Package ‘cobindR’

April 25, 2017

Title Finding Co-occurring motifs of transcription factor binding sites
Description Finding and analysing co-occurring motifs of transcription factor binding sites in groups of genes
Version 1.14.0
Date 2013-07-09
Author Manuela Benary, Stefan Kroeger, Yuehien Lee, Robert Lehmann
Maintainer Manuela Benary <manuela.benary@cms.hu-berlin.de>
License Artistic-2.0
Imports methods, seqinr, yaml, rtfbs, gplots, mclust, gmp,
   BiocGenerics (>= 0.13.8), IRanges, Biostrings, BSgenome,
   biomaRt
Suggests RUnit
Enhances rGADEM, seqLogo, genoPlotR, parallel, VennDiagram,
   RColorBrewer, vcd, MotifDb, snowfall
biocViews ChIPSeq, CellBiology, MultipleComparison, SequenceMatching
NeedsCompilation no

R topics documented:

  cobindR-package ...................................................  3
  bg_binding_sites ..................................................  3
  bg_pairs .............................................................  4
  bg_sequences .......................................................  5
  bg_sequence_origin ................................................  6
  bg_sequence_source ................................................  6
  bg_sequence_type ..................................................  7
  binding_sites .......................................................  8
  cobindr-class ......................................................  9
  cobindRConfiguration ............................................. 10
  comment ............................................................... 11
  configuration ......................................................... 12
  configuration-class ................................................ 13
  downstream .......................................................... 14
  experiment_description ............................................ 15
  fdrThreshold ......................................................... 16
  find.pairs ........................................................... 16
R topics documented:

get.bindingsite.ranges .............................................. 17
get.pairs ................................................................ 18
get.significant.pairs ................................................. 19
id ........................................................................... 19
location ................................................................. 20
mart ......................................................................... 21
max_distance .......................................................... 22
name ....................................................................... 22
pairs ......................................................................... 23
pairs_of_interest ...................................................... 24
path ......................................................................... 25
pfm .......................................................................... 26
pfm_path .................................................................... 27
plot.detrending ......................................................... 27
plot.gc ....................................................................... 28
plot.pairdistance ...................................................... 29
plot.pairdistribution ................................................. 30
plot.positionprofile .................................................. 31
plot.positions .......................................................... 31
plot.positions.simple .............................................. 32
plot.tfbs.heatmap ..................................................... 33
plot.tfbs.venndiagram .............................................. 34
plot.tfbslogo ............................................................ 35
predicted2pwm .......................................................... 35
pseudocount ............................................................. 36
pValue ...................................................................... 37
rtfbs ........................................................................ 37
search.gadem ............................................................. 39
search.pwm ............................................................... 41
seqObj ...................................................................... 42
SeqObj-class ............................................................. 43
sequence ................................................................. 44
sequences .................................................................. 45
sequence_origin ....................................................... 46
sequence_source ...................................................... 46
sequence_type .......................................................... 47
species ..................................................................... 48
testCpG ..................................................................... 49
threshold ................................................................. 50
uid .......................................................................... 51
upstream ................................................................. 52
write.bindingsites ...................................................... 52
write.bindingsites.table ............................................. 53
write.pairs ............................................................... 54
write.sequences ......................................................... 54

Index ........................................................................ 56
An R package for analyzing co-occurring transcription factor binding sites

Description

Many transcription factors (TFs) regulate gene expression by binding to specific DNA motifs near genes. Often the regulation of gene expression is not only controlled by one TF, but by many TFs together, that can either interact in a cooperative manner or interfere with each other. In recent years high throughput methods, like ChIP-Seq, have become available to produce large amounts of data, that contain potential regulatory regions. In silico analysis of transcription factor binding sites can help to interpret these enormous datasets in a convenient and fast way or narrow down the results to the most significant regions for further experimental studies.

cobindR provides a complete set of methods to analyse and detect pairs of TFs, including support of diverse input formats and different background models for statistical testing. Several visualization tools are implemented to ease the interpretation of the results.

