Package: hicream (via r-universe)

November 27, 2026

Title HIC diffeREntial Analysis Method

Version 0.1.0

Date 2026-11-25

Maintainer Elise Jorge <elise.jorge@inrae.fr>

Description Perform Hi-C data differential analysis based on pixel-level differential analysis and a post hoc inference strategy to quantify signal in clusters of pixels. Clusters of pixels are obtained through a connectivity-constrained two-dimensional hierarchical clustering. The detailed description of the method is provided in <doi:10.1093/bioinformatics/btag631>.

License GPL (>= 3)

URL https://scales.pages-forge.inrae.fr/hicream/, https://forge.inrae.fr/scales/hicream

BugReports https://forge.inrae.fr/scales/hicream/-/issues

Depends R (>= 4.0.0), reticulate

SystemRequirements Python (>= 3.9)

Imports adjclust, auk, BiocGenerics, csaw, diffHic, dplyr, edgeR, GenomeInfoDb, GenomicRanges, ggplot2, grid, InteractionSet, limma, Matrix, methods, reshape2, rlang, S4Vectors, stats, SummarizedExperiment, tidyselect, utils, viridis

Suggests knitr, rmarkdown

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Elise Jorge [aut, cre], Sylvain Foissac [aut] (ORCID: <https://orcid.org/0000-0002-2631-5356>), Toby Hocking [aut] (ORCID: <https://orcid.org/0000-0002-3146-0865>), Pierre Neuvial [aut] (ORCID: <https://orcid.org/0000-0003-3584-9998>),

Nathalie Vialaneix [aut] (ORCID:
<https://orcid.org/0000-0003-1156-0639>), Gilles Blanchard
[ctb], Guillermo Durand [ctb] (ORCID:
<https://orcid.org/0000-0003-4056-5631>), Nicolas
Enjalbert-Courrech [ctb], Etienne Roquain [ctb]

Config/pak/sysreqs make libbz2-dev libicu-dev liblzma-dev libpng-dev libxml2-dev libssl-dev python3 libx11-dev xz-utils zlib1g-dev

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

Date/Publication 2026-11-27 02:57:54 UTC

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

RemoteRef HEAD

RemoteSha 3f4fc7ce0fbb62d4443f336bd5c11bc02d26b611

Contents

AggloClust2D 2
clusterPCA 3
loadData 4
performTest 6
pighic 7
postHoc 8

Index **10**

`AggloClust2D` *Perform Constrained 2D Agglomerative Clustering*

Description

This function performs a connectivity constrained 2D agglomerative clustering using `scikit-learn` function `AgglomerativeClustering` and outputs an object of class `hclust` that stores the hierarchy of merges and value of criterion at each merge. It also outputs the optimal level of the hierarchy with respect to the elbow heuristic.

Usage

```
AggloClust2D(counts, nbClust = NULL)

## S3 method for class 'res2D'
print(x, ...)

## S3 method for class 'res2D'
summary(object, ...)

## S3 method for class 'res2D'
plot(x, ...)
```

Arguments

counts	an object of class InteractionSet obtained from the function loadData or an object of class resdiff obtained from function performTest.
nbClust	integer. Number of clusters to obtain. Set to NULL by default.
x	a res2D object to plot
...	not used
object	a res2D object to summarize

Value

An object of class res2D containing:

tree	an object of class hclust
nbClust	the number of clusters corresponding either to the value passed by the user or to the optimal level of clusters as provided by the elbow heuristic
clustering	obtained clustering

Author(s)

Élise Jorge <elise.jorge@inrae.fr>
Sylvain Foissac <sylvain.foissac@inrae.fr>
Toby Dylan Hocking <toby.hocking@r-project.org>
Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Examples

```
data("pighic")

res2D <- AggloClust2D(pighic$data)
if (!is.null(res2D)) { # in case Python or modules are not available
  clusters <- res2D$clustering
  print(res2D)
  summary(res2D)
  plot(res2D)
}
```

---

clusterPCA *Plot PCA of cluster characteristics*

Description

This function performs a principal component analysis (PCA) on selected cluster characteristics obtained from postHoc and returns a plot of the variable loadings for the first two principal components.

Usage

```
clusterPCA(
  resposthoc,
 selectedVar = c("TDP", "meanlogFC", "nbPixels", "nbBins", "minHeight", "maxHeight",
    "stripeScore", "triangleScore")
)
```

Arguments

resposthoc	An object of class resposthoc obtained from the function postHoc.
selectedVar	A character vector specifying the variables to include in the PCA. The default is c("TDP", "meanlogFC", "nbPixels", "nbBins","minHeight", "maxHeight", "stripeScore", "triangleScore").

