Package 'HiCDOC'

September 19, 2024

Title A/B compartment detection and differential analysis

Version 1.7.0

Date 2022-11-30

Description HiCDOC normalizes intrachromosomal Hi-C matrices, uses unsupervised learning to predict A/B compartments from multiple replicates, and detects significant compartment changes between experiment conditions.
It provides a collection of functions assembled into a pipeline to filter and normalize the data, predict the compartments and visualize the results.
It accepts several type of data: tabular `.tsv` files, Cooler `.cool` or `.mcool` files, Juicer `.hic` files or HiC-Pro `.matrix` and `.bed` files.

License LGPL-3

Encoding UTF-8

SystemRequirements C++11

Depends InteractionSet, GenomicRanges, SummarizedExperiment, R (>= 4.1.0)

Imports methods, zlibbioc, ggplot2, Rcpp (>= 0.12.8), stats, S4Vectors, gtools, pbapply, BiocParallel, BiocGenerics, grid, cowplot, gridExtra, data.table, multiHiCcompare, GenomeInfoDb

LinkingTo Rcpp

Suggests knitr, rmarkdown, testthat, BiocStyle, BiocManager, rhdf5

RoxygenNote 7.2.3

biocViews HiC, DNA3DStructure, Normalization, Sequencing, Software, Clustering

VignetteBuilder knitr

URL https://github.com/mzytnicki/HiCDOC

BugReports https://github.com/mzytnicki/HiCDOC/issues

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

git_branch devel

git_last_commit 7216e65

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Author Kurylo Cyril [aut],
Zytnicki Matthias [aut],
Foissac Sylvain [aut],
Maigné Élise [aut, cre]

Maintainer Maigné Élise <elise.maigne@inrae.fr>

Contents

`HiCDOC-package` *A/B compartment detection and differential analysis*

Description

HiCDOC normalizes intrachromosomal Hi-C matrices, uses unsupervised learning to predict A/B compartments from multiple replicates, and detects significant compartment changes between experiment conditions. It provides a collection of functions assembled into a pipeline to filter and normalize the data, predict the compartments and visualize the results. It accepts several type of data: tabular '.tsv' files, Cooler '.cool' or '.mcool' files, Juicer '.hic' files or HiC-Pro '.matrix' and '.bed' files.

Author(s)

Maintainer: Maigné Élise <elise.maigne@inrae.fr>

Authors:

- Kurylo Cyril <cyril.kurylo@inrae.fr>
- Zytnicki Matthias <matthias.zytnicki@inrae.fr>
- Foissac Sylvain <sylvain.foissac@inrae.fr>

See Also

Useful links:

- https://github.com/mzytnicki/HiCDOC
- Report bugs at https://github.com/mzytnicki/HiCDOC/issues

`detectCompartments` *A and B compartments detection and differences across conditions.*

Description

Detects compartments for each genomic position in each condition, and computes p-values for compartment differences between conditions.

Usage

```
detectCompartments(
    object,
    parallel = FALSE,
    kMeansDelta = NULL,
    kMeansIterations = NULL,
    kMeansRestarts = NULL,
    PC1CheckThreshold = NULL
)
```

Arguments

`object`	A `HiCDOCDataSet`.
`parallel`	Whether or not to parallelize the processing. Defaults to FALSE See 'Details'.
`kMeansDelta`	The convergence stop criterion for the clustering. When the centroids' distances between two iterations is lower than this value, the clustering stops. Defaults to `object$kMeansDelta` which is originally set to `defaultHiCDOCParameters$kMeansDelta` = 0.0001.
`kMeansIterations`	The maximum number of iterations during clustering. Defaults to `object$kMeansIterations` which is originally set to `defaultHiCDOCParameters$kMeansIterations` = 50.
`kMeansRestarts`	The amount of times the clustering is restarted. For each restart, the clustering iterates until convergence or reaching the maximum number of iterations. The clustering that minimizes inner-cluster variance is selected. Defaults to `object$kMeansRestarts` which is originally set to `defaultHiCDOCParameters$kMeansRestarts` = 20.

`PC1CheckThreshold`
: The minimum percentage of variance that should be explained by the first principal component of centroids to pass sanity check. Defaults to `object$PC1CheckThreshold` which is originally set to `defaultHiCDOCParameters$PC1CheckThreshold` = 0.75

Details

Genomic positions clustering: To clusterize genomic positions, the algorithm follows these steps:

1. For each chromosome and condition, get the interaction vectors of each genomic position. Each genomic position can have multiple interaction vectors, corresponding to the multiple replicates in that condition.
2. For each chromosome and condition, use constrained K-means to clusterize the interaction vectors, forcing replicate interaction vectors into the same cluster. The euclidean distance between interaction vectors determines their similarity.
3. For each interaction vector, compute its concordance, which is the confidence in its assigned cluster. Mathematically, it is the log ratio of its distance to each centroid, normalized by the distance between both centroids, and min-maxed to a [-1,1] interval.
4. For each chromosome, compute the distance between all centroids and the centroids of the first condition. The cross-condition clusters whose centroids are closest are given the same cluster label. This results in two clusters per chromosome, spanning all conditions.

