Package ‘HiCaptuRe’

April 6, 2026

Type Package

Title HiCaptuRe: Manipulating and integrating Capture Hi-C data

Version 1.0.0

URL https://github.com/LaureTomas/HiCaptuRe

Description Capture Hi-C is a set of techniques that enable the detection of genomic interactions involving regions of interest, known as baits. By focusing on selected loci, these approaches reduce sequencing costs while maintaining high resolution at the level of restriction fragments. HiCaptuRe provides tools to import, annotate, manipulate, and export Capture Hi-C data. The package accounts for the specific structure of bait–otherEnd interactions, facilitates integration with other omics datasets, and enables comparison across samples and conditions.

BugReports https://github.com/LaureTomas/HiCaptuRe/issues

License GPL-3

Encoding UTF-8

LazyData false

Depends R (>= 4.5.0)

Imports Biostrings, BSgenome, cli, data.table, dplyr, GenomeInfoDb, GenomicInteractions, GenomicRanges, InteractionSet, ggplot2, ggpubr, ggVennDiagram, gplots, igraph, IRanges, memoise, methods, S4Vectors, stringr, tibble, tidyr, UpSetR, utils

RoxygenNote 7.3.2

Suggests BSgenome.Hsapiens.NCBI.GRCh38, knitr, rmarkdown, DT, testthat, BiocStyle, kableExtra

biocViews Epigenetics, HiC, Sequencing, DataImport, Software

VignetteBuilder knitr

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

git_branch RELEASE_3_22

git_last_commit 0a37211

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-04-05

Author Laureano Tomas-Daza [aut, cre] (ORCID: <https://orcid.org/0000-0002-6127-1694>)

Maintainer Laureano Tomas-Daza <`lauretomas@gmail.com`>

Contents

`HiCaptuRe-package` *HiCaptuRe: HiCaptuRe: Manipulating and integrating Capture Hi-C data*

Description

Capture Hi-C is a set of techniques that enable the detection of genomic interactions involving regions of interest, known as baits. By focusing on selected loci, these approaches reduce sequencing costs while maintaining high resolution at the level of restriction fragments. HiCaptuRe provides tools to import, annotate, manipulate, and export Capture Hi-C data. The package accounts for the specific structure of bait–otherEnd interactions, facilitates integration with other omics datasets, and enables comparison across samples and conditions.

Author(s)

Maintainer: Laureano Tomas-Daza <`lauretomas@gmail.com`> (ORCID)

See Also

Useful links:

- `https://github.com/LaureTomas/HiCaptuRe`
- Report bugs at `https://github.com/LaureTomas/HiCaptuRe/issues`

`.annotate_BOE` *Annotates interactions into B_B or B_OE*

Description

This internal function annotate a HiCaptuRe object based on the presence or not of annotation in a given restriction fragment

Usage

```
.annotate_BOE(interactions)
```

Arguments

`interactions`	HiCaptuRe object from `load_interactions`

Value

HiCaptuRe object with the column int updated

`annotate_interactions` *Annotates HiCaptuRe object from load_interactions*

Description

This function annotate a HiCaptuRe object from load_interactions, based on a given annotations file

Usage

```
annotate_interactions(interactions, annotation, ...)
```

Arguments

`interactions`	HiCaptuRe object from `load_interactions`
`annotation`	full path to annotations file or a dataframe with 5 columns: chr, start, end, fragmentID, annotation
`...`	additional parameters for fread

Value

HiCaptuRe object annotated, with columns bait_1 and bait_2 substituted based on the given annotation

Examples

```
annotation <- system.file("extdata", "annotation_example.txt", package = "HiCaptuRe")
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
interactions <- annotate_interactions(interactions = interactions, annotation = annotation)
```

`constants` *Column name constants for format detection and export*

Description

‘.IBED_COLS‘ and ‘.PEAKMATRIX_COLS‘ store canonical column names for ibed and peakmatrix formats. These are used in both ‘.detect_format()‘ and export functions to avoid repetition and ensure consistency between import and export.

Usage

```
.IBED_COLS
```

Format

An object of class `character` of length 10.

`digest_genome` *Digest genome for a specific restriction enzyme*

Description

This function takes a genome installed and generates its digest for a given restriction enzyme

Usage

```
digest_genome(
  genome = "GRCh38",
  RE_name = "HindIII",
  motif = NULL,
  cut_position = NULL,
  select_chr = c(seq_len(22), "X", "Y"),
  PAR_mask = TRUE,
  PAR_file = NULL,
  ...
)
```

Arguments

`genome`	character with the name of the genome version
`RE_name`	name of the restriction enzyme
`motif`	recognition motif of the restriction enzyme
`cut_position`	cut position of the restriction enzyme inside the motif
`select_chr`	a character vector containing the specific chromosomes to used from this genome, if NULL all chromosomes will be used
`PAR_mask`	a logical value where the Y chromosome should exclude the Pseudoautosomical Regions (PAR) or not
`PAR_file`	a full path to a file containing the coordinates of Y chromosome PAR with at least 3 columns with header: seqnames, start, end
`...`	extra arguments for read.table

Value

list object with 2 elements: a dataframe with the digested genome, and a the parameters used for the digestion.

