Package 'dinamic'

May 8, 2026

Title A Method to Analyze Recurrent DNA Copy Number Aberrations in Tumors

Version 1.0.1

Description In tumor tissue, underlying genomic instability can lead to DNA copy number alterations, e.g., copy number gains or losses. Sporadic copy number alterations occur randomly throughout the
genome, whereas recurrent alterations are observed in the same genomic region across multiple independent samples, perhaps because they provide a selective growth advantage.
This package implements the DiNAMIC procedure for assessing the statistical significance of recurrent DNA copy number aberrations (Bioinformatics (2011) 27(5) 678 - 685).

Depends R (>= 4.2.0)

Suggests R.rsp

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Vonn Walter [aut, cre] (ORCID: <https://orcid.org/0000-0001-6114-6714>), Andrew B. Nobel [aut], Fred A. Wright [aut]

Maintainer Vonn Walter <vwalter1@pennstatehealth.psu.edu>

Repository CRAN

Date/Publication 2024-03-15 21:00:02 UTC

Contents

`annot.file` *Cytoband annotation data frame*

Description

Cytoband annotation information from the hg19 genome build

Usage

```
annot.file
```

Format

This four-column data frame contains cytoband annotation data that is used by the `makeCytoband` function. Each row corresponds to a distinct cytoband, and column 1 contains the chromosome number, column 2 contains the start position (in base pairs), column 3 contains the end position (in base pairs), and column 4 contains the cytoband name (e.g. p21.3). Additional columns may be present, but they are not used.

Details

The file cytoBand.txt.gz for the hg19 build can be downloaded from the UCSC Genome Browser at `http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/`. The format of cytoBand.txt differs from that of annot.file, but it can be used by the function `makeCytoband` if `reformat.cytoband = TRUE`.

Source

`http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/`

`detailedLook` *Assessing the Significance of Recurrent DNA Copy Number Aberrations*

Description

Assessing the Significance of Recurrent DNA Copy Number Aberrations

Usage

```
detailedLook(
  x,
  marker.data,
  annot.file,
  num.perms,
  num.iters,
  gain.loss = "gain",
  reformat.annot = FALSE,
  random.seed = NULL
)
```

Arguments

x	An n by m numeric matrix containing DNA copy number data from n subjects at m markers.
marker.data	A dataframe containing marker position data for markers in the autosomes. Column 1 contains the chromosome number for each marker, and column 2 contains the position (in base pairs) each markers. Additional columns, if present, represent information about the markers (e.g. probe names).
annot.file	A cytoband annotation dataframe. Each row corresponds to a distinct cytoband, and column 1 contains the chromosome number, column 2 contains the start position (in base pairs), column 3 contains the end position (in base pairs), and column 4 contains the cytoband name (e.g. p21.3). Additional columns may be present, but they are not used.
num.perms	A positive integer that represents the number of cyclic shifts used to create the empirical null distribution.
num.iters	A positive integer that represents the number of distinct gain (loss) loci that will be assessed.
gain.loss	A character string that indicates whether recurrent gains (gain.loss = "gain") or recurrent losses (gain.loss = "loss") are assessed.
reformat.annot	A logical value that indicates whether annot.file needs to be reformatted (default = FALSE). See the "note" section of makeCytoband for additional information.
random.seed	An optional random seed (default = NULL).

Details

This function applies the *Detailed Look* version of DiNAMIC's cyclic shift procedure to assess the statistical significance of recurrent DNA copy number aberrations. Either recurrent gains (gain.loss = "gain") or recurrent losses (gain.loss = "loss") are assessed using a null distribution based on num.perms cyclic shifts of x. Iterative calls to DiNAMIC's *peeling* procedure (implemented here in the peeling function) allow users to assess the statistical significance of num.iters distinct gains (losses). As noted in Bioinformatics (2011) 27(5) 678 - 685, the Detailed Look procedure recalculates the null distribution after each iteration of the peeling procedure. While this approach is more computationally intensive, simulations suggest that it provides more power to detect recurrent gains (losses).

Value

A matrix with num.iters rows. The entries of each row correspond to the marker that is being assessed. More specifically, the entries are (1) the chromosome number, (2) the marker position (in base pairs), (3) additional marker information present in marker.data, (4) the marker number, and (5) the p-value obtained from the null distribution, (6) the endpoints of the peak interval (in base pairs), as described in Bioinformatics (2011) 27(5) 678 - 685.

