Package: CINmetrics (via r-universe)

September 15, 2024

Type Package

Title Calculate Chromosomal Instability Metrics

Version 0.1.0

Author Vishal H. Oza, Roshan Darji, Brittany N. Lasseigne

Maintainer Vishal H. Oza <vishoza@uab.edu>

Description Implement various chromosomal instability metrics. 'CINmetrics' (Chromosomal INstability metrics) provides functions to calculate various chromosomal instability metrics on masked Copy Number Variation(CNV) data at individual sample level. The chromosomal instability metrics have been implemented as described in the following studies: Baumbusch LO et al. 2013 <doi:10.1371/journal.pone.0054356>, Davidson JM et al. 2014 <doi:10.1371/journal.pone.0079079>, Chin SF et al. 2007 <doi:10.1186/gb-2007-8-10-r215>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10), stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Date/Publication 2021-08-02 08:20:02 UTC

Repository https://vishaloza.r-universe.dev

RemoteUrl https://github.com/cran/CINmetrics

RemoteRef HEAD

RemoteSha 7e0cfc53032d4891fbbe2c27b33f947d1e265ef0

Contents

`CINmetrics` *CINmetrics*

Description

Calculate all CINmetrics on a given dataframe

Usage

```
CINmetrics(
  cnvData,
  segmentMean_tai = 0.2,
  segmentMean_cna = (log(1.7, 2) - 1),
  segmentMean_base_segments = 0.2,
  segmentMean_break_points = 0.2,
  segmentMean_fga = 0.2,
  numProbes = NA,
  segmentDistance_cna = 0.2,
  minSegSize_cna = 10,
  genomeSize_fga = 2873203431
)
```

Arguments

`cnvData`	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
`segmentMean_tai`	numerical value for the minimum segment_mean cutoff/ threshold for Total Aberration Index calculation. Default is 0.2
`segmentMean_cna`	numerical value for the minimum segment_mean cutoff/ threshold for Copy Number Aberration calculation. Default is 0.2
`segmentMean_base_segments`	numerical value for the minimum segment_mean cutoff/ threshold for Base segments calculation. Default is 0.2

segmentMean_break_points
: numerical value for the minimum segment_mean cutoff/ threshold for Break points calculation. Default is 0.2

segmentMean_fga
: numerical value for the minimum segment_mean cutoff/ threshold for Fraction of genome altered calculation. Default is 0.2

numProbes
: Number of Probes

segmentDistance_cna
: Segment distance threshold

minSegSize_cna
: Minimum segment size

genomeSize_fga
: Size of the genome derived from Affymetrix 6.0 array probe. Default is 2873203431 calculated based on hg38 **excluding sex chromosomes**

Value

All Chromosomal INstability metrics

Examples

```
CINmetrics(cnvData = maskCNV_BRCA)
```

---

cna *Copy Number Aberration*

Description

Calculates the number of copy number aberrations

Usage

```
cna(
  cnvData,
  segmentMean = (log(1.7, 2) - 1),
  numProbes = NA,
  segmentDistance = 0.2,
  minSegSize = 10
)
```

Arguments

cnvData
: dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean

segmentMean
: numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2

numProbes
: Number of Probes

segmentDistance
: Segment distance threshold

minSegSize
: Minimum segment size

Details

Copy Number Aberrations (CNA) (Davidson JM, et al), are defined as a segment with copy number outside the pre-defined range of 1.7-2.3

$$(\log_2 1.7 - 1) \leq \bar{y}_{S_i} \leq (\log_2 2.3 - 1)$$

that is not contiguous with an adjacent independent CNA of identical copy number. For our purposes, we have adapted the range to be

$$|\bar{y}_{S_i}| \geq |\log_2 1.7|$$

, which is only slightly larger than the original. It is nearly identical to countingBreakPoints, except this one calculates breaks as adjacent segments that have a difference in segment means of ≥ 0.2.

$$Total\ Copy\ Number\ Aberration = \sum_{i=1}^{R} n_i\ where\ \bar{y}_{S_i}| \geq |\log_2 1.7|,\ \bar{y}_{S_{i-1}} - \bar{y}_{S_i}| \geq 0.2,\ d_i \geq 10$$

Value

Number of copy number aberrations between segments

See Also

countingBreakPoints

Examples

```
cna(cnvData = maskCNV_BRCA)
```

`countingBaseSegments` *countingBaseSegments*

Description

Function for counting altered base segments

Usage

```
countingBaseSegments(cnvData, segmentMean = 0.2, numProbes = NA)
```

Arguments

`cnvData`	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
`segmentMean`	numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2
`numProbes`	Number of Probes

Details

The Altered Base Segment calculation takes all the CNV data for a single patient and first filters it for a segmentation mean of > 0.2 and, if specified, the minimum number of probes covering that area. Then, it calculates the sums of the lengths of each segment for a particular patient and outputs that.