Value

A ggplot object displaying the loadings of the first two principal components computed from the selected cluster characteristics.

Author(s)

Élise Jorge <elise.jorge@inrae.fr>
Sylvain Foissac <sylvain.foissac@inrae.fr>
Toby Dylan Hocking <toby.hocking@r-project.org>
Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Examples

```
data("pighic")
resdiff <- performTest(pighic$data, pighic$conditions)
res2D <- AggloClust2D(resdiff)
if (!is.null(res2D)) {
  resposthoc <- postHoc(resdiff, res2D$clustering, alpha = 0.05)
  clusterPCA(resposthoc)
}
```

---

loadData *Load and Normalize Hi-C Data*

Description

This function loads data necessary for the analysis and outputs them in a suitable format for performTest and 2Dclust.

Usage

```
loadData(files, index, chromosome, normalize = TRUE)
```

Arguments

files	character vector. Paths to Hi-C matrices in bed format.
index	character. A path to an index file in bed format.
chromosome	character or integer. Chromosome to select.
normalize	logical. Whether or not to normalize the output (with MA method). Set to TRUE by default.

Value

An InteractionSet corresponding to all interactions present in at least one of the input matrices and corresponding counts across all matrices.

Author(s)

Élise Jorge <elise.jorge@inrae.fr>
Sylvain Foissac <sylvain.foissac@inrae.fr>
Toby Dylan Hocking <toby.hocking@r-project.org>
Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Examples

```
replicates <- 1:2
cond <- "90"
allBegins <- interaction(expand.grid(replicates, cond), sep = "-")
allBegins <- as.character(allBegins)
chromosome <- 1
nbChr <- 1
allMat <- sapply(allBegins, function(ab) {
  matFile <- paste0("Rep", ab, "-chr", chromosome, "_200000.bed")
  })
index <- system.file("extdata", "index.200000.longest18chr.abs.bed",
                     package = "hicream")
                     format <- rep("HiC-Pro", length(replicates) * length(cond) * nbChr)
binsize <- 200000
files <- system.file("extdata", unlist(allMat), package = "hicream")
exData <- loadData(files, index, chromosome, normalize = TRUE)
```

performTest *Perform diffHic Test*

Description

This function performs diffHic differential analysis for every pixel of the matrix.

Usage

```
performTest(
  matrices,
  cond,
  outFile = NULL,
  filterLc = FALSE,
  filterFlank = FALSE,
  flank = NULL
)

## S3 method for class 'resdiff'
print(x, ...)

## S3 method for class 'resdiff'
summary(object, ...)

## S3 method for class 'resdiff'
plot(x, whichPlot = c("p.value", "p.adj", "logFC"), ...)
```

Arguments

matrices	an object of class InteractionSet obtained from the function loadData. Conditions correspond to the columns (one column per replicate).
cond	a vector indicating the condition of each column of matrices.
outFile	path to export outputs of the function, set to NULL by default, in which case results are not exported.
filterLc	logical. Whether to filter out low counts or not. Set to FALSE by default, in which case no filtering is performed. See filterTrended for more details.
filterFlank	logical. Whether to filter out on enriched pairs. Set to FALSE by default, in which case no filtering is performed. See enrichedPairs and filterPeaks for more details.
flank	flank parameter used only if filterFlank = TRUE. Set to NULL by default. See enrichedPairs for more details.
x	a resdiff object to plot
...	not used
object	a resdiff object to print
whichPlot	a character string indicating which plot to display. Possible values are "p.value", "p.adj" and "logFC". Set to "p.value" by default.

Value

An object of class `resdiff` with the following entries:

`region1`	the first bin of the interaction
`region2`	the second bin of the interaction
`p.value`	the p-value of the `diffHic` test
`p.adj`	the adjusted p-value of the `diffHic` test
`logFC`	the log2-fold-change of the interaction

Author(s)

Élise Jorge <elise.jorge@inrae.fr>
Sylvain Foissac <sylvain.foissac@inrae.fr>
Toby Dylan Hocking <toby.hocking@r-project.org>
Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Examples

```
data("pighic")
resdiff <- performTest(pighic$data, pighic$conditions)
resdiff
summary(resdiff)
plot(resdiff)
plot(resdiff, whichPlot = "p.adj")
plot(resdiff, whichPlot = "logFC")
```

---

pighic *Dataset "pighic"*

Description

6 Hi-C matrices (3 in each condition) obtained from two different developmental stages of pig embryos (90 and 110 days of gestation) corresponding to chromosome 1.