Note

The package provides for a PAR coordinates file only for Homo sapiens for the genome version 38

The package provides the motives and cut positions for several restriction enzymes (HindIII, MboI, DpnII, EcoRI, BamHI)

Examples

```
digest <- digest_genome(genome = "GRCh38", RE_name = "HindIII", select_chr = "19")
```

`distance_summary` *Computes the number of interactions by distance*

Description

This function computes the number of interactions by distance for a given GenomicInteractions object

Usage

```
distance_summary(interactions, breaks = seq(0, 10^6, 10^5), sample = "sample")
```

Arguments

`interactions`	GenomicInteractions object from `load_interactions`
`breaks`	vector with breaks for split distances
`sample`	character variable with the name of the sample

Value

list with 2 tables: short_int_dist_table and long_int_dist_table, with the number of short interactions and long interactions respectively

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions1 <- load_interactions(ibed1, select_chr = "19")
df <- distance_summary(interactions = interactions1)
```

`export_interactions` *Export interactions in the desired output format*

Description

This function exports interactions in different formats

Usage

```
export_interactions(
  interactions,
  file,
  format = "ibed",
  over.write = FALSE,
  cutoff = 5,
  parameters = FALSE
)
```

Arguments

`interactions`	GenomicInteractions object from `load_interactions`
`file`	full path to desired output file (ibed, peakmatrix, washU, washUold, cytoscape, bedpe)
`format`	type of output format (ibed, peakmatrix, washU, washUold, cytoscape, bedpe, seqmonk)
`over.write`	TRUE/FALSE to over write the output file
`cutoff`	Chicago score cutoff to export interactions
`parameters`	TRUE/FALSE to also export the parameters of the given object

Value

tibble object with the ibed table and save it in the desired output file

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
export_interactions(interactions = interactions, file = tempfile(), format = "ibed", over.write = TRUE)
```

`getters` *Internal functions to access data held in a HiCaptuRe object.*

Description

Use these functions to access data stored in each of the slots of a HiCapture object.

Usage

```
getParameters(x)

## S4 method for signature 'HiCaptuRe'
getParameters(x)

getByBaits(x)

## S4 method for signature 'HiCaptuRe'
getByBaits(x)

getByRegions(x)

## S4 method for signature 'HiCaptuRe'
getByRegions(x)
```

Arguments

`x`	A HiCaptuRe object

Value

getParameters(x) A named list of named vectors with all parameters used.

getByBaits(x) A list of tibbles with bait-centric summaries.

getByRegions(x) A list of ‘GRanges‘ with region-centric summaries.

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
getParameters(interactions)

baits <- c("ENST00000332235", "ENST00000516525")
interactions_baits <- interactionsByBaits(interactions = interactions, baits = baits)
getByBaits(interactions_baits)

regions <- GenomicRanges::GRanges(seqnames = 19, ranges = IRanges::IRanges(start = c(500000, 1000000), end = c(
interactions_regions <- interactionsByRegions(interactions = interactions, regions = regions)
getByRegions(interactions_regions)
```

`HiCaptuRe-class` *HiCaptuRe Class*

Description

A S4 class to represent interactions between genomic regions

Slots

`parameters` List of parameters used to create the object and subsequence analysis

`ByBaits` List of Baits used by interactionsByBaits.

`ByRegions` List of Regions used by interactionsByRegions

Note

This class contains a GenomicInteractions object inside therefore all methods available to it can be used. This type of object should be created through the function load_interactions

---

`interactionsByBaits` *Filters HiCaptuRe object by baits*

Description

This function filters a HiCaptuRe object from load_interactions by a set of given bait(s)

Usage

```
interactionsByBaits(interactions, baits, sep = ",", invert = FALSE)
```

Arguments

`interactions`	a HiCaptuRe object
`baits`	character vector containing bait names
`sep`	character separating baits names when several baits in same fragment
`invert`	TRUE/FALSE if need those interactions that do NOT contain the given baits

Value

The returned object includes a filtered set of interactions and updates the ByBaits slot with a tibble summarizing bait-wise interaction statistics (number of interactions, other ends, etc.). Baits that are not present in the interaction data will have empty statistics.

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
baits <- c("ENST00000332235", "ENST00000516525")
interactions_baits <- interactionsByBaits(interactions = interactions, baits = baits)
```

