Package 'TopDom'

May 6, 2021

Version 0.10.1

Title An Efficient and Deterministic Method for Identifying Topological Domains in Genomes

Depends R (>= 3.3.0)

Imports matrixStats, grid, ggplot2, reshape2, tibble

Suggests diffobj (>= 0.1.11)

Description The 'TopDom' method identifies topological domains in genomes from Hi-C sequence data (Shin et al., 2016 <doi:10.1093/nar/gkv1505>). The authors published an implementation of their method as an R script (two different versions; also available in this package). This package originates from those original 'TopDom' R scripts and provides help pages adopted from the original 'TopDom' PDF documentation. It also provides a small number of bug fixes to the original code.

License GPL

LazyLoad TRUE

URL https://github.com/HenrikBengtsson/TopDom

BugReports https://github.com/HenrikBengtsson/TopDom/issues

RoxygenNote 7.1.1

NeedsCompilation no

Author Henrik Bengtsson [aut, cre, cph],
Hanjun Shin [aut, ctr, cph],
Harris Lazaris [ctr, cph] (PhD Student, NYU),
Gangqing Hu [ctr, cph] (Staff Scientist, NIH),
Xianghong Zhou [ctr]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Repository CRAN

Date/Publication 2021-05-06 07:40:03 UTC

R topics documented:

`countsPerRegion` *Calculates Counts per Region in a TopDomData Object*

Description

Calculates Counts per Region in a TopDomData Object

Usage

```
countsPerRegion(data, regions)
```

Arguments

`data`	A TopDomData object.
`regions`	TopDom regions (a data.frame), e.g. domains.

Value

A numeric vector of length `nrow(regions)`.

Author(s)

Henrik Bengtsson.

ggCountHeatmap *Produce a Count Heatmap*

Description

Produce a Count Heatmap

Usage

```
ggCountHeatmap(data, transform, colors, ...)
```

Arguments

`data`	A TopDomData object.
`transform`	A function applied to the counts prior to generating heatmap colors.
`colors`	A named list to control to color scale.
`...`	Not used.

Value

A ggplot2::ggplot object.

Author(s)

Henrik Bengtsson.

See Also

See TopDom for an example.

ggDomain *Add a Topological Domain to a Count Heatmap*

Description

Add a Topological Domain to a Count Heatmap

Usage

```
ggDomain(td, dx = NULL, delta = 0.04, vline = 0, size = 2, color = "#666666")
```

Arguments

`td`	A single-row data.frame.
`dx, delta, vline`	Absolute distance to heatmap. If `dx = NULL` (default), then `dx = delta * w + vline` where `w` is the width of the domain.
`size, color`	The thickness and color of the domain line.

Value

A ggplot2::geom_segment object to be added to the count heatmap.

`ggDomainLabel` *Add a Topological Domain Label to a Count Heatmap*

Description

Add a Topological Domain Label to a Count Heatmap

Usage

```
ggDomainLabel(
  td,
  fmt = "%s: %.2f - %.2f Mbp",
  rot = 45,
  dx = 0,
  vjust = 2.5,
  cex = 1.5
)
```

Arguments

`td`	A single-row data.frame.
`fmt`	The base::sprintf-format string taking (chromosome, start, stop) as (string, numeric, numeric) input.
`rot`	The amount of rotation in [0,360] of label.
`dx, vjust`	The vertical adjustment of the label (relative to rotation)
`cex`	The scale factor of the label.

Value

A ggplot2::ggproto object to be added to the count heatmap.

legacy *Easy Access to the Original TopDom 0.0.1 and 0.0.2 Implementations*

Description

Easy Access to the Original TopDom 0.0.1 and 0.0.2 Implementations

Usage

```
legacy(version = c("0.0.1", "0.0.2"))
```

Arguments

version	A version string.

Value

An environment containing the legacy TopDom API.

Examples

```
TopDom::legacy("0.0.2")$TopDom
TopDom::legacy("0.0.1")$Detect.Local.Extreme
```

overlapScores *Calculates Overlap Scores Between Two Sets of Topological Domains*

Description

Calculates Overlap Scores Between Two Sets of Topological Domains

Usage

```
overlapScores(a, reference, debug = getOption("TopDom.debug", FALSE))
```

Arguments

a, reference	Topological domain (TD) set A and TD reference set R both in a format as returned by TopDom().
debug	If TRUE, debug output is produced.

Details

The *overlap score*, $overlap(A', r_i)$, represents how well a *consecutive* subset A' of topological domains (TDs) in A overlap with topological domain r_i in reference set R. For each reference TD r_i, the *best match* A'_max is identified, that is, the A' subset that maximize $overlap(A', r_i)$. For exact definitions, see Page 8 in Shin et al. (2016).

