

```
score_avgcn(segs.chas_example, oncoscan_na33.cov)
```

score_estwgd *Estimates the number of whole-genome doubling events (WGD).*

Description

Estimates the number of whole-genome doubling events (WGD).

Usage

```
score_estwgd(segments, kit.coverage)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.

Details

Based on the publication from Carter et al. (Nature Biotechnology 2012; PubMed ID: 22544022). On a pan-cancer cohort, they observed that tumors that underwent one whole-genome doubling event had a ploidy (average copy number) between 2.2 and 3.4. This function relies on the function `score_avgcn` to compute the ploidy.

Value

A named list with two values: WGD (whole-genome doubling events) and avgCN (the average copy number). WGD values are 0 for no WGD event, 1 for one WGD event, 2 for several WGD events.

Examples

```
score_estwgd(segs.chas_example, oncoscan_na33.cov)
```

`score_gloh` *Compute the genomic LOH score.*

Description

Compute the genomic LOH score.

Usage

```
score_gloh(segments, arms.loh, arms.hetloss, kit.coverage)
```

Arguments

`segments`	A `GRanges` object containing the segments, their copy number and copy number types.
`arms.loh`	A list of arms with global/arm-level LOH alteration.
`arms.hetloss`	A list of arms with global/arm-level heterozygous loss.
`kit.coverage`	A `GRanges` object containing the regions covered on each chromosome arm.

Details

The percentage genomic LOH score is computed as described in the FoundationFocus CDx BRCA LOH assay; i.e. the percentage of bases covered by the Oncoscan that display a loss of heterozygosity independently of the number of copies, excluding chromosomal arms that have a global LOH (>=90 arm length). To compute with the `armlevel_alt` function on LOH segments only). This score was linked to BRCA1/2-deficient tumors.

Value

An integer representing the percentage of LOH bases.

Examples

```
armlevel.loh <- armlevel_alt(get_loh_segments(segs.chas_example),
                             kit.coverage = oncoscan_na33.cov)
armlevel.hetloss <- armlevel_alt(get_hetloss_segments(segs.chas_example),
                             kit.coverage = oncoscan_na33.cov)
score_gloh(segs.chas_example, names(armlevel.loh), names(armlevel.hetloss),
oncoscan_na33.cov)
```

---

score_loh *Compute the number HR deficiency-associated LOH regions.*

Description

Compute the number HR deficiency-associated LOH regions.

Usage

```
score_loh(segments, arms.loh, arms.hetloss, kit.coverage)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
arms.loh	A list of arms with global/arm-level LOH alteration.
arms.hetloss	A list of arms with global/arm-level heterozygous losses.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.

Details

Procedure based on the paper from Abkevich et al., Br J Cancer 2012 (PMID: 23047548). All LOH segments larger than 15Mb but excluding chromosome with a global LOH alteration (to compute with the armlevel_alt function on LOH segments only). This score was linked to BRCA1/2-deficient tumors. Note that the function will merge overlapping or neighbor LOH segments (at a distance of 1bp).

Value

An integer representing the number of HRD-LOH regions.

Examples

```
armlevel.loh <- armlevel_alt(get_loh_segments(segs.chas_example),
                             kit.coverage = oncoscan_na33.cov)
armlevel.hetloss <- armlevel_alt(get_hetloss_segments(segs.chas_example),
                             kit.coverage = oncoscan_na33.cov)
score_loh(segs.chas_example, names(armlevel.loh), names(armlevel.hetloss),
oncoscan_na33.cov)
```

score_lst *Compute the number of Large-scale State Transitions (LSTs).*

Description

Compute the number of Large-scale State Transitions (LSTs).

Usage

```
score_lst(segments, kit.coverage)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.

Details

Procedure based on the paper from Popova et al, Can. Res. 2012 (PMID: 22933060). First segments smaller than 3Mb are removed, then segments are smoothed with respect to copy number at a distance of 3Mb. The number of LSTs is the number of breakpoints (breakpoints closer than 3Mb are merged) that have a segment larger or equal to 10Mb on each side. This score was linked to BRCA1/2-deficient tumors.

Value

An integer representing the number of LSTs.

Examples

```
score_lst(segs.chas_example, oncoscan_na33.cov)
```

score_mbalt *Computes the total number of Mbp altered.*

Description

Computes the total number of Mbp altered.

Usage

```
score_mbalt(segments, kit.coverage, loh.rm = TRUE)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.
loh.rm	A boolean (TRUE by default) to indicate whether LOH segments should be excluded.

Value

A named list representing the Mbp altered in the sample and the total Mbp of the kit.

Examples

```
score_mbalt(segs.chas_example, oncoscan_na33.cov)
score_mbalt(segs.chas_example, oncoscan_na33.cov, FALSE)
```

---

score_nlst — *Compute the number of LSTs, normalized by the number of WGD events.*

Description

Compute the number of LSTs, normalized by the number of WGD events.

