

# Package ‘spatzie’

April 12, 2022

**Type** Package

**Title** Identification of enriched motif pairs from chromatin interaction data

**Version** 1.0.1

**Description** Identifies motifs that are significantly co-enriched from enhancer-promoter interaction data. While enhancer-promoter annotation is commonly used to define groups of interaction anchors, spatzie also supports co-enrichment analysis between preprocessed interaction anchors. Supports BEDPE interaction data derived from genome-wide assays such as HiC, ChIA-PET, and HiChIP. Can also be used to look for differentially enriched motif pairs between two interaction experiments.

**License** GPL-3

**URL** <https://spatzie.mit.edu>

**Depends** R (>= 4.1)

**Imports** BiocGenerics, BSgenome, GenomeInfoDb, GenomicFeatures, GenomicInteractions, GenomicRanges, ggplot2, IRanges, matrixStats, motifmatchr, S4Vectors, stats, SummarizedExperiment, TFBSTools, utils

**Suggests** BiocManager, Biostrings, knitr, pheatmap, rmarkdown, testthat, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Mmusculus.UCSC.mm10.knownGene, TxDb.Mmusculus.UCSC.mm9.knownGene

**VignetteBuilder** knitr

**biocViews** DNA3DStructure, GeneRegulation, PeakDetection, Epigenetics, FunctionalGenomics, Classification, HiC, Transcription

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**git\_url** <https://git.bioconductor.org/packages/spatzie>

**git\_branch** RELEASE\_3\_14

**git\_last\_commit** cb4b61e

**git\_last\_commit\_date** 2022-02-02

**Date/Publication** 2022-04-12

**Author** Jennifer Hammelman [aut, cre, cph]

(<https://orcid.org/0000-0002-1008-2666>),

Konstantin Krismer [aut] (<https://orcid.org/0000-0001-8994-3416>),

David Gifford [ths, cph] (<https://orcid.org/0000-0003-1709-4034>)

**Maintainer** Jennifer Hammelman <jhammelm@mit.edu>

## R topics documented:

anchor_pair_enrich . . . . .	2
anchor_pair_example_count . . . . .	5
anchor_pair_example_match . . . . .	5
anchor_pair_example_score . . . . .	6
compare_motif_pairs . . . . .	6
compare_pairs_example . . . . .	7
filter_motifs . . . . .	7
filter_pairs_example . . . . .	8
filter_pair_motifs . . . . .	9
find_ep_coenrichment . . . . .	10
get_specific_interactions . . . . .	12
interactions_yy1 . . . . .	13
interactions_yy1_enhancer . . . . .	14
interactions_yy1_ep . . . . .	14
interactions_yy1_promoter . . . . .	15
int_data_k562 . . . . .	15
int_data_mslcl . . . . .	16
int_data_yy1 . . . . .	16
plot_motif_occurrence . . . . .	17
scan_interactions_example . . . . .	18
scan_interactions_example_filtered . . . . .	18
scan_motifs . . . . .	19
spatzie . . . . .	20

<b>Index</b>	<b>21</b>
--------------	-----------

---

anchor_pair_enrich	<i>Determine enriched motifs in anchors</i>
--------------------	---

---

### Description

Determine whether motifs between paired bed regions have a statistically significant relationship. Options for significance are motif score correlation, motif count correlation, or hypergeometric motif co-occurrence.

**Usage**

```
anchor_pair_enrich(interaction_data, method = c("count", "score", "match"))
```

**Arguments**

`interaction_data` an interactionData object of paired genomic regions

`method` method for co-occurrence, valid options include:

`count`: correlation between counts (for each anchor, tally positions where motif score  $> 5 * 10^{-5}$ )

`score`: correlation between motif scores (for each anchor, use the maximum score over all positions)

`match`: association between motif matches (for each anchor, a match is defined if there is at least one position with a motif score

**Value**

an interactionData object where `obj$pair_motif_enrich` contains the p-values for significance of seeing a higher co-occurrence than what we get by chance.