Format

An object with two entries:

data an object of class `InteractionSet` containing 6 Hi-C matrices from 2 conditions (3 each) for 21 interactions.

conditions a two-level factor indicating the condition of each matrix in `data`.

Details

The data are identical to the dataset present in the package `treediff`.

References

Nathalie Vialaneix, Gwendaelle Cardenas, Marie Chavent, Sylvain Foissac, Pierre Neuvial, and Nathanael Randriamihamison (2023). treediff: Testing Differences Between Families of Trees. https://cran.r-project.org/package=treediff

Examples

```
data(pighic)
```

---

postHoc *Perform Post Hoc Inference*

Description

This function performs post-hoc inference on all provided clusters using `performTest` results.

Usage

```
postHoc(resdiff, clusters, alpha, fill = FALSE, stepDown = TRUE)

## S3 method for class 'resposthoc'
print(x, ...)

## S3 method for class 'resposthoc'
summary(object, ...)

## S3 method for class 'resposthoc'
plot(x, ...)
```

Arguments

`resdiff`	An object of class `resdiff` obtained from the function `performTest`.
`clusters`	A vector corresponding to a clustering of resdiff rows.
`alpha`	A number between 0 and 1 at which the computed post hoc bounds will be valid.
`fill`	A boolean value. If TRUE (the default), enforce the total number of tests to be *p*(p+1)/2* by adding ones to the p-value vector for non tested entries.
`stepDown`	A boolean value. If FALSE, the single step Simes method is used. If TRUE (the default), the step-down Simes method is used. The latter is more powerful when the signal is strong.
`x`	a `resposthoc` object to plot
`...`	not used
`object`	a `resposthoc` object to summarize

Value

An object of class `resposthoc` containing a matrix with true positive proportions for each interaction and a dataframe with the following entries:

`region1`	The first bin of the interaction.
`region2`	The second bin of the interaction.
`clust`	The cluster the interaction belongs to.
`TDP`	The minimal post hoc true discovery (positive) proportion of the cluster the interaction belongs to.
`p.value`	The p-value of the diffHic test.
`p.adj`	The adjusted p-value of the diffHic test.
`logFC`	The log2-fold-change of the interaction.
`meanlogFC`	The mean of the log2-fold-change for the cluster the interaction belongs to.
`varlogFC`	The variance of the log2-fold-change for the cluster the interaction belongs to.
`propPoslogFC`	The proportion of interactions with positive log2-fold-change in the cluster the interaction belongs to.
`triangleScore`	Score indicating how similar the cluster shape is to a triangle, computed as npixels/((nbins**2)/2).
`stripeScore`	Score indicating how similar the cluster shape is to a stripe, computed as log(iproj/jproj) where iproj (resp. jproj) corresponds to the size of leftmost (resp. rightmost) genomic projection, which is the number of distinct region1 (resp. region2) bins involved.

Author(s)

Élise Jorge <elise.jorge@inrae.fr>
Sylvain Foissac <sylvain.foissac@inrae.fr>
Toby Dylan Hocking <toby.hocking@r-project.org>
Pierre Neuvial <pierre.neuvial@math.univ-toulouse.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Examples

```
data("pighic")
resdiff <- performTest(pighic$data, pighic$conditions)
res2D <- AggloClust2D(pighic$data)
if (!is.null(res2D)) { # in case Python or modules are not available
  clusters <- res2D$clustering
  alpha <- 0.05
  resposthoc <- postHoc(resdiff, clusters, alpha)
  resposthoc
  summary(resposthoc)
  plot(resposthoc)
}
```

Index

AggloClust2D, 2

clusterPCA, 3

diffHic, *7*

enrichedPairs, *6*

filterPeaks, *6*
filterTrended, *6*

InteractionSet, *3*, *5–7*

loadData, *3*, 4, *6*

performTest, *3*, 6, *8*
pighic, 7
plot.res2D *(*AggloClust2D*)*, 2
plot.resdiff *(*performTest*)*, 6
plot.resposthoc *(*postHoc*)*, 8
postHoc, *3*, *4*, 8
print.res2D *(*AggloClust2D*)*, 2
print.resdiff *(*performTest*)*, 6
print.resposthoc *(*postHoc*)*, 8

summary.res2D *(*AggloClust2D*)*, 2
summary.resdiff *(*performTest*)*, 6
summary.resposthoc *(*postHoc*)*, 8