Note that the overlap score is an asymmetric score, which means that `overlapScores(a,b) != overlapScores(b,a)`.

Value

Returns a named list of class `TopDomOverlapScores`, where the names correspond to the chromosomes in domain reference set R. Each of these chromosome elements contains a data.frame with fields:

- `chromosome` - $D_{R,c}$ character strings
- `best_score` - $D_{R,c}$ numerics in $[0, 1]$
- `best_length` - $D_{R,c}$ positive integers
- `best_set` - list of $D_{R,c}$ index vectors

where $D_{R,c}$ is the number of TDs in reference set R on chromosome c. If a TD in reference R is not a `"domain"`, then the corresponding `best_score` and `best_length` values are `NA_real_` and `NA_integer_`, respectively, while `best_set` is an empty list.

Warning - This might differ not be the correct implementation

The original TopDom scripts do not provide an implementation for calculating overlap scores. Instead, the implementation of `TopDom::overlapScores()` is based on the textual description of overlap scores provided in Shin et al. (2016). It is not known if this is the exact same algorithm and implementation as the authors of the TopDom article used.

Author(s)

Henrik Bengtsson - based on the description in Shin et al. (2016).

References

- Shin et al., TopDom: an efficient and deterministic method for identifying topological domains in genomes, *Nucleic Acids Research*, 44(7): e70, April 2016. doi: 10.1093/nar/gkv1505, PMCID: PMC4838359, PMID: 26704975

See Also

TopDom.

Examples

```
library(tibble)
path <- system.file("exdata", package = "TopDom", mustWork = TRUE)

## Original count data (on a subset of the bins to speed up example)
chr <- "chr19"
pathname <- file.path(path, sprintf("nij.%s.gz", chr))
data <- readHiC(pathname, chr = chr, binSize = 40e3, bins = 1:500)
print(data)

## Find topological domains using TopDom method for two window sizes
tds_5 <- TopDom(data, window.size = 5L)
tds_6 <- TopDom(data, window.size = 6L)

## Overlap scores (in both directions)
overlap_56 <- overlapScores(tds_6, reference = tds_5)
print(overlap_56)
print(as_tibble(overlap_56))

overlap_65 <- overlapScores(tds_5, reference = tds_6)
print(overlap_65)
print(as_tibble(overlap_65))
```

`readHiC` *Reads Hi-C Contact Data from File*

Description

Reads Hi-C Contact Data from File

Usage

```
readHiC(
  file,
  chr = NULL,
  binSize = NULL,
  ...,
  debug = getOption("TopDom.debug", FALSE)
)
```

Arguments

`file`	The pathname of a normalize Hi-C contact matrix file stored as a whitespace-delimited file. See below for details. Also a gzip-compressed file can be used.
`chr, binSize`	If the file contains a count matrix without bin annotation, the latter is created from these parameters.
`debug`	If `TRUE`, debug output is produced.
`...`	Arguments passed to `utils::read.table()` as-is.

Value

A list with elements `bins` (an N-by-4 data.frame) and `counts` (N-by-N matrix).

Format of HiC contact-matrix file

The contact-matrix file should be a whitespace-delimited text file with neither row names nor column names. The content should be a N-by-(3+N) table where the first three columns correspond to `chr` (string), `from.coord` (integer position), and `to.coord` (integer position). These column defines the genomic location of the N Hi-C bins (in order). The last N columns should contain normalized contact counts (float) such that element (r,3+c) in this table corresponds to count (r,c) in the normalized contact matrix.

If an N-by-(4+N) table, then the first column is assumed to contain an `id` (integer), and everything else as above.

Example:

```
chr10       0   40000  0 0 0 0 ...
chr10   40000   80000  0 0 0 0 ...
chr10   80000  120000  0 0 0 0 ...
chr10  120000  160000  0 0 0 0 ...
...
```

See Also

TopDom.

Examples

```
path <- system.file("exdata", package = "TopDom", mustWork = TRUE)

## Original count data
chr <- "chr19"
pathname <- file.path(path, sprintf("nij.%s.gz", chr))
data <- readHiC(pathname, chr = chr, binSize = 40e3)
print(data)
str(data)
```

`subsetByRegion` *Subset a TopDomData Object by Region*

Description

Subset a TopDomData Object by Region

Usage

```
subsetByRegion(data, region, margin = 1/2)
```

Arguments

data	A TopDomData object.
region	A TopDom domain (a data.frame).
margin	An non-negative numeric specifying the additional margin extracted around the domain. If margin < 1, then the size of the margin is relative to the size of the domain.

Value

A TopDomData object.

Author(s)

Henrik Bengtsson.