Author(s)

Yue-Hien Lee, Robert Lehmann, Stefan Kroeger, Manuela Benary

See Also

The core class in this package: cobindr-class. The core function in this package: find.pairs.

bg_binding_sites

motif hits in the background sequences

Description

motif hits in the background sequences

Usage

## S4 method for signature 'cobindr'
bg Binding_sites(x)

## S4 replacement method for signature 'cobindr,data.frame'
bg Binding_sites(x) <- value

Arguments

x a cobindr object

value data.frame holding the binding site hits in the background sequences

Value

motif hits in background sequences (data.frame)
Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid, name, sequences, bg_sequences, desc, configuration, binding_sites, pfm, bg_binding_sites, pairs, bg_pairs, ...

Examples

cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette_seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak bg_binding_sites'
cbr <- cobindr(cfg)
bg_binding_sites(cbr)

---

bg_pairs

motif hit pairs in the background sequences

Description

motif hit pairs in the background sequences

Usage

## S4 method for signature 'cobindr'
bg_pairs(x)
## S4 replacement method for signature 'cobindr, data.frame'
bg_pairs(x) <- value

Arguments

x a cobindr object
value data.frame holding the binding site pairs in the background sequences

Value

background motif pairs (data.frame)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid, name, sequences, bg_sequences, desc, configuration, binding_sites, bg_binding_sites, pfm, pairs, bg_pairs, ...
Examples

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette.seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak bg_pairs'
cbr <- cobindr(cfg)
bg_pairs(cbr)
```

---

**Description**

list of background sequence

**Usage**

```r
## S4 method for signature 'cobindr'
bg_sequences(x)
## S4 replacement method for signature 'cobindr,list'
bg_sequences(x) <- value
```

**Arguments**

- `x`: a cobindr object
- `value`: list of background sequence of type SeqObj

**Value**

list of background sequences (SeqObj)

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

`uid`, `name`, `bg_sequences`, `desc`, `configuration`, `binding_sites`, `bg_binding_sites`, `pfm`, `pairs`, `bg_pairs`, `pairs_of_interest`

**Examples**

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette.seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak bg_sequences'
cbr <- cobindr(cfg)
length(bg_sequences(cbr))
```
Description

background sequence origin note

Usage

```r
## S4 method for signature 'configuration'
bg_sequence_origin(x)
## S4 replacement method for signature 'configuration,character'
bg_sequence_origin(x) <- value
```

Arguments

- **x**: a cobindR configuration object
- **value**: a character

Value

background sequence origin (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

- `id`, `experiment_description`, `sequence_source`, `sequence_origin`, `sequence_type`, `bg_sequence_source`, `bg_sequence_type`, `species`, `downstream`, `upstream`, `max_distance`, `pairs`, `pfm_path`, `threshold`, `fdrThreshold`, `path`, `mart`, `pValue`

Examples

```r
cfg <- cobindRConfiguration()
bg_sequence_origin(cfg)
```

Description

background sequence source note

Usage

```r
## S4 method for signature 'configuration'
bg_sequence_source(x)
## S4 replacement method for signature 'configuration,character'
bg_sequence_source(x) <- value
```
Arguments

x  a cobindR configuration object
value  a character

Value

background sequence source (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, bg_sequence_type, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue

Examples

cfg <- cobindRConfiguration()
bg_sequence_source(cfg)

---

bg_sequence_type  background sequence type note

Description

background sequence type note

Usage

## S4 method for signature 'configuration'
bg_sequence_type(x)
## S4 replacement method for signature 'configuration,character'
bg_sequence_type(x) <- value

Arguments

x  a cobindR configuration object
value  a character

Value

bg_sequence_type (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, bg_sequence_type, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue
Examples

cfg <- cobindRConfiguration()
binding_sites(cbr)

binding_sites  motif hits on the foreground sequences

Description

motif hits on the foreground sequences

Usage

## S4 method for signature 'cobindr'
binding_sites(x)
## S4 replacement method for signature 'cobindr,data.frame'
binding_sites(x) <- value