Usage

```
score_nlst(segments, n.wgd, kit.coverage, threshold = 15)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
n.wgd	Number of whole-genome doubling events (0 if diploid).
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.
threshold	A number above which the test is returned positive (>=).

Details

Compute the number of LSTs in non-LOH segments via the score_lst function and subtract the extra noise induced by WGD events: nLST = LST - 7*W/2 where W is the number of WGD events. A sample is HRD positive (deficient in HR pathway) if nLST is greater or equal to the threshold (15 by default). This score was linked to BRCA1/2-deficient tumors.

Value

A named list with the number of nLSTs and the corresponding label ('Positive', 'Negative').

Examples

```
w <- score_estwgd(segs.chas_example, oncoscan_na33.cov)
score_nlst(segs.chas_example, w['WGD'], oncoscan_na33.cov)
```

`score_td` *Compute the number of large tandem duplication (TDplus).*

Description

Compute the number of large tandem duplication (TDplus).

Usage

```
score_td(segments)
```

Arguments

`segments`	A `GRanges` object containing the segments, their copy number and copy number types.

Details

Procedure based on the paper from Popova et al., Cancer Res 2016 (PMID: 26787835). The TDplus score is defined as the number of regions larger than 1Mb but smaller or equal to 10Mb with a gain of one or two copies. This score was linked to CDK12-deficient tumors. They also identified as second category of tandem duplication whose size is smaller or equal than 1Mb and around 300Kb but could not link it to a phenotype. Note that due to its resolution the Oncoscan assay will most likely miss this second category. Nonetheless it is reported by the function.

Value

A list of integer containing the TDplus score (`'TDplus'`) and the small TD score (`'TD'`).

Examples

```
score_td(segs.chas_example)
```

segs.chas_example *Expected segments from loading the ChAS file 'chas_example.txt'.*

Description

Expected segments from loading the ChAS file 'chas_example.txt'.

Usage

```
segs.chas_example
```

Format

A GRanges object containing the segments, their copy number (field cn) and their copy number types (field cn.type).

Source

segs.filename <- system.file('extdata', 'chas_example.txt', package = 'oncoscanR') mykit.cov <- get_oncoscan_coverage_from_probes() segs.chas_example <- load_chas(segs.filename, kit.coverage = mykit.cov)

trim_to_coverage *Trim segments with respect to the kit's coverage.*

Description

Trim segments with respect to the kit's coverage.

Usage

```
trim_to_coverage(segments, kit.coverage)
```

Arguments

segments	A GRanges object containing the segments, their copy number and copy number types.
kit.coverage	A GRanges object containing the regions covered on each chromosome arm.

Details

All segments that are not entirely contained within the kit coverage will be trimmed to the coverage's limits.

Value

A GRanges object containing the cleaned segments, their copy number and copy number types.

Examples

```
segs.trimmed <- trim_to_coverage(segs.chas_example, oncoscan_na33.cov)
```

`workflow_oncoscan.ascat` *Run the standard workflow for ASCAT files (from oncoscan data).*

Description

Run the standard workflow for ASCAT files (from oncoscan data).

Usage

```
workflow_oncoscan.ascat(ascat.fn)
```

Arguments

`ascat.fn`	Path to the text-export ASCAT file

Details

Identifies the globally altered arms (\>=90% of arm altered), computes the HRD and TD+ scores. The amplification is defined as a CN>=5. An arm is gained if of CN type `cntype.gain` unless the arm is amplified.

Value

A list of lists with the following elements: `armlevel = list(AMP= list of arms, GAIN= list of arms, LOSS= list of arms, LOH= list of arms),scores = list(LST= number, LOH= number, TDplus= number, TD= number),file = path of the ChAS file as given by the parameter)`

Examples

```
segs.filename <- system.file('extdata', 'ascat_example.txt',
package = 'oncoscanR')
workflow_oncoscan.ascat(segs.filename)
```

workflow_oncoscan.chas *Run the standard workflow for Oncoscan ChAS files.*

Description

Run the standard workflow for Oncoscan ChAS files.

Usage

```
workflow_oncoscan.chas(chas.fn)
```

Arguments

chas.fn	Path to the text-export ChAS file

Details

Identifies the globally altered arms (\>=90% of arm altered), computes the HRD and TD+ scores. The amplification is defined as a CN subtype cntype.weakamp or cntype.strongamp. An arm is gained if of CN type cntype.gain unless the arm is amplified.

Value

A list of lists with the following elements: armlevel = list(AMP= list of arms, GAIN= list of arms, LOSS= list of arms, LOH= list of arms),scores = list(LST= number, LOH= number, TDplus= number, TD= number),file = path of the ChAS file as given by the parameter)

Examples

```
segs.filename <- system.file('extdata', 'chas_example.txt',
package = 'oncoscanR')
workflow_oncoscan.chas(segs.filename)
```

Index