A/B compartments prediction: To match each cluster with an A or B compartment, the algorithm follows these steps:

1. For each genomic position, compute its self interaction ratio, which is the difference between its self interaction and the median of its other interactions.
2. For each chromosome, for each cluster, get the median self interaction ratio of the genomic positions in that cluster.
3. For each chromosome, the cluster with the smallest median self interaction ratio is matched with compartment A, and the cluster with the greatest median self interaction ratio is matched with compartment B. Compartment A being open, there are more overall interactions between distant genomic positions, so it is assumed that the difference between self interactions and other interactions is lower than in compartment B.

Significant differences detection: To find significant compartment differences across conditions, and compute their p-values, the algorithm follows three steps:

1. For each pair of replicates in different conditions, for each genomic position, compute the absolute difference between its concordances.
2. For each pair of conditions, for each genomic position, compute the median of its concordance differences.
3. For each pair of conditions, for each genomic position whose assigned compartment switches, rank its median against the empirical cumulative distribution of medians of all non-switching positions in that condition pair. Adjust the resulting p-value with the Benjamini–Hochberg procedure.

Parallel processing: The parallel version of detectCompartments uses the `bpmapply` function. Before to call the function in parallel you should specify the parallel parameters such as:

- On Linux:
  `multiParam <- BiocParallel::MulticoreParam(workers = 10)`
- On Windows:
  `multiParam <- BiocParallel::SnowParam(workers = 10)`

And then you can register the parameters to be used by BiocParallel:

```
BiocParallel::register(multiParam, default = TRUE)
```

You should be aware that using MulticoreParam, reproducibility of the detectCompartments function using a RNGseed may not work. See this issue for more details.

Value

A HiCDOCDataSet, with compartments, concordances, distances, centroids, and differences.

Examples

```
data(exampleHiCDOCDataSet)
## Run all filtering and normalization steps (not run for timing reasons)
# object <- filterSmallChromosomes(exampleHiCDOCDataSet)
# object <- filterSparseReplicates(object)
# object <- filterWeakPositions(object)
# object <- normalizeTechnicalBiases(object)
# object <- normalizeBiologicalBiases(object)
# object <- normalizeDistanceEffect(object)

# Detect compartments and differences across conditions
object <- detectCompartments(exampleHiCDOCDataSet)
```

---

exampleHiCDOCDataSet *Example HiCDOCDataSet.*

Description

A S4 HiCDOCDataSet object with 4 chromosomes, 3 conditions and 3 replicates.

Usage

```
data(exampleHiCDOCDataSet)
```

Format

S4 HiCDOCDataSet object with the following characteristics:

chromosomes 4 chromosomes: W, X, Y, Z

conditions 3 conditions: 1, 2, 3

replicates 3 replicates: R1, R2, R3

binSize A resolution of 137 bases

Value

A HiCDOCDataSet.

Examples

```
data(exampleHiCDOCDataSet)
exampleHiCDOCDataSet
```

exampleHiCDOCDataSetProcessed

Example HiCDOCDataSet, filtered, normalized and with compartements detected.

Description

A S4 HiCDOCDataSet object with 3 chromosomes, 3 conditions and 3 replicates. Can be retrieved by running : `data(exampleHiCDOCDataSet); set.seed(123); exampleHiCDOCDataSetProcessed <- HiCDOC(exampleHiCDOCDataSet)`

Usage

```
data(exampleHiCDOCDataSetProcessed)
```

Format

S4 HiCDOCDataSet object with the following characteristics:

chromosomes 4 chromosomes: X, Y, Z

conditions 3 conditions: 1, 2, 3

replicates 3 replicates: R1, R2, R3

binSize A resolution of 137 bases

Value

A HiCDOCDataSet, already filtered and normalized.

Examples

```
data(exampleHiCDOCDataSetProcessed)
exampleHiCDOCDataSetProcessed
```

filterSmallChromosomes

Filter small chromosomes.

Description

Removes chromosomes whose length (in number of positions) is smaller than the threshold.

Usage

```
filterSmallChromosomes(object, threshold = NULL)
```

Arguments

object	A HiCDOCDataSet.
threshold	The minimum length (number of positions) for a chromosome to be kept. Defaults to `object$smallChromosomeThreshold` which is originally set to `defaultHiCDOCParameters` = 100.

Value

A filtered HiCDOCDataSet.