**Score-based correlation**

We assume motif scores follow a normal distribution and are independent between enhancers and promoters. We can therefore compute how correlated scores of any two transcription factor motifs are between enhancer and promoter regions using Pearson's product-moment correlation coefficient:

$$r = \frac{\sum(x'_i - \bar{x}')(y'_i - \bar{y}')}{\sqrt{\sum(x'_i - \bar{x}')^2 \sum(y'_i - \bar{y}')^2}}$$

, where the input vectors  $\mathbf{x}$  and  $\mathbf{y}$  from above are transformed to vectors  $\mathbf{x}'$  and  $\mathbf{y}'$  by replacing the set of scores with the maximum score for each region:

$$x'_i = \max x_i$$

$x'_i$  is then the maximum motif score of motif  $a$  in the promoter region of interaction  $i$ ,  $y'_i$  is the maximum motif score of motif  $b$  in the enhancer region of interaction  $i$ , and  $\bar{x}'$  and  $\bar{y}'$  are the sample means.

Significance is then computed by transforming the correlation coefficient  $r$  to test statistic  $t$ , which is Student  $t$ -distributed with  $n - 2$  degrees of freedom.

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

All p-values are calculated as one-tailed p-values of the probability that scores are greater than or equal to  $r$ .

**Count-based correlation**

Instead of calculating the correlation of motif scores directly, the count-based correlation metric first tallies the number of instances of a given motif within an enhancer or a promoter region, which are defined as all positions in those regions with motif score p-values of less than  $5 * 10^{-5}$ . Formally,

the input vectors  $\mathbf{x}$  and  $\mathbf{y}$  are transformed to vectors  $\mathbf{x}''$  and  $\mathbf{y}''$  by replacing the set of scores with the cardinality of the set:

$$x_i'' = |x_i|$$

And analogous for  $y_i''$ . Finally, the correlation coefficient  $r$  between  $\mathbf{x}''$  and  $\mathbf{y}''$  and its associated significance are calculated as described above.

### Match-based association

Instance co-occurrence uses the presence or absence of a motif within an enhancer or promoter to determine a statistically significant association, thus  $\mathbf{x}'''$  and  $\mathbf{y}'''$  are defined by:

$$x_i''' = \mathbf{1}_{x_i'' > 0}$$

Instance co-occurrence is computed using the hypergeometric test:

$$p = \sum_{k=I_{ab}}^{P_a} \frac{\text{binom}(P_a, k) \text{binom}(n - P_a, E_b - k)}{\text{binom}(n, E_b)},$$

where  $I_{ab}$  is the number of interactions that contain a match for motif  $a$  in the promoter and motif  $b$  in the enhancer,  $P_a$  is the number of promoters that contain motif  $a$  ( $P_a = \sum_i^n x_i'''$ ),  $E_b$  is the number of enhancers that contain motif  $b$  ( $E_b = \sum_i^n y_i'''$ ), and  $n$  is the total number of interactions, which is equal to the number of promoters and to the number of enhancers.

### Author(s)

Jennifer Hammelman  
Konstantin Krismer

### Examples

```
## Not run:
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"
if (!(genome_id %in% rownames(utils::installed.packages()))) {
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)
}
genome <- BSgenome::getBSgenome(genome_id)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
                           package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")

yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)
yy1_pd_interaction <- filter_motifs(yy1_pd_interaction, 0.4)
yy1_pd_count_corr <- anchor_pair_enrich(yy1_pd_interaction, method = "count")

## End(Not run)

res <- anchor_pair_enrich(spatzie::scan_interactions_example_filtered,
                          method = "score")
```

---

anchor\_pair\_example\_count  
*spatzie count correlation data set*

---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs, filtered for motifs present in at least 10 interactions with count correlation. It serves as unit test data.