TopDom *Identify Topological Domains from a Hi-C Contact Matrix*

Description

Identify Topological Domains from a Hi-C Contact Matrix

Usage

```
TopDom(
  data,
  window.size,
  outFile = NULL,
  statFilter = TRUE,
  ...,
  debug = getOption("TopDom.debug", FALSE)
)
```

Arguments

data	A TopDomData object, or the pathname to a normalized Hi-C contact matrix file as read by readHiC(), that specify N bins.
window.size	The number of bins to extend (as a non-negative integer). Recommended range is in 5, ..., 20.
outFile	(optional) The filename without extension of the three result files optionally produced. See details below.
statFilter	(logical) Specifies whether non-significant topological-domain boundaries should be dropped or not.
...	Additional arguments passed to readHiC().
debug	If TRUE, debug output is produced.

Value

A named list of class TopDom with data.frame elements binSignal, domain, and bed.

- The binSignal data frame (N-by-7) holds mean contact frequency, local extreme, and p-value for every bin. The first four columns represent basic bin information given by matrix file, such as bin id (id), chromosome(chr), start coordinate (from.coord), and end coordinate (to.coord) for each bin. The last three columns (local.ext, mean.cf, and p-value) represent computed values by the TopDom algorithm. The columns are:
  - id: Bin ID
  - chr: Chromosome
  - from.coord: Start coordinate of bin
  - to.coord: End coordinate of bin
  - local.ext:
    - -1: Local minima.
    - -0.5: Gap region.
    - 0: General bin.
    - 1: Local maxima.
  - mean.cf: Average of contact frequencies between lower and upper regions for bin i = *1,2,...,N*.
  - p-value: Computed p-value by Wilcox rank sum test. See Shin et al. (2016) for more details.
- The domain data frame (D-by-7): Every bin is categorized by basic building block, such as gap, domain, or boundary. Each row indicates a basic building block. The first five columns include the basic information about the block, 'tag' column indicates the class of the building block.
  - id: Identifier of block
  - chr: Chromosome
  - from.id: Start bin index of the block
  - from.coord: Start coordinate of the block
  - to.id: End bin index of the block
  - to.coord: End coordinate of the block
  - tag: Categorized name of the block. Three possible blocks exists:
    - gap
    - domain
    - boundary
  - size: size of the block
- The bed data frame (D-by-4) is a representation of the domain data frame in the BED file format. It has four columns:
  - chrom: The name of the chromosome.
  - chromStart: The starting position of the feature in the chromosome. The first base in a chromosome is numbered 0.
  - chromEnd: The ending position of the feature in the chromosome. The chromEnd base is *not* included in the feature. For example, the first 100 bases of a chromosome are defined as chromStart=0, chromEnd=100, and span the bases numbered 0-99.

- name: Defines the name of the BED line. This label is displayed to the left of the BED line in the UCSC Genome Browser window when the track is open to full display mode or directly to the left of the item in pack mode.

If argument outFile is non-NULL, then the three elements (binSignal, domain, and bed) returned are also written to tab-delimited files with file names '<outFile>.binSignal', '<outFile>.domain', and '<outFile>.bed', respectively. None of the files have row names, and all but the BED file have column names.

Windows size

The window.size parameter is by design the only tuning parameter in the TopDom method and affects the amount of smoothing applied when calculating the TopDom bin signals. The binning window extends symmetrically downstream and upstream from the bin such that the bin signal is the average window.size^2 contact frequencies. For details, see Equation (1) and Figure 1 in Shin et al. (2016). Typically, the number of identified TDs decreases while their average lengths increase as this window-size parameter increases (Figure 2). The default is window.size = 5 (bins), which is motivated as: "Considering the previously reported minimum TD size (approx. 200 kb) (Dixon et al., 2012) and our bin size of 40 kb, *w*[indow.size] = 5 is a reasonable setting" (Shin et al., 2016).

Author(s)

Hanjun Shin, Harris Lazaris, and Gangqing Hu. R package, help, and code refactoring by Henrik Bengtsson.

References

- Shin et al., TopDom: an efficient and deterministic method for identifying topological domains in genomes, *Nucleic Acids Research*, 44(7): e70, April 2016. DOI: 10.1093/nar/gkv1505, PMCID: PMC4838359, PMID: 26704975
- Shin et al., R script 'TopDom_v0.0.2.R', 2017 (originally from http://zhoulab.usc.edu/TopDom/; later available on https://github.com/jasminezhoulab/TopDom via https://zhoulab.dgsom.ucla.edu/pages/software)
- Shin et al., TopDom Manual, 2016-07-08 (original from http://zhoulab.usc.edu/TopDom/TopDom%20Manual_v0.0 later available on https://github.com/jasminezhoulab/TopDom via https://zhoulab.dgsom.ucla.edu/pages/software)
- Hanjun Shin, Understanding the 3D genome organization in topological domain level, Doctor of Philosophy Dissertation, University of Southern California, March 2017, http://digitallibrary.usc.edu/cdm/ref/collection/p15799coll40/id/347735
- Dixon JR, Selvaraj S, Yue F, Kim A, et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*; 485(7398):376-80, April 2012. DOI: 10.1038/nature11082, PMCID: PMC3356448, PMID: 22495300.