See Also

filterSparseReplicates, filterWeakPositions, HiCDOC

Examples

```
data(exampleHiCDOCDataSet)
object <- exampleHiCDOCDataSet

chromosomes(object)
object <- filterSmallChromosomes(object)
chromosomes(object)
```

---

filterSparseReplicates

Filter sparse replicates.

Description

Removes chromosome replicates whose percentage of non-zero interactions is smaller than the threshold.

Usage

```
filterSparseReplicates(object, threshold = NULL)
```

Arguments

object	A HiCDOCDataSet.
threshold	The minimum percentage of non-zero interactions for a chromosome replicate to be kept. If a chromosome replicate's percentage of non-zero interactions is lower than this value, it is removed. Defaults to object$smallChromosomeThreshold which is originally set to defaultHiCDOCParameters$smallChromosomeThreshold = 30%.

Value

A filtered HiCDOCDataSet.

See Also

filterSmallChromosomes, filterWeakPositions, HiCDOC

Examples

```
data(exampleHiCDOCDataSet)
object <- exampleHiCDOCDataSet

object <- filterSparseReplicates(object)
```

`filterWeakPositions` *Filter weak positions.*

Description

Removes weak genomic positions whose interactions average is lower than the threshold.

Usage

```
filterWeakPositions(object, threshold = NULL)
```

Arguments

`object`	A `HiCDOCDataSet`.
`threshold`	The minimum average interaction for a position to be kept. If a position's average interaction with the entire chromosome is lower than this value in any of the replicates, it is removed from all replicates and conditions. Defaults to `object$smallChromosomeThreshold` which is originally set to `defaultHiCDOCParameters$small` = 1.

Details

Detects weak genomic positions in each replicate, and removes them from all replicates to guarantee comparability across conditions when calling `detectCompartments`.

Value

A filtered `HiCDOCDataSet`.

See Also

`filterSmallChromosomes`, `filterSparseReplicates`, `HiCDOC`

Examples

```
data(exampleHiCDOCDataSet)
object <- exampleHiCDOCDataSet

object <- filterWeakPositions(object)
```

HiCDOC *Default pipeline to run on the HiCDOC analysis.*

Description

Runs the default filtering, normalization, and computational steps on a HiCDOCDataSet. To learn more about HiCDOC, browse the vignette: browseVignettes(package = "HiCDOC").

Usage

```
HiCDOC(object, parallel = FALSE)
```

Arguments

object	A HiCDOCDataSet.
parallel	Whether or not to parallelize each step. Defaults to FALSE.

Details

HiCDOC **pipeline:** The HiCDOC pipeline has seven steps:

Three filtering steps:
- filterSmallChromosomes to filter out small chromosomes
- filterWeakPositions to filter out weak positions with very few interactions
- filterSparseReplicates to filter out sparse replicates with many null interactions

Three normalization steps:
- normalizeTechnicalBiases to normalize technical biases in each replicates
- normalizeBiologicalBiases to normalize biological biases in each replicate
- normalizeDistanceEffect to normalize the distance effect in each chromosome

One computational step:
- detectCompartments to detect compartments in each condition and find significant changes between conditions.

Parallel processing: The parallel version of HiCDOC uses the BiocParallel package. Before to call the function in parallel you should specify the parallel parameters such as:

- On Linux:
  `multiParam <- BiocParallel::MulticoreParam(workers = 10)`
- On Windows:
  `multiParam <- BiocParallel::SnowParam(workers = 10)`

And then you can register the parameters to be used by BiocParallel:

`BiocParallel::register(multiParam, default = TRUE)`

You should be aware that using MulticoreParam, reproducibility of the detectCompartments function using a RNGseed may not work. See this issue for more details.

Value

A HiCDOCDataSet with all slots filled.

See Also

HiCDOCDataSet, filterSmallChromosomes, filterWeakPositions, filterSparseReplicates, normalizeTechnicalBiases, normalizeBiologicalBiases, normalizeDistanceEffect, detectCompartments

Examples

```
data(exampleHiCDOCDataSet)
# Default HiCDOC pipeline
# Not printing loess warnings for example purpose.
# Results should be inspected if there is any.
suppressWarnings(
object <- HiCDOC(exampleHiCDOCDataSet)
)

# Equivalent to
if(FALSE){
    object <- filterSmallChromosomes(exampleHiCDOCDataSet)
    object <- filterSparseReplicates(object)
    object <- filterWeakPositions(object)
    object <- normalizeTechnicalBiases(object)
    object <- normalizeBiologicalBiases(object)
    object <- normalizeDistanceEffect(object)
    object <- detectCompartments(object)
}
```

`HiCDOCDataSet-class` `HiCDOCDataSet` *S4 class.*

Description

Data structure for a Hi-C experiment.