$$Number\ of\ Altered\ Bases = \sum_{i=1}^{R} d_i\ where\ |\bar{y}_{S_i}| \geq 0.2$$

Value

Number of Base segments for each unique sample

Examples

```
countingBaseSegments(cnvData = maskCNV_BRCA)
```

`countingBreakPoints` *countingBreakPoints*

Description

The Break Point calculation takes all the CNV data for a single patient and first filters it for segmentation mean of > 0.2 and, if specified, the minimum number of probes covering that area. Then it counts the number of rows of data and multiplies it by 2. This represents the break points at the 5' and 3' ends of each segment.

$$Number\ of\ Break\ Points = \sum_{i=1}^{R}(n_i \cdot 2)\ where\ |\bar{y}_{S_i}| \geq 0.2$$

Usage

```
countingBreakPoints(cnvData, segmentMean = 0.2, numProbes = NA)
```

Arguments

`cnvData`	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
`segmentMean`	numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2
`numProbes`	Number of Probes

Value

Number of Break points for each unique sample

fga *Fraction Genome Altered*

Description

Fraction Genome Altered looks at the fraction of the genome that deviates from a diploid state fga calculates the fraction of the genome altered (FGA; [Chin SF, et. al.](https://www.ncbi.nlm.nih.gov/pubmed/17925008)), measured by taking the sum of the number of bases altered and dividing it by the genome length covered (G). Genome length covered was calculated by summing the lengths of each probe on the Affeymetrix 6.0 array. This calculation **excludes** sex chromosomes.

$$Fraction\ Genome\ Altered = \frac{\sum_{i=1}^{R} d_i}{G} \ \ where\ |\bar{y}_{S_i}| \geq 0.2$$

Usage

```
fga(cnvData, segmentMean = 0.2, numProbes = NA, genomeSize = 2873203431)
```

Arguments

cnvData	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
segmentMean	numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2
numProbes	Number of Probes
genomeSize	Size of the genome derived from Affymetrix 6.0 array probe. Default is 2873203431 calculated based on hg38 **excluding sex chromosomes**

Value

Fraction of the genome altered

Examples

```
fga(cnvData = maskCNV_BRCA)
```

maskCNV_BRCA *Breast Cancer Data from TCGA Data Release 25.0 GDC Product: Data Release Date: July 22, 2020 Masked Copy Number variation data for Breast Cancer for 10 unique samples selected randomly from TCGA*

Description

Breast Cancer Data from TCGA Data Release 25.0 GDC Product: Data Release Date: July 22, 2020 Masked Copy Number variation data for Breast Cancer for 10 unique samples selected randomly from TCGA

Usage

```
data(maskCNV_BRCA)
```

Format

An object of class dataframe

Source

https://portal.gdc.cancer.gov/

References

Koboldt, D., Fulton, R., McLellan, M. et al. (2012) Nature 490, 61–70 https://www.nature.com/articles/nature11412

Examples

```
data(maskCNV_BRCA)
tai <- tai(maskCNV_BRCA)
```

---

tai *Total Aberration Index*

Description

Total Aberration Index calculation takes the sum of lengths of each segment times its segmentation mean for each sample and divides it by the sum of the lengths of each sample.

Usage

```
tai(cnvData, segmentMean = 0.2, numProbes = NA)
```

Arguments

cnvData	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
segmentMean	numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2
numProbes	Number of Probes

Details

The Total Aberration Index (TAI) (Baumbusch LO, et. al.) is "a measure of the abundance of genomic size of copy number changes in a tumour". It is defined as a weighted sum of the segment means

$$Total\ Aberration\ Index = \frac{\sum_{i=1}^{R} d_i \cdot |\bar{y}_{S_i}|}{\sum_{i=1}^{R} d_i} \quad where |\bar{y}_{S_i}| \geq |\log_2 1.7|$$

Value

Average of lengths weighted by segmentation mean for each unique sample

Examples

```
tai(cnvData = maskCNV_BRCA)
```

`taiModified` *Modified Total Aberration Index*

Description

Modified Total Aberration Index calculation takes the sum of lengths of each segment times its segmentation mean for each sample and divides it by the sum of the lengths of each sample.

Usage

```
taiModified(cnvData, segmentMean = 0, numProbes = NA)
```

Arguments

`cnvData`	dataframe containing following columns: Sample, Start, End, Num_Probes, Segment_Mean
`segmentMean`	numerical value for the minimum segment_mean cutoff/ threshold. Default is 0.2
`numProbes`	Number of Probes

Details

Modified Total Aberration Index uses all sample values instead of those in aberrant copy number state, thus does not remove the directionality from the score.

$$Modified\ Total\ Aberration\ Index = \frac{\sum_{i=1}^{R} d_i \cdot \bar{y}_{S_i}}{\sum_{i=1}^{R} d_i}$$

Value

Average of lengths weighted by segmentation mean for each unique sample

See Also

`tai`

Examples

```
taiModified(cnvData = maskCNV_BRCA)
```

Index