**Usage**

```
data(anchor_pair_example_count)
```

**Format**

An interactionData object

---

anchor\_pair\_example\_match  
*spatzie match association data set*

---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs, filtered for motifs present in at least 10 interactions with using the hypergeometric test. It serves as unit test data.

**Usage**

```
data(anchor_pair_example_match)
```

**Format**

A interactionData object

---

anchor\_pair\_example\_score  
*spatzie score correlation data set*

---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs, filtered for motifs present in at least 10 interactions with score correlation. It serves as unit test data.

**Usage**

```
data(anchor_pair_example_score)
```

**Format**

An interactionData object

---

compare\_motif\_pairs    *Compare pairs of motifs between two interaction datasets*

---

**Description**

Compute the log-likelihood ratio that a motif pair is differential between two interaction datasets. Note that motif pair significance should have been computed using the same method for both datasets.

**Usage**

```
compare_motif_pairs(
  interaction_data1,
  interaction_data2,
  differential_p = 0.05
)
```

**Arguments**

interaction\_data1  
an interactionData object of paired genomic regions that has been scanned for significant motif:motif interactions

interaction\_data2  
an interactionData object of paired genomic regions that has been scanned for significant motif:motif interactions

differential\_p threshold for significance of differential p-value

**Value**

a matrix of the log likelihood ratio of motif pairs that are significantly differential between two interactionData sets

**Author(s)**

Jennifer Hammelman

**Examples**

```
pheatmap::pheatmap(compare_motif_pairs(spatzie::int_data_k562,
                                       spatzie::int_data_mslcl, 5e-06),
                    fontsize = 6)
```

---

compare\_pairs\_example *compare\_motif\_pairs example*

---

**Description**

This is a matrix containing example result from compare\_motif\_pairs. It serves as unit test data.

**Usage**

```
data(compare_pairs_example)
```

**Format**

A matrix

---

filter\_motifs *Filter motifs based on occurrence within interaction data*

---

**Description**

Select a subset of motifs that are in at least a threshold fraction of regions. Motif subsets are selected separately for anchor one and anchor two regions.

**Usage**

```
filter_motifs(interaction_data, threshold)
```

**Arguments**

interaction\_data  
 an interactionData object of paired genomic regions

threshold  
 fraction of interactions that should contain a motif for a motif to be considered

**Value**

an interactionData object where obj\$anchor1\_motif\_indices and obj\$anchor2\_motif\_indices have been filtered to motifs that are present in a threshold fraction of interactions

**Author(s)**

Jennifer Hammelman

**Examples**

```
## Not run:
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"
if (!(genome_id %in% rownames(utils::installed.packages()))) {
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)
}
genome <- BSgenome::getBSgenome(genome_id)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
                           package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")

yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)
yy1_pd_interaction <- filter_motifs(yy1_pd_interaction, 0.4)

## End(Not run)

res <- filter_motifs(spatzie::scan_interactions_example, threshold = 0.1)
```

---

filter\_pairs\_example *spatzie score correlation filtered data set*

---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs, filtered for motifs present in at least 10 interactions with score correlation, and filtered for pairs with  $p < 0.5$ . It serves as unit test data.

**Usage**

```
data(filter_pairs_example)
```

**Format**

An interactionData object



---

filter_pair_motifs	<i>Filter significant motif interactions</i>
--------------------	--

---

## Description

Multiple hypothesis correction applied to filter for significant motif interactions.

## Usage

```
filter_pair_motifs(interaction_data, method = "fdr", threshold = 0.05)
```

## Arguments

interaction_data	an interactionData object of paired genomic regions
method	statistical method for multiple hypothesis correction, defaults to Benjamini-Hochberg ("fdr") (see <a href="#">p.adjust</a> for options)
threshold	p-value threshold for significance cut-off

## Value

an interactionData object where `obj$pair_motif_enrich` contains multiple hypothesis corrected p-values for significance of seeing a higher co-occurrence than what we get by chance and `obj$pair_motif_enrich_sig` contains only motifs that have at least one significant interaction.