Details

An instance of `HiCDOCDataSet` describes a Hi-C experiment with slots for path(s) to input file(s), interactions, pipeline parameters defaulting to `defaultHiCDOCParameters`, and computation results. It can be constructed from 4 different types of data:

- Tabular files: see `HiCDOCDataSetFromTabular`
- (m)Cool files: see `HiCDOCDataSetFromCool`
- HiC files: see `HiCDOCDataSetFromHiC`
- HiC-Pro matrices and bed files: see `HiCDOCDataSetFromHiCPro` An example `HiCDOCDataSet` is also available, see `exampleHiCDOCDataSet`. The `HiCDOCDataSet` object can be explored using the appropriate accessors.

Accessors

The accessors for a HiCDOCDataset object are the following:

- `chromosomes` to retrieve the vector of chromosome names.
- `sampleConditions` to retrieve the vector of condition names, one for each sample.
- `sampleReplicates` to retrieve the vector of replicate names, one for each sample.

After the detection of compartments you can use this accessors:

- `compartments` returns a GenomicRange of the compartment of every position in every condition.

- concordances returns a GenomicRange of the significant compartment differences between conditions, and their p-values.
- differences returns a GenomicRange of the concordance (confidence in assigned compartment) of every position in every replicate.

See the HiCDOCDataSet-methods man page for more details on methods and accessors.

See Also

HiCDOC, exampleHiCDOCDataSet, HiCDOCDataSetFromTabular, HiCDOCDataSetFromCool, HiCDOCDataSetFromHiC, HiCDOCDataSetFromHiCPro

---

HiCDOCDataSet-methods *Methods to access a* HiCDOCDataSet *components.*

Description

Retrieve information and results from a HiCDOCDataSet.

Usage

```
## S4 method for signature 'HiCDOCDataSet'
chromosomes(object)

## S4 method for signature 'HiCDOCDataSet'
sampleConditions(object)

## S4 method for signature 'HiCDOCDataSet'
sampleReplicates(object)

## S4 method for signature 'HiCDOCDataSet'
compartments(object, passChecks = TRUE)

## S4 method for signature 'HiCDOCDataSet'
differences(object, threshold = NULL)

## S4 method for signature 'HiCDOCDataSet'
concordances(object, passChecks = TRUE)

## S4 method for signature 'HiCDOCDataSet'
show(object)
```

Arguments

object	a HiCDOCDataSet object
passChecks	logical. Display only the concordances/compartments for the chromosomes passing sanity checks.
threshold	a numeric value between 0 and 1. If no threshold, all the differences will be printed even the non significant ones. Otherwise the differences printed are filtered to show the ones with an adjusted p-value <= threshold.

Value

A character vector (for `chromosomes`, `sampleConditions`, `sampleReplicates`), or a GRanges object (for `compartments`, `concordances`, `differences`).

Functions

- `chromosomes()`: Retrieves the vector of chromosome names.
- `sampleConditions()`: Retrieves the vector of condition names, one for each sample.
- `sampleReplicates()`: Retrieves the vector of replicate names, one for each sample.
- `compartments()`: Retrieves a `GenomicRange` of the compartment of every position in every condition.
- `differences()`: Retrieves a `GenomicRange` of the significant compartment differences between conditions, and their p-values.
- `concordances()`: Retrieves a `GenomicRange` of the concordance (confidence in assigned compartment) of every position in every replicate.

Examples

```
# Load an example dataset already processed
# (i.e. after the detection of compartments)
data(exampleHiCDOCDataSetProcessed)

exampleHiCDOCDataSetProcessed
chromosomes(exampleHiCDOCDataSetProcessed)
sampleConditions(exampleHiCDOCDataSetProcessed)
sampleReplicates(exampleHiCDOCDataSetProcessed)
compartments(exampleHiCDOCDataSetProcessed)
differences(exampleHiCDOCDataSetProcessed)
concordances(exampleHiCDOCDataSetProcessed)
```

HiCDOCDataSet-parameters

Access the parameters of a `HiCDOCDataSet`.

Description

Retrieves or sets parameters used for filtering, normalization, and prediciton of compartments.

Usage

```
defaultHiCDOCParameters

## S4 method for signature 'HiCDOCDataSet'
parameters(object)

## S4 replacement method for signature 'HiCDOCDataSet'
parameters(object) <- value
```

Arguments

object	A HiCDOCDataSet.
value	a named list containing the names and valued of the parameters to change (see Details).

Format

An object of class list of length 9.

Details

A HiCDOCDataSet's parameters are automatically set to default values retrieved from defaultHiCDOCParameters. They are accessed by filtering, normalization, and compartment detection functions. If those functions are called with custom arguments, the object's parameters are updated to record the actual parameters used. If the object's parameters are customized before calling the functions, the custom parameters will be used.

See filterSmallChromosomes, filterSparseReplicates, filterWeakPositions, normalizeDistanceEffect, and detectCompartments, for details on how these parameters are used.