Examples

```
path <- system.file("exdata", package = "TopDom", mustWork = TRUE)

## Original count data (on a subset of the bins to speed up example)
chr <- "chr19"
```

```
pathname <- file.path(path, sprintf("nij.%s.gz", chr))
data <- readHiC(pathname, chr = chr, binSize = 40e3, bins = 1:500)
print(data)  ## a TopDomData object

## Find topological domains using the TopDom method
fit <- TopDom(data, window.size = 5L)
print(fit)  ## a TopDom object

## Display the largest domain
td <- subset(subset(fit$domain, tag == "domain"), size == max(size))
print(td) ## a data.frame

## Subset TopDomData object
data_s <- subsetByRegion(data, region = td, margin = 0.9999)
print(data_s)  ## a TopDomData object

vp <- grid::viewport(angle = -45, width = 0.7, y = 0.3)
gg <- ggCountHeatmap(data_s)
gg <- gg + ggDomain(td, color = "#cccc00") + ggDomainLabel(td)
print(gg, newpage = TRUE, vp = vp)

gg <- ggCountHeatmap(data_s, colors = list(mid = "white", high = "black"))
gg_td <- ggDomain(td, delta = 0.08)
dx <- attr(gg_td, "gg_params")$dx
gg <- gg + gg_td + ggDomainLabel(td, vjust = 2.5)
print(gg, newpage = TRUE, vp = vp)

## Subset TopDom object
fit_s <- subsetByRegion(fit, region = td, margin = 0.9999)
print(fit_s)  ## a TopDom object
for (kk in seq_len(nrow(fit_s$domain))) {
  gg <- gg + ggDomain(fit_s$domain[kk, ], dx = dx * (4 + kk %% 2), color = "red", size = 1)
}

print(gg, newpage = TRUE, vp = vp)


gg <- ggCountHeatmap(data_s)
gg_td <- ggDomain(td, delta = 0.08)
dx <- attr(gg_td, "gg_params")$dx
gg <- gg + gg_td + ggDomainLabel(td, vjust = 2.5)
fit_s <- subsetByRegion(fit, region = td, margin = 0.9999)
for (kk in seq_len(nrow(fit_s$domain))) {
  gg <- gg + ggDomain(fit_s$domain[kk, ], dx = dx * (4 + kk %% 2), color = "blue", size = 1)
}

print(gg, newpage = TRUE, vp = vp)
```

TopDom-data *Data for the TopDom Package*

Description

The 'exdata/' folder of this package provides a example data set used in examples. The data are also used to validate the **TopDom** implementation toward the original TopDom scripts.

Origin

The data herein contain a tiny subset of the HiC and TopDom data used in the TopDom study (Shin et al., 2016). More precisely, it contains:

1. A TopDom file 'mESC_5w_chr19.nij.HindIII.comb.40kb.domain', which is part of the 'mESC_5w_domain.zip' file (5,504 bytes; md5 ffb19996f681a4d35d5c9944f2c44343) from the Supplementary Materials of Shin et al. (2016). These data were downloaded from the TopDom website (http://zhoulab.usc.edu/TopDom/ - now defunct).
2. A normalized HiC-count matrix file 'nij.chr19.gz', where the non-compressed version is part of the 'mESC.norm.tar.gz' file (1,305,763,679 bytes; md5 2e79d0f57463b5b7c4bf86b187086d3c) originally downloaded from the UCSD Ren Lab. It is a tab-delimited file containing a 3250-by-3250 numeric matrix non-negative decimal values. The underlying HiC sequence data is available from GSE35156 on GEO and was published part of Dixon, et al. (2012).

References

1. Dixon JR, Selvaraj S, Yue F, Kim A, et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature 2012 Apr 11; 485(7398):376-80, doi: 10.1038/nature11082, PMCID: PMC3356448, PMID: 22495300.
2. Shin, et al., TopDom: an efficient and deterministic method for identifying topological domains in genomes, Nucleic Acids Res. 2016 Apr 20; 44(7): e70., 2016. doi: 10.1093/nar/gkv1505, PMCID: PMC4838359, PMID: 26704975.

Index