Examples

```
detailedLook(wilms.data, wilms.markers, annot.file, 100, 3)
```

---

makeCytoband *Find the chromosome arm for each marker*

Description

Find the chromosome arm for each marker

Usage

```
makeCytoband(marker.data, annot.file, reformat.annot = FALSE)
```

Arguments

marker.data	A two-column numeric matrix of marker position data for markers in the autosomes. Column 1 contains the chromosome number for each marker, and column 2 contains the position (in base pairs) for each marker. This is a submatrix of the marker position matrix used by quickLook and detailedLook.
annot.file	A dataframe containing cytoband annotation for the autosomes. Each row corresponds to a distinct cytoband, and column 1 contains the chromosome number, column 2 contains the start position (in base pairs), column 3 contains the end position (in base pairs), and column 4 contains the cytoband name (e.g. p21.3). Additional columns may be present, but they are not used.
reformat.annot	A logical value that indicates whether annot.file needs to be reformatted.

Details

DiNAMIC's peeling procedure is detailed in Bioinformatics (2011) 27(5) 678 - 685, and it is performed by the peeling function. By construction, the peeling procedure only affects markers in a given chromosome arm. This function is used internally by the peeling function to restrict the peeling procedure to the chromosome arm containing the marker that corresponds to max(colSums(x)).

Value

A character vector of length m, where m is the number of markers.

Examples

```
wilms.pq = makeCytoband(wilms.markers, annot.file)
#A character vector of length 3288, and each entry is either
#"p" or "q", depending on the chromosome arm of the given marker.
table(wilms.pq)
#Produces the following output:
#wilms.pq
#   p    q
#1147 2141
```

`peeling` *Apply the peeling procedure at a given marker*

Description

Apply the peeling procedure at a given marker

Usage

```
peeling(x, marker.data, cytoband, k)
```

Arguments

`x`	An n by m numeric matrix containing DNA copy number data from n subjects at m markers.
`marker.data`	marker.data A two-column numeric matrix of marker position data for markers in the autosomes. Column 1 contains the chromosome number for each marker, and column 2 contains the position (in base pairs) for each markers. This is a submatrix of the marker position matrix used by `quickLook` and `detailedLook`.
`cytoband`	A character vector of length m that contains the chromosome arm (p or q) for each marker. This is produced by the `makeCytoband` function.
`k`	A positive integer between 1 and m that represents the most aberrant marker.

Details

The peeling procedure is detailed in Algorithm 2 of Bioinformatics (2011) 27(5) 678 - 685, but here we provide a brief overview. By construction, marker `k` represents the most aberrant gain (loss). The peeling procedure rescales all copy number values in `x` that contribute to making marker `k` aberrant, so that after applying the peeling procedure marker `k` is "null." By construction, the rescaling procedure is restricted to entries in `x` that correspond to markers in the same chromosome arm as `k`. This allows users to assess the statistical significance of multiple gains (losses) throughout the genome.

Value

A list containing two components: (1) the n by m matrix produced by applying the peeling algorithm to the matrix x at marker k, and (2) the peak interval around marker k, as described in Bioinformatics (2011) 27(5) 678 - 685.

`quickLook` *Find DiNAMIC's null distribution*

Description

Find DiNAMIC's null distribution

Usage

```
quickLook(
  x,
  marker.data,
  annot.file,
  num.perms,
  num.iters,
  gain.loss = "gain",
  reformat.annot = FALSE,
  random.seed = NULL
)
```

Arguments

`x`	An n by m numeric matrix containing DNA copy number data from n subjects at m markers.
`marker.data`	A dataframe containing marker position data for markers in the autosomes. Column 1 contains the chromosome number for each marker, and column 2 contains the position (in base pairs) for each markers. Additional columns, if present, represent information about the markers (e.g. probe names).
`annot.file`	A cytoband annotation dataframe. Each row corresponds to a distinct cytoband, and column 1 contains the chromosome number, column 2 contains the start position (in base pairs), column 3 contains the end position (in base pairs), and column 4 contains the cytoband name (e.g. p21.3). Additional columns may be present, but they are not used.
`num.perms`	A positive integer that represents the number of cyclic shifts used to create the empirical distribution.
`num.iters`	A positive integer that represents the number of distinct gain (loss) loci that will be assessed. See "Details" for more information.
`gain.loss`	A character string that indicates whether recurrent gains (`gain.loss = "gain"`) or recurrent losses (`gain.loss = "loss"`) are assessed.
`reformat.annot`	A logical value that indicates whether annot.file needs to be reformatted (default = FALSE). See the "Note" section of `makeCytoband` for additional information.
`random.seed`	An optional random seed (default = NULL).