All parameters are listed here::

smallChromosomeThreshold The minimum length (number of positions) for a chromosome to be kept when filtering with filterSmallChromosomes. Defaults to defaultHiCDOCParameters$smallChromoso = 100.

sparseReplicateThreshold The minimum percentage of non-zero interactions for a chromosome replicate to be kept when filtering with filterSparseReplicates. If a chromosome replicate's percentage of non-zero interactions is lower than this value, it is removed. Defaults to defaultHiCDOCParameters$smallChromosomeThreshold = 30

weakPositionThreshold The minimum average interaction for a position to be kept when filtering with filterWeakPositions. If a position's average interaction with the entire chromosome is lower than this value in any of the replicates, it is removed from all replicates and conditions. Defaults to defaultHiCDOCParameters$smallChromosomeThreshold = 1.

cyclicLoessSpan The span for cyclic loess normalization used in normalizeTechnicalBiases. This value is passed to multiHiCcompare::cyclic_loess. Defaults to NA indicating that span will be automatically calculated using generalized cross validation. For large dataset, it is highly recommended to set this value to reduce computing time and necessary memory.

loessSampleSize The number of positions used as a sample to estimate the effect of distance on proportion of interactions when normalizing with normalizeDistanceEffect. Defaults to defaultHiCDOCParameters$loessSampleSize = 20000.

kMeansDelta The convergence stop criterion for the clustering when detecting compartments with detectCompartments. When the centroids' distances between two iterations is lower than this value, the clustering stops. Defaults to defaultHiCDOCParameters$kMeansDelta = 0.0001.

kMeansIterations The maximum number of iterations during clustering when detecting compartments with detectCompartments. Defaults to defaultHiCDOCParameters$kMeansIterations = 50.

kMeansRestarts The amount of times the clustering is restarted when detecting compartments with detectCompartments. For each restart, the clustering iterates until convergence or reaching the maximum number of iterations. The clustering that minimizes inner-cluster variance is selected. Defaults to defaultHiCDOCParameters$kMeansRestarts = 20.

PC1CheckThreshold The minimum percentage of variance that should be explained by the first principal component of centroids to pass sanity check. Defaults to defaultHiCDOCParameters$PC1CheckThresh = 0.75

Examples

```
data(exampleHiCDOCDataSet)

# Retrieve parameters
parameters(exampleHiCDOCDataSet)

# Set parameters
parameters(exampleHiCDOCDataSet) <- list("smallChromosomeThreshold" = 50)
parameters(exampleHiCDOCDataSet) <- list(
    "weakPositionThreshold" = 10,
    "kMeansRestarts" = 30
)
```

`HiCDOCDataSetFromCool` `HiCDOCDataSet` *constructor from Cool files.*

Description

Constructs a `HiCDOCDataSet` from a set of `.cool` or `.mcool` files.

Usage

```
HiCDOCDataSetFromCool(paths, replicates, conditions, binSize = NA)
```

Arguments

`paths`	A vector of paths to `.cool` or `.mcool` files.
`replicates`	A vector of replicate names repeated along the conditions.
`conditions`	A vector of condition names repeated along the replicates.
`binSize`	The resolution (span of each position in number of bases). Optionally provided to select the appropriate resolution in `.mcool` files. Defaults to NULL.

Value

A `HiCDOCDataSet`.

Examples

```
## Not run:
    # Path to each file
    paths = c(
      'path/to/condition-1.replicate-1.cool',
      'path/to/condition-1.replicate-2.cool',
      'path/to/condition-2.replicate-1.cool',
      'path/to/condition-2.replicate-2.cool',
      'path/to/condition-3.replicate-1.cool'
    )

    # Replicate and condition of each file. Can be names instead of numbers.
    replicates <- c(1, 2, 1, 2, 1)
    conditions <- c(1, 1, 2, 2, 3)
```

```
    # Resolution to select in .mcool files
    binSize = 500000

    # Instantiation of data set
    object <- HiCDOCDataSetFromCool(
      paths,
      replicates = replicates,
      conditions = conditions,
      binSize = binSize # Specified for .mcool files.
    )

## End(Not run)
```

`HiCDOCDataSetFromHiC` `HiCDOCDataSet` *constructor from HiC files.*

Description

Constructs a `HiCDOCDataSet` from a set of `.hic` files.

Usage

```
HiCDOCDataSetFromHiC(paths, replicates, conditions, binSize)
```

Arguments

`paths`	A vector of paths to `.hic` files.
`replicates`	A vector of replicate names repeated along the conditions.
`conditions`	A vector of condition names repeated along the replicates.
`binSize`	The resolution (span of each position in number of bases) to select within the `.hic` files.

Value

A `HiCDOCDataSet`.