Arguments

x a cobindr object
value data.frame holding the binding site hits in the foreground sequences

Value

motif hits in foreground sequences as data.frame

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid,name,sequences,bg_sequences,desc,configuration,binding_sites,bg_binding_sites,pfm,pairs,bg_pairs,pairs_of_interest

Examples

cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette_seqs.fasta',package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak binding_sites'
cbr <- cobindr(cfg)
binding_sites(cbr)
**cobindr-class**

**Class"cobindr"**

**Description**

Container for experiment run and its meta-data

**Objects from the Class**

Objects can be created by calls of the form `new("cobindr", conf, name, desc)`.

**Slots**

- `uid`: Object of class "character" ~~ unique id for internal representation
- `name`: Object of class "character" ~~ name of the experiment
- `sequences`: Object of class "list" ~~ list of sequence objects to be analyzed
- `bg_sequences`: Object of class "list" ~~ list of background sequences for statistical analyses
- `desc`: Object of class "character" ~~ verbal experiment description
- `configuration`: Object of class "configuration" ~~ the configuration object used to describe the experiment
- `pfm`: Object of class "list" ~~ list of pfms to be used
- `pairs_of_interest`: Object of class "factor" ~~ contains pairs for search

**Methods**

- `detrending` signature(object = "cobindr"): ...
- `find.pairs` signature(object = "cobindr"): ...
- `generate.background` signature(object = "cobindr"): ...
- `get.bindingsite.ranges` signature(object = "cobindr"): ...
- `get.pairs` signature(object = "cobindr"): ...
- `get.significant.pairs` signature(object = "cobindr"): ...
- `initialize` signature(.Object = "cobindr"): ...
- `input.pwm` signature(object = "cobindr"): ...
plot.detrending signature(object = "cobindr"):
plot.gc signature(object = "cobindr"):
plot.pairdistance signature(object = "cobindr"):
plot.pairdistribution signature(object = "cobindr"):
plot.positionprofile signature(object = "cobindr"):
plot.positions.simple signature(object = "cobindr"):
plot.positions signature(object = "cobindr"):
plot.tfbs.heatmap signature(object = "cobindr"):
plot.tfbs.venndiagram signature(object = "cobindr"):
plot.tfbslogo signature(object = "cobindr"):
predicted2pwm signature(object = "cobindr"):
rtfbs signature(object = "cobindr"):
search.gadem signature(object = "cobindr"):
search.pwm signature(object = "cobindr"):
testCpG signature(object = "cobindr"):
write.bindingsites.table signature(object = "cobindr"):
write.bindingsites signature(object = "cobindr"):
write.sequences signature(object = "cobindr"):
write signature(x = "cobindr", file = "character"):

Author(s)
Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also
SeqObj configuration

Examples
showClass("cobindr")

---

cobindRConfiguration  cobindR configuration object constructor

description

cobindR configuration object constructor

Usage
## S4 method for signature 'character'

cobindRConfiguration(x)

Arguments

x          path to configuration file. NULL by default
Value

cobindR configuration object

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

seqObj

Examples

cfg <- cobindRConfiguration()

cobindR SeqObj object

description of cobindR SeqObj object

Usage

## S4 method for signature 'SeqObj'
comment(x)
## S4 replacement method for signature 'SeqObj,character'
comment(x) <- value

Arguments

x a cobindR seqObj object
value comment to the sequence (character)

Value

comment (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid, name, species, comment, location, sequence

Examples

library(Biostrings)
so <- seqObj(DNAString('A'), id='', name='', species='', comment='', location='')
comment(so)
configuration

configuration of cobindr object

Description
configuration of cobindr object

Usage

## S4 method for signature 'cobindr'
configuration(x)
## S4 replacement method for signature 'cobindr,configuration'
configuration(x) <- value

Arguments

x a cobindr object
value
returns the configuration object used in this cobindR object

Value
cobindR configuration object

Author(s)
Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also
uid, name, sequences, bg_sequences, desc, configuration, binding_sites, bg_binding_sites, pfm, pairs, bg_pairs, p

Examples

cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette_seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak configuration'
cbr <- cobindr(cfg)
configuration(cbr)
configuration-class  

Class "configuration"

Description

Container for experiment description.