Details

This function applies the "Quick Look" version of DiNAMIC's cyclic shift procedure to assess the statistical significance of recurrent DNA copy number aberrations. Either recurrent gains (`gain.loss = "gain"`) or recurrent losses (`gain.loss = "loss"`) are assessed using a null distribution based on num.perms cyclic shifts of `x`. Iterative calls to DiNAMIC's peeling procedure (implemented here in the `peeling` function) allow users to assess the statistical significance of num.iters distinct gains (losses). As noted in Bioinformatics (2011) 27(5) 678 - 685, the "Quick Look" procedure calculates the null distribution once, and the same distribution is used to assess the statistical significance of the most aberrant gain or loss after each iteration of the peeling procedure. This approach is less computationally intensive than "Detailed Look" because the null distribution is only computed once, but simulations suggest that it provides less power to detect recurrent gains (losses). The resulting p-values are corrected for multiple comparisons because the null distribution is based on computing `max(colSums(x))` or `min(colSums(x))`.

Value

A matrix with `num.iters` rows. The entries of each row correspond to the marker that is being assessed. More specifically, the entries are (1) the chromosome number, (2) the marker position (in base pairs), (3) additional marker information present in marker.data, (4) the marker number, and (5) the p-value obtained from the null distribution, (6) the endpoints of the peak interval (in base pairs), as described in Bioinformatics (2011) 27(5) 678 - 685.

Examples

```
quickLook(wilms.data, wilms.markers, annot.file, 100, 3)
```

`recodeBinary` *Recode binary vectors*

Description

Recode binary vectors

Usage

```
recodeBinary(binary.vec, k)
```

Arguments

`binary.vec`	A binary vector of length `m` (>= 1) whose kth entry is 1.
`k`	A positive integer.

Details

This function is called internally by `peeling`.

Value

A binary vector of length m that contains a single contiguous string of 1's, namely the string that contains the 1 in the kth position of binary.vec.

Examples

```
test = c(1, 0, 0, 1, 1, 0, 0, 1, 0)
recodeBinary(test, 5)
#Returns (0, 0, 0, 1, 1, 0, 0, 0, 0)
```

`wilms.data` *DNA copy number data from Wilms' tumor*

Description

Probe-level DNA copy number data from Wilms' tumor (Natrajan et al., 2006)

Usage

```
wilms.data
```

Format

A 97 by 3288 numeric matrix containing DNA copy number data, as described below.

Details

Natrajan et al. (J. Pathology (2006) 210: 49 - 58) used array comparative genomic hybridization to obtain genome-wide DNA copy number data from 97 Wilms' tumor samples at 3288 markers. This matrix contains the DNA copy number data after applying the bias-correction procedure outlined in Bioinformatics (2011) 27(5) 678 - 685. Each row corresponds to DNA copy number from one subject at 3288 markers, while each column contains DNA copy number data for 97 subjects at one marker.

Source

https://www.ebi.ac.uk/biostudies/arrayexpress accession number E-TABM-10.

`wilms.markers` *Array comparative genomic hybridization marker data*

Description

Array comparative genomic hybridization marker data from Natrajan et al. (2006)

Usage

```
wilms.markers
```

Format

A data frame with 3288 observations on the following 3 variables.

`Chromosome` The chromosome for the given marker

`Position` The position (in bp) for the given marker

`Name` The name of the marker (e.g., R:A-MEXP-192:RP11-465B22)

Details

Natrajan et al. (J. Pathology (2006) 210: 49 - 58) used array comparative genomic hybridization to obtain genome-wide DNA copy number data from 97 Wilms' tumor samples at 3288 markers. This data frame contains genomic position data for the probes in the array.

Source

https://www.ebi.ac.uk/biostudies/arrayexpress accession number E-TABM-10.

Index