## Author(s)

Jennifer Hammelman

## Examples

```
## Not run:
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"
if (!(genome_id %in% rownames(utils::installed.packages()))) {
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)
}
genome <- BSgenome::getBSgenome(genome_id)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
  package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")

yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)
yy1_pd_interaction <- filter_motifs(yy1_pd_interaction, 0.4)
yy1_pd_score_corr <- anchor_pair_enrich(yy1_pd_interaction, method = "score")
yy1_pd_score_corr_adj <- filter_pair_motifs(yy1_pd_score_corr)

## End(Not run)
```

```
res <- filter_pair_motifs(spatzie::anchor_pair_example_count,
                        threshold = 0.5)
```

---

find\_ep\_coenrichment *Find co-enriched motif pairs in enhancer-promoter interactions*

---

## Description

Identifies co-enriched pairs of motifs in enhancer-promoter interactions selected from a data frame of general genomic interactions.

If `identify_ep`: Promoters and enhancers are identified using genomic annotations, where anchors close to promoter annotations (within 2500 base pairs) are considered promoters and all other anchors are considered gene-distal enhancers. Only interactions in `int_raw_data` between promoters and enhancers are used for motif co-enrichment analysis.

If `!identify_ep`: Instead of automatically identifying promoters and enhancers based on genomic annotations, all interactions in `int_raw_data` must be preprocessed in a way that anchor 1 contains promoters and anchor 2 contains enhancers. Motif co-enrichment analysis is performed under this assumption.

Calls functions `scan_motifs`, `filter_motifs`, and `anchor_pair_enrich` internally.

## Usage

```
find_ep_coenrichment(
  int_raw_data,
  motifs_file,
  motifs_file_matrix_format = c("pfm", "ppm", "pwm"),
  genome_id = c("hg38", "hg19", "mm9", "mm10"),
  identify_ep = TRUE,
  cooccurrence_method = c("count", "score", "match"),
  filter_threshold = 0.4
)
```

## Arguments

`int_raw_data` a `GenomicInteractions` object or a data frame with at least six columns:

- column 1: character; genomic location of interaction anchor 1 - chromosome (e.g., "chr3")
- column 2: integer; genomic location of interaction anchor 1 - start coordinate
- column 3: integer; genomic location of interaction anchor 1 - end coordinate
- column 4: character; genomic location of interaction anchor 2 - chromosome (e.g., "chr3")
- column 5: integer; genomic location of interaction anchor 2 - start coordinate
- column 6: integer; genomic location of interaction anchor 2 - end coordinate

**motifs\_file**     **JASPAR format** matrix file containing multiple motifs to scan for, gz-zipped files allowed  
**motifs\_file\_matrix\_format**  
                   type of position-specific scoring matrices in `motifs_file`, valid options include:

**pfm:**   position frequency matrix, elements are absolute frequencies, i.e., counts (default)  
**ppm:**   position probability matrix, elements are probabilities, i.e., Laplace smoothing corrected relative frequencies  
**pwm:**   position weight matrix, elements are log likelihoods

**genome\_id**       ID of genome assembly interactions in `int_raw_data` were aligned to, valid options include hg19, hg38, mm9, and mm10, defaults to hg38  
**identify\_ep**     logical, set FALSE if enhancers and promoters should not be identified based on genomic annotations, but instead assumes anchor 1 contains promoters and anchor 2 contains enhancers, for all interactions in `int_raw_data`, defaults to TRUE, i.e., do identify enhancers and promoters of interactions in `int_raw_data` based on genomic interactions and filter all interactions which are not between promoters and enhancers  
**cooccurrence\_method**  
                   method for co-occurrence, valid options include:

**count:**   correlation between counts (for each anchor, tally positions where motif score  $> 5 * 10^{-5}$ )  
**score:**   correlation between motif scores (for each anchor, use the maximum score over all positions)  
**match:**   association between motif matches (for each anchor, a match is defined if there is at least one position with a motif score)

                  See [anchor\\_pair\\_enrich](#) for details.