Examples

```
## Not run:
    #' # Path to each file
    paths = c(
      'path/to/condition-1.replicate-1.hic',
      'path/to/condition-1.replicate-2.hic',
      'path/to/condition-2.replicate-1.hic',
      'path/to/condition-2.replicate-2.hic',
      'path/to/condition-3.replicate-1.hic'
    )

    # Replicate and condition of each file. Can be names instead of numbers.
    replicates <- c(1, 2, 1, 2, 1)
    conditions <- c(1, 1, 2, 2, 3)
```

```
    # Resolution to select
    binSize <- 500000

    # Instantiation of data set
    hic.experiment <- HiCDOCDataSetFromHiC(
      paths,
      replicates = replicates,
      conditions = conditions,
      binSize = binSize
    )

## End(Not run)
```

`HiCDOCDataSetFromHiCPro` *HiCDOCDataSet constructor from HiC-Pro files.*

Description

Constructs a `HiCDOCDataSet` from a set of HiC-Pro generated files.

Usage

```
HiCDOCDataSetFromHiCPro(matrixPaths, bedPaths, replicates, conditions)
```

Arguments

`matrixPaths`	A vector of paths to HiC-Pro matrix files.
`bedPaths`	A vector of paths to HiC-Pro bed files.
`replicates`	A vector of replicate names repeated along the conditions.
`conditions`	A vector of condition names repeated along the replicates.

Value

A `HiCDOCDataSet`.

Examples

```
## Not run:
    # Path to each matrix file
    matrixPaths = c(
      'path/to/condition-1.replicate-1.matrix',
      'path/to/condition-1.replicate-2.matrix',
      'path/to/condition-2.replicate-1.matrix',
      'path/to/condition-2.replicate-2.matrix',
      'path/to/condition-3.replicate-1.matrix'
    )

    # Path to each bed file
    bedPaths = c(
      'path/to/condition-1.replicate-1.bed',
      'path/to/condition-1.replicate-2.bed',
```

```
  'path/to/condition-2.replicate-1.bed',
  'path/to/condition-2.replicate-2.bed',
  'path/to/condition-3.replicate-1.bed'
)

# Replicate and condition of each file. Can be names instead of numbers.
replicates <- c(1, 2, 1, 2, 1)
conditions <- c(1, 1, 2, 2, 3)

# Instantiation of data set
hic.experiment <- HiCDOCDataSetFromHiCPro(
  matrixPaths = matrixPaths,
  bedPaths = bedPaths,
  replicates = replicates,
  conditions = conditions
)

## End(Not run)
```

`HiCDOCDataSetFromTabular` *HiCDOCDataSet constructor from a tabular file.*

Description

Constructs a `HiCDOCDataSet` from a tabular file.

Usage

```
HiCDOCDataSetFromTabular(path, sep = '\t')
```

Arguments

`path`	A path to a tabular file.
`sep`	The separator of the tabular file. Default to tabulation.

Details

Accepts a tabular file with `chromosome`, `position 1`, `position 2`, and multiple replicate columns listing interaction counts. Null interactions do not have to be listed. Values must be separated by tabulations. The header must be `chromosome position 1 position 2 x.y x.y x.y ...` with `x` replaced by condition names and `y` replaced by replicate names.

Value

A `HiCDOCDataSet`.

Examples

```
path <- system.file("extdata", "liver_18_10M_500000.tsv", package = "HiCDOC")
object <- HiCDOCDataSetFromTabular(path, sep = '\t')
```

normalizeBiologicalBiases *Normalize biological biases.*

Description

Normalizes biological biases such as GC content and repeated regions. Uses the Knight-Ruiz balancing algorithm to transform interaction matrices into doubly stochastic matrices, with sum of each row and sum of each column equal to 1.

Usage

```
normalizeBiologicalBiases(object, parallel = FALSE)
```

Arguments

object	A HiCDOCDataSet.
parallel	Should the normalization be run in parallel mode? Default to FALSE.

Value

A HiCDOCDataSet with normalized interactions.

See Also

filterSparseReplicates, filterWeakPositions, normalizeTechnicalBiases, normalizeDistanceEffect, HiCDOC

Examples

```
data(exampleHiCDOCDataSet)
object <- exampleHiCDOCDataSet
object <- filterSparseReplicates(object)
object <- filterWeakPositions(object)
object <- normalizeBiologicalBiases(object)
```

normalizeDistanceEffect *Normalize distance effect.*

Description

Normalizes interactions by their "expected" value relative to the distance that separates their positions. The "expected" values are estimated with a loess regression on the proportion of interactions for each distance.

Usage

```
normalizeDistanceEffect(object, loessSampleSize = NULL, parallel = FALSE)
```

Arguments

object	A HiCDOCDataSet.
loessSampleSize	The number of positions used as a sample to estimate the effect of distance on proportion of interactions. Defaults to object$loessSampleSize which is originally set to defaultHiCDOCParameters$loessSampleSize = 20000.
parallel	Should the normalization be run in parallel mode? Default to FALSE.