Objects from the Class

Objects can be created by calls of the form `new("configuration", fname)`.

Slots

- `id`: Object of class "character" ~~ unique id for internal representation
- `experiment_description`: Object of class "character" ~~ verbal experiment description
- `sequence_source`: Object of class "character" ~~ file path or list of paths
- `sequence_origin`: Object of class "character" ~~ source of sequence data, e.g. ensembl
- `sequence_type`: Object of class "character" ~~ either ChipSeq or Fasta or BED are available
- `bg_sequence_source`: Object of class "character" ~~ file path or list of paths
- `bg_sequence_origin`: Object of class "character" ~~ how the background is obtained - either simulated or from fasta files or from gene ids
- `bg_sequence_type`: Object of class "character" ~~ determines the generation of the background sequences. Possible values: simulated, fasta and geneid
- `species`: Object of class "character" ~~ reference species
- `downstream`: Object of class "numeric" ~~ length of sequence downstream of reference point, e.g. transcription start site
- `upstream`: Object of class "numeric" ~~ length of sequence upstream of reference point, e.g. transcription start site
- `max_distance`: Object of class "numeric" ~~ maximal distance allowed between cooccurring transcription factor binding sites
- `pairs`: Object of class "character" ~~ list of pairs of interesting transcription factors
- `pfm_path`: Object of class "character" ~~ path to pfm matrix file
- `threshold`: Object of class "numeric" ~~ threshold for transcription factor binding site prediction
- `fdrThreshold`: Object of class "numeric" ~~ false discovery rate for filtering results (used in rdfb)
- `date`: Object of class "character" ~~ data of experiment run
- `path`: Object of class "character" ~~ path of configuration file
- `mart`: Object of class "character" ~~ optional mirror for biomart
- `pseudocount`: Object of class "numeric" ~~ sets the pseudocount for the detrending analysis
- `pValue`: Object of class "numeric" ~~ optional p-Value for search with RGadem
downstream

Methods

initialize signature(.Object = "configuration"):
read.background.fasta signature(object = "configuration"):
read.pfm signature(object = "configuration"):
read.sequences signature(object = "configuration"):
write signature(x = "configuration", file = "character"):

Author(s)

Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also

SeqObj cobindr

Examples

showClass("configuration")

downstream 	downstream range [bp] used in experiment
downstream

Description

downstream range [bp] used in experiment

Usage

## S4 method for signature 'configuration'
downstream(x)
## S4 replacement method for signature 'configuration,numeric'
downstream(x) <- value

Arguments

x a cobindR configuration object
value downstream distance [bp] of feature to be included (numeric)

Value

considered downstream range [bp]

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id,experiment_description,sequence_source,sequence_origin,sequence_type,bg_sequence_source,bg_sequence_origin,vocation,pvalue,species,downstream,upstream,max_distance,pairs,pfm_path,threshold,fdrThreshold,path,mart
experiment_description

**Examples**

```r
cfg <- cobindRConfiguration()
downstream(cfg)

experiment_description
description of cobindR or configuration object

**Description**
description of cobindR or configuration object

**Usage**

```r
## S4 method for signature 'configuration'
experiment_description(x)
## S4 replacement method for signature 'configuration,character'
experiment_description(x) <- value
## S4 method for signature 'cobindr'
experiment_description(x)
## S4 replacement method for signature 'cobindr,character'
experiment_description(x) <- value
```

**Arguments**

- `x`: a cobindR or configuration object
- `value`: description

**Value**

experiment description (character)

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

- `id`
- `experiment_description`
- `sequence_source`
- `sequence_origin`
- `sequence_type`
- `bg_sequence_source`
- `bg_sequence_origin`
- `bg_sequence_type`
- `species`
- `downstream`
- `upstream`
- `max_distance`
- `pairs`
- `pfm_path`
- `threshold`
- `fdrThreshold`
- `path`
- `mart`
- `pValue`

**Examples**

```r
cfg <- cobindRConfiguration()

experiment_description(cbr)
```

```
find.pairs

function to find pairs of binding sites for every sequence in a given object of class "cobindr"

Description

find.pairs creates a data frame with all pairs in all sequences within the given distance.