`interactionsByRegions` *Filters HiCaptuRe object by overlaping regions*

Description

This function filters a HiCaptuRe object from load_interactions by overlaping regions

Usage

```
interactionsByRegions(
  interactions,
  regions,
  chr = NULL,
  start = NULL,
  end = NULL,
  invert = FALSE
)
```

Arguments

`interactions`	HiCaptuRe object
`regions`	full path to regions file (bed format) or a GRanges object
`chr`	column name of chromosome values
`start`	column name of start positions
`end`	column name of end positions
`invert`	TRUE/FALSE if need those interactions that do NOT overlaps with any regions

Value

HiCaptuRe object filtered by regions, by default with additional columns regarding overlap on each node. If invert=T no additional columns. And an additional slot ByRegions with region-centric statistics

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
regions <- GenomicRanges::GRanges(seqnames = 19, ranges = IRanges::IRanges(start = c(500000, 1000000), end = c(
interactions_regions <- interactionsByRegions(interactions = interactions, regions = regions)
```

`intersect_interactions` *Computes the intersect between a list of interactions objects*

Description

This function computes all the possible intersects for a given list of HiCaptuRe objects

Usage

```
intersect_interactions(interactions_list, distance.boxplot = FALSE, ...)
```

Arguments

`interactions_list`	list of HiCaptuRe objects from `load_interactions`
`distance.boxplot`	logical, by default FALSE, plot a boxplot of log10 distance of each intersection in the upset plot
`...`	extra arguments for UpSetR

Value

A list with:

intersections A named list of intersected HiCaptuRe objects

upset An UpSetR plot

venn A ggVennDiagram plot (NULL if > 7 sets)

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions1 <- load_interactions(ibed1, select_chr = "19")
ibed2 <- system.file("extdata", "ibed2_example.zip", package = "HiCaptuRe")
interactions2 <- load_interactions(ibed2, select_chr = "19")
interactions_list <- list(ibed1 = interactions1, ibed2 = interactions2)
intersection <- intersect_interactions(interactions_list = interactions_list)
```

`load_interactions` *Loads interaction file into GenomicInteractions Object*

Description

This function loads interaction files from Chicago R package into a GenomicInteractions Object, and remove possible duplicated interactions

Usage

```
load_interactions(file, sep = "\t", ...)
```

Arguments

`file`	full path to the interaction file (seqmonk, ibed, washU)
`sep`	separator to read the file
`...`	arguments to pass to digest_genome

Value

HiCaptuRe object

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(ibed1, select_chr = "19")
```

`peakmatrix2list` *Creates a list of HiCaptuRe objects based on a peakmatrix*

Description

This function creates a list of HiCaptuRe objects separating all the samples included on a peakmatrix.

Usage

```
peakmatrix2list(peakmatrix, cutoff = 5)
```

Arguments

`peakmatrix`	HiCaptuRe object of a peakmatrix
`cutoff`	Chicago score cut-off to filter interactions

Value

A named list of `HiCaptuRe` objects, each corresponding to a sample (inferred from CS_* columns in the peakmatrix).

Examples

```
peakmatrix <- system.file("extdata", "peakmatrix_example.zip", package = "HiCaptuRe")
interactions <- load_interactions(peakmatrix, select_chr = "19")
interactions_list <- peakmatrix2list(peakmatrix = interactions)
```

`plot_distance_summary` *Generate the plot based on distance output*

Description

This function plots the distances by different approaches for the output of distance_summary

Usage

```
plot_distance_summary(distances, type_of_value = "absolute")
```

Arguments

`distances`	dataframes output from `distance_summary`
`type_of_value`	an element from: absolute, by_int_type, by_total

Details

The type_of_value argument could be: Absolute: plots the Absolute number of interactions. By_int_type: plots the number of interactions divided by the total number of interactions of each type. By_total: plots the number of interactions divided by the total number of interactions

Value

list with the plots

Examples

```
ibed1 <- system.file("extdata", "ibed1_example.zip", package = "HiCaptuRe")
interactions1 <- load_interactions(ibed1, select_chr = "19")
df <- distance_summary(interactions = interactions1)
plots <- plot_distance_summary(distances = df)
```

`setters` *Functions to set data held in a HiCaptuRe object.*

Description

Use these functions to set data stored in each of the slots of a HiCapture object.

Usage

```
.setParameters(x, y)

## S4 method for signature 'HiCaptuRe,list'
.setParameters(x, y)

.setByBaits(x, y)
```

```
## S4 method for signature 'HiCaptuRe,list'
.setByBaits(x, y)

.setByRegions(x, y)

## S4 method for signature 'HiCaptuRe,list'
.setByRegions(x, y)
```

Arguments

`x`	A HiCaptuRe object
`y`	Data to add (parameters, tibble with bait-centric information or GRanges with region-centric)

Value

A HiCaptuRe object

Index