Value

A HiCDOCDataSet with normalized interactions.

See Also

normalizeTechnicalBiases, normalizeBiologicalBiases, HiCDOC

Examples

```
data(exampleHiCDOCDataSet)
object <- normalizeDistanceEffect(exampleHiCDOCDataSet)
```

---

normalizeTechnicalBiases

Normalize technical biases.

Description

Normalizes technical biases such as sequencing depth by using a cyclic loess to recursively normalize each pair of interaction matrices. Depends on multiHiCcompare.

Usage

```
normalizeTechnicalBiases(object, parallel = FALSE, cyclicLoessSpan = NULL)
```

Arguments

object	A HiCDOCDataSet.
parallel	Logical. Whether or not to parallelize the processing. Defaults to FALSE
cyclicLoessSpan	A numeric value in between 0 and 1. The span for cyclic loess normalization. This value is passed to multiHiCcompare::cyclic_loess. Defaults to NULL, NULL indicates that the value of parameters(object)$cyclicLoessSpan will be used. If this value is NA, the span will be automatically calculated using generalized cross validation. **For large dataset, it is highly recommended to set this value to reduce computing time and necessary memory.**

Details

Parallel processing: If `parallel = TRUE`, the function `cyclic_loess` is launched in parallel mode, using `bplapply` function. Before to call the function in parallel you should specify the parallel parameters such as:

- On Linux:
  `multiParam <- BiocParallel::MulticoreParam(workers = 10)`
- On Windows:
  `multiParam <- BiocParallel::SnowParam(workers = 10)`

And then you can register the parameters to be used by BiocParallel:

`BiocParallel::register(multiParam, default = TRUE)`

Value

A `HiCDOCDataSet` with normalized interactions.

See Also

`filterSparseReplicates`, `filterWeakPositions`, `normalizeBiologicalBiases`, `normalizeDistanceEffect`, `HiCDOC`

Examples

```
data(exampleHiCDOCDataSet)
object <- filterSmallChromosomes(exampleHiCDOCDataSet)
object <- filterSparseReplicates(object)
object <- filterWeakPositions(object)
# Not printing loess warnings for example purpose.
# Results should be inspected if there is any.
suppressWarnings(
    object <- normalizeTechnicalBiases(object)
)
```

---

plotCentroids *Plot centroids.*

Description

Plots the result of the PCA on the compartments' centroids.

Usage

```
plotCentroids(object, chromosome, size = 2, checks = TRUE)
```

Arguments

`object`	A `HiCDOCDataSet`.
`chromosome`	A chromosome name or index in `chromosomes(object)`.
`size`	Size of each point. Defaults to 2.
`checks`	Whether or not to add sanity checks messages on centroids. Default to TRUE.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotCentroids(exampleHiCDOCDataSetProcessed, chromosome = 1)
```

`plotCompartmentChanges` *Plot compartment changes.*

Description

Plots the predicted compartments, along with their concordance in each replicate, and significant changes between experiment conditions.

Usage

```
plotCompartmentChanges(
  object,
  chromosome,
  threshold = 0.05,
  xlim = NULL,
  points = FALSE,
  checks = TRUE,
  colour = "gray90"
)
```

Arguments

`object`	A `HiCDOCDataSet`.
`chromosome`	A chromosome name or index in `chromosomes(object)`.
`threshold`	Significance threshold for the compartment changes. Defaults to 0.05.
`xlim`	A vector of the minimum and maximum positions to display. If NULL, displays all positions. Defaults to NULL.
`points`	Whether or not to add points to the concordances. Defaults to FALSE.
`checks`	Whether or not to add sanity checks messages. Default to TRUE.
`colour`	Border color for the compartments. Default to ‘gray90‘. ‘NA‘ means no border.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotCompartmentChanges(exampleHiCDOCDataSetProcessed, chromosome = 1)
```

plotCompartments *Plot A/B compartments.*

Description

Plots the predicted compartments in each experiment condition.

Usage

```
plotCompartments(object, chromosome, xlim = NULL, colour = "gray90")
```

Arguments

object	A HiCDOCDataSet.
chromosome	A chromosome name or index in chromosomes(object).
xlim	A vector of the minimum and maximum positions to display. If NULL, displays all positions. Defaults to NULL.
colour	Border color for the compartments. Default to 'gray90'. 'NA' means no border.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotCompartments(exampleHiCDOCDataSetProcessed, chromosome = 1)
```

plotConcordanceDifferences
Plot the distribution of concordance differences.

Description

Plots the distribution of concordance differences, which are the differences between concordances of each pair of replicates from different conditions. A concordance can be understood as a confidence in a genomic position's assigned compartment. Mathematically, it is the log ratio of a genomic position's distance to each compartment's centroid, normalized by the distance between both centroids, and min-maxed to a [-1,1] interval.

Usage

```
plotConcordanceDifferences(object)
```

Arguments

object	A HiCDOCDataSet.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotConcordanceDifferences(exampleHiCDOCDataSetProcessed)
```

`plotConcordances` *Plot concordances.*

Description

Plots the concordances of each replicate in each experiment condition. A concordance can be understood as a confidence in a genomic position's assigned compartment. Mathematically, it is the log ratio of a genomic position's distance to each compartment's centroid, normalized by the distance between both centroids, and min-maxed to a [-1,1] interval.