**filter\_threshold**  
                   fraction of interactions that should contain a motif for a motif to be considered, see [filter\\_motifs](#), defaults to 0.4

## Value

a list with the following items:

**int\_data**        [GenomicInteractions](#) object; promoter-enhancer interactions  
**int\_data\_motifs:**   interactionData object; return value of [scan\\_motifs](#)  
**filtered\_int\_data\_motifs:** interactionData object; return value of [filter\\_motifs](#)  
**annotation\_pie\_chart:** ggplot2 plot; return value of [plotInteractionAnnotations](#)  
**motif\_cooccurrence:** interactionData object; return value of [anchor\\_pair\\_enrich](#)

## Author(s)

Jennifer Hammelman  
 Konstantin Krismer

## Examples

```
## Not run:
```

```
interactions_file <- system.file("extdata/yy1_interactions.bedpe.gz",
                                package = "spatzie")
motifs_file <- system.file("extdata/motifs_subset.txt.gz",
                           package = "spatzie")

df <- read.table(gzfile(interactions_file), header = TRUE, sep = "\t")
res <- find_ep_coenrichment(df, motifs_file,
                           motifs_file_matrix_format = "pfm",
                           genome_id = "mm10")

## End(Not run)
```

---

get\_specific\_interactions

*Get interactions that contain a specific motif pair*

---

### Description

Select interactions that contain anchor1\_motif within anchor 1 and anchor2\_motif within anchor 2.

### Usage

```
get_specific_interactions(
  interaction_data,
  anchor1_motif = NULL,
  anchor2_motif = NULL
)
```

### Arguments

interaction\_data  
an interactionData object of paired genomic regions

anchor1\_motif Motif name from interactionData\$anchor1\_motifs

anchor2\_motif Motif name from interactionData\$anchor2\_motifs

### Value

a [GenomicInteractions](#) object containing a subset subset of interactions that contain an instance of anchor1\_motif in anchor 1 and anchor2\_motif in anchor 2

### Author(s)

Jennifer Hammelman

**Examples**

```
## Not run:
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"
if (!(genome_id %in% rownames(utils::installed.packages()))) {
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)
}
genome <- BSgenome::getBSgenome(genome_id)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
                           package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")

yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)
yy1_pd_interaction <- filter_motifs(yy1_pd_interaction, 0.4)
yy1_pd_count_corr <- anchor_pair_enrich(yy1_pd_interaction,
                                       method = "score")
yy1_yy1_interactions <- get_specific_interactions(
  yy1_pd_interaction,
  anchor1_motif = "YY1",
  anchor2_motif = "YY1")

## End(Not run)

res <- get_specific_interactions(spatzie::int_data_yy1,
                               anchor1_motif = "YY1",
                               anchor2_motif = "YY1")
```

---

interactions\_yy1

*Mouse YY1 Enhancer - Promoter Interactions Data Set*


---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET and serves as example and unit test data. The same data set is used in the vignette.

**Usage**

```
data(interactions_yy1)
```

**Format**

A [GenomicInteractions](#) object

---

interactions\_yy1\_enhancer

*Mouse YY1 Enhancer - Promoter Interactions Data Set - YY1 enhancers*

---

### Description

This is a [GenomicInteractions](#) object containing processed results from YY1 ChIA-PET of interactions that contain a YY1 motif in the enhancer (anchor 2) region. It serves as unit test data.

### Usage

```
data(interactions_yy1_enhancer)
```

### Format

A [GenomicInteractions](#) object

---

interactions\_yy1\_ep

*Mouse YY1 Enhancer - Promoter Interactions Data Set - YY1 enhancers/promoters*

---

### Description

This is a [GenomicInteractions](#) object containing processed results from YY1 ChIA-PET of interactions that contain a YY1 motif in the promoter (anchor 1) region and a YY1 motif in the enhancer (anchor 2) region. It serves as unit test data.