Usage

```
plotConcordances(
  object,
  chromosome,
  xlim = NULL,
  threshold = 0.05,
  points = FALSE
)
```

Arguments

`object`	A `HiCDOCDataSet`.
`chromosome`	A chromosome name or index in `chromosomes(object)`.
`xlim`	A vector of the minimum and maximum positions to display. If NULL, displays all positions. Defaults to NULL.
`threshold`	Significance threshold for the compartment changes. Defaults to 0.05.
`points`	Whether or not to add points to the concordances. Defaults to FALSE.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotConcordances(exampleHiCDOCDataSetProcessed, chromosome = 1)
```

plotDistanceEffect *Plot the distance effect.*

Description

Plots the distance effect on proportion of interactions. Each point is a cell in the interaction matrix, such that the x-axis is the distance with respect to the diagonal, the y-axis is number of counts. Dots are binned.

Usage

```
plotDistanceEffect(
  object,
  chromosome = NULL,
  transformX = "identity",
  transformY = "identity"
)
```

Arguments

object	A HiCDOCDataSet.
chromosome	Name (character) or index of the chromosome, if the plot should be restricted to only one chromosome. Default to NULL.
transformX	Transformation of the X axis. Default to "identity". See scale_x_continuous for other accepted values.
transformY	Transformation of the Y axis. Default to "identity". See scale_y_continuous for other accepted values.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSet)
plotDistanceEffect(exampleHiCDOCDataSet)
```

plotInteractions *Plot interaction matrices.*

Description

Plots the interaction matrices as heatmaps.

Usage

```
plotInteractions(
  object,
  chromosome,
  transform = NULL,
  colours = c(low = "#2c7bb6", mid = "#ffffbf", high = "#d7191c"),
  midpoint = 0
)
```

Arguments

object	A HiCDOCDataSet.
chromosome	A chromosome name or index in chromosomes(object).
transform	Transformation of the color scale. Default to NULL (no transformation). See scale_fill_gradient2 for other accepted values.
colours	A character vector colours of length 3 to use for the gradient. See scale_fill_gradient2 for more info. Defaults to c("low"=#2c7bb6", "mid"=#ffffbf", "high"="#d7191c").
midpoint	midpoint value to be passed to scale_fill_gradient2. Default to 0.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSet)
plotInteractions(exampleHiCDOCDataSet, chromosome = 1)
```

---

plotSelfInteractionRatios

Plot boxplots of self interaction ratios.

Description

Plots the boxplots of self interaction ratios, which are the differences between self interaction and median of other interactions for each genomic position. Since the A compartment is open with more interactions overall, it is assumed that self interaction ratios in compartment A are smaller than in compartment B.

Usage

```
plotSelfInteractionRatios(object, chromosome, checks = TRUE)
```

Arguments

object	A HiCDOCDataSet.
chromosome	A chromosome name or index in chromosomes(object). A HiCDOCDataSet.
checks	Logical. Should sanity checks messages be printed on plot ? Default to TRUE.

Value

A ggplot.

Examples

```
data(exampleHiCDOCDataSetProcessed)
plotSelfInteractionRatios(exampleHiCDOCDataSetProcessed, chromosome = 1)
```

`reduceHiCDOCDataSet` *Reduce a* `HiCDOCDataSet`.

Description

Reduces a `HiCDOCDataSet` by keeping only given chromosomes, conditions, or replicates.

Usage

```
reduceHiCDOCDataSet(
  object,
  chromosomes = NULL,
  conditions = NULL,
  replicates = NULL,
  dropLevels = TRUE
)
```

Arguments

`object`	A `HiCDOCDataSet`.
`chromosomes`	The chromosome names or indices in `chromosomes(object)` to keep. Defaults to NULL.
`conditions`	The condition names in `sampleConditions(object)` to keep. Defaults to NULL.
`replicates`	The replicate names in `sampleReplicates(object)` to keep. Defaults to NULL.
`dropLevels`	Whether or not to also remove unused factor levels after filtering. Should be set to FALSE if the reduced objects are meant to be re-combined later. Defaults to TRUE.

Value

A reduced `HiCDOCDataSet`.

Examples

```
data(exampleHiCDOCDataSet)
reduced <- reduceHiCDOCDataSet(exampleHiCDOCDataSet, chromosomes = c(1, 2))
```

Index