### Usage

```
data(interactions_yy1_ep)
```

### Format

A [GenomicInteractions](#) object

---

interactions\_yy1\_promoter

*Mouse YY1 Enhancer - Promoter Interactions Data Set - YY1 promoters*

---

### Description

This is a [GenomicInteractions](#) object containing processed results from YY1 ChIA-PET of interactions that contain a YY1 motif in the promoter (anchor 1) region. It serves as unit test data.

### Usage

```
data(interactions_yy1_promoter)
```

### Format

A [GenomicInteractions](#) object

---

int\_data\_k562

*K562 Enhancer - Promoter Interactions Data Set*

---

### Description

This object contains genomic interactions obtained by human RAD21 ChIA-PET from K562 cells and serves as unit test data.

### Usage

```
data(int_data_k562)
```

### Format

An `interactionData` object

---

int_data_mslcl	<i>MSLCL Enhancer - Promoter Interactions Data Set</i>
----------------	--

---

**Description**

This object contains genomic interactions obtained by human RAD21 ChIA-PET from MSLCL cells and serves as unit test data.

**Usage**

```
data(int_data_mslcl)
```

**Format**

An interactionData object

---

int_data_yy1	<i>Mouse YY1 Enhancer - Promoter Interactions Data Set</i>
--------------	--

---

**Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET and serves as example and unit test data.

**Usage**

```
data(int_data_yy1)
```

**Format**

An interactionData object



---

plot\_motif\_occurrence *Plot motif occurrence*

---

### Description

Plots a histogram of motif values (either counts, instances, or scores) for anchor 1 and anchor 2 regions.

### Usage

```
plot_motif_occurrence(  
  interaction_data,  
  method = c("counts", "instances", "scores")  
)
```

### Arguments

interaction_data	an interactionData object of paired genomic regions
method	way to interpret motif matching for each anchor region as "counts" number of motifs per region, "instances" motif present or absent each region, or "scores" maximum motif PWM match score for each region

### Value

plot containing histogram for each anchor

### Author(s)

Jennifer Hammelman

### Examples

```
## Not run:  
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"  
if (!(genome_id %in% rownames(utils::installed.packages()))) {  
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)  
}  
genome <- BSgenome::getBSgenome(genome_id)  
  
motifs_file <- system.file("extdata/motifs_subset.txt.gz",  
  package = "spatzie")  
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")  
  
yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)  
yy1_pd_interaction <- filter_motifs(yy1_pd_interaction, 0.4)  
plot_motif_occurrence(yy1_pd_interaction, "counts")  
  
## End(Not run)
```

```
plot_motif_occurrence(spatzie::anchor_pair_example_score)
```

---

```
scan_interactions_example
```

*Interactions scanned for motifs - interactionData object*

---

### **Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs and serves as unit test data.

### **Usage**

```
data(scan_interactions_example)
```

### **Format**

An interactionData object

---

```
scan_interactions_example_filtered
```

*Interactions with motifs filtered for significance - interactionData object*

---

### **Description**

This object contains genomic interactions obtained by mouse YY1 ChIA-PET scanned for mouse transcription factor motifs and filtered for motifs present in at least 10

### **Usage**

```
data(scan_interactions_example_filtered)
```

### **Format**

An interactionData object

---

scan_motifs	<i>Scans interaction file for motif instances</i>
-------------	---

---

## Description

Uses motifmatchR to scan interaction regions for given motifs.

## Usage

```
scan_motifs(int_data, motifs, genome)
```

## Arguments

int_data	a <a href="#">GenomicInteractions</a> object of paired genomic regions
motifs	a TFBS tools matrix of DNA binding motifs
genome	BSgenome object or DNASTringSet object, must match chromosomes from interaction data file

## Value

an interaction data object where obj\$anchor1\_motifs and obj\$anchor2\_motifs contain information about the scores and matches to motifs from anchor one and anchor two of interaction data genomic regions

## Author(s)

Jennifer Hammelman

## Examples

```
## Not run:
genome_id <- "BSgenome.Mmusculus.UCSC.mm9"
if (!(genome_id %in% rownames(utils::installed.packages()))) {
  BiocManager::install(genome_id, update = FALSE, ask = FALSE)
}
genome <- BSgenome::getBSgenome(genome_id)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
  package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")

yy1_pd_interaction <- scan_motifs(spatzie::interactions_yy1, motifs, genome)

## End(Not run)

motifs_file <- system.file("extdata/motifs_subset.txt.gz",
  package = "spatzie")
motifs <- TFBSTools::readJASPARMatrix(motifs_file, matrixClass = "PFM")
```

```
left <- GenomicRanges::GRanges(  
  seqnames = c("chr1", "chr1", "chr1"),  
  ranges = IRanges::IRanges(start = c(1, 15, 20),  
                             end = c(10, 35, 31))  
right <- GenomicRanges::GRanges(  
  seqnames = c("chr1", "chr2", "chr2"),  
  ranges = IRanges::IRanges(start = c(17, 47, 41),  
                             end = c(28, 54, 53))  
test_interactions <- GenomicInteractions::GenomicInteractions(left, right)  
  
# toy DNASTringSet to replace BSgenome object  
seqs <- c("chr1" = "CCACTAGCCACGCGTCACTGGTTAGCGTGATTGAACTAAATCGTATGAAAATCC",  
          "chr2" = "CTACAAACTAGGAATTTAGGCAAACCTGTGTTAAAATCTTAGCTCATTCAAT")  
toy_genome <- Biostrings::DNASTringSet(seqs, use.names = TRUE)  
  
res <- scan_motifs(test_interactions, motifs, toy_genome)
```

---

spatzie

*spatzie*

---

### Description

Looks for motifs which are significantly co-enriched from enhancer-promoter interaction data, derived from assays such as as HiC, ChIA-PET, etc. It can also look for differentially enriched motif pairs between to interaction experiments.

### Author(s)

Jennifer Hammelman

Konstantin Krismer

# Index

## \* datasets

- anchor\_pair\_example\_count, 5
  - anchor\_pair\_example\_match, 5
  - anchor\_pair\_example\_score, 6
  - compare\_pairs\_example, 7
  - filter\_pairs\_example, 8
  - int\_data\_k562, 15
  - int\_data\_mslcl, 16
  - int\_data\_yy1, 16
  - interactions\_yy1, 13
  - interactions\_yy1\_enhancer, 14
  - interactions\_yy1\_ep, 14
  - interactions\_yy1\_promoter, 15
  - scan\_interactions\_example, 18
  - scan\_interactions\_example\_filtered, 18
- anchor\_pair\_enrich, 2, 10, 11
- anchor\_pair\_example\_count, 5
- anchor\_pair\_example\_match, 5
- anchor\_pair\_example\_score, 6
- compare\_motif\_pairs, 6
- compare\_pairs\_example, 7
- filter\_motifs, 7, 10, 11
- filter\_pair\_motifs, 9
- filter\_pairs\_example, 8
- find\_ep\_coenrichment, 10
- GenomicInteractions, 10–15, 19
- get\_specific\_interactions, 12
- int\_data\_k562, 15
- int\_data\_mslcl, 16
- int\_data\_yy1, 16
- interactions\_yy1, 13
- interactions\_yy1\_enhancer, 14
- interactions\_yy1\_ep, 14
- interactions\_yy1\_promoter, 15
- p.adjust, 9
- plot\_motif\_occurrence, 17
- plotInteractionAnnotations, 11
- scan\_interactions\_example, 18
- scan\_interactions\_example\_filtered, 18
- scan\_motifs, 10, 11, 19
- spatie, 20