Usage

find.pairs(x, background_scan = FALSE, n.cpu = NA)

fdrThreshold

fdrThreshold of cobindR configuration object

Description

fdrThreshold of cobindR configuration object.

Usage

## S4 method for signature 'configuration'
fdrThreshold(x)
## S4 replacement method for signature 'configuration,numeric'
fdrThreshold(x) <- value

Arguments

x a cobindR configuration object

value the false discovery rate threshold to be used for hit search

Value

fdrThreshold (numeric)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue
get.bindingsite.ranges

Arguments

x
  an object of the class "cobindr", which will hold all necessary information about
  the sequences and the hits.

background_scan
  logical flag, if background_scan = TRUE the pairs for the background sequences
  will be found.

n.cpu
  number of CPUs to be used for parallelization. Default value is 'NA' in which
  case the number of available CPUs is checked and than used.

Value

runObj
  an object of the class "cobindr" including the pairs of transcription factor binding
  sites

Author(s)

Yue-Hien Lee <>

See Also

plot.detrending

---

### Description

Function converts predicted binding sites into a GRanges object (package: GenomicFeatures). This
allows for easy interaction with other tools as well as output of different formats (bed, gff).

### Usage

get.bindingsite.ranges(x, ...)

### Arguments

x
  An object of the class "cobindr", which will hold the predicted binding site
  locations.

... optional additional parameters

### Value

A GRanges object holding the positions of all predicted transcription factor binding sites relative to
the input sequence.

### Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>
get.pairs

See Also

get.pairs write.bindingsites write.bindingsites.table

Examples

# export(get.bindingsite.ranges(runObj), "tfbs_hits.gff3")

get.pairs function to get output of findPairs

Description

Function returns the results of findPairs() as a data frame. The data.frame consists of 6 columns, namely

- a unique id for each pair,
- the unique id of the sequence, where the pair was found,
- the names of the corresponding PFM,
- the unique id for each PFM, and
- the distance window in which the pair occurs.

Usage

## S4 method for signature 'cobindr'
get.pairs(x, background = FALSE)

Arguments

x an object of the class "cobindr", which holds all necessary information about the sequences and the predicted binding sites.

background logical flag. If background is ‘TRUE’ the pairs found in the background sequences are used.

Author(s)

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

See Also

get.significant.pairs, write.bindingsites, write.sequences, write
**get.significant.pairs**  
function to returns the results of detrending as a data.frame

**Description**

get.significant.pairs returns a data.frame of observed distances between the specified pair of PWMs in the foreground set of the sequences as well as the background set of sequences. The distance distribution for the pair in the background is used for detrending.

**Usage**

```r
## S4 method for signature 'cobindr'
get.significant.pairs(x, pwm1, pwm2, bin_length=20, z_value=3, overlap=0, abs.distance=FALSE)
```

**Arguments**

- `x` an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `pwm1` name of the first PWM
- `pwm2` name of the second PWM
- `bin_length` defines size of bins for distance analysis, default value is 20nucleotides
- `z_value` level of significance
- `overlap` number of nucleotides which are allowed for an overlap
- `abs.distance` logical flag

**Author(s)**

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

**See Also**

`plot.detrending`, `get.pairs`, `find.pairs`

---

**id**  
id of cobindR configuration object

**Description**

id of cobindR configuration object.

**Usage**

```r
## S4 method for signature 'configuration'
id(x)
## S4 replacement method for signature 'configuration,character'
id(x) <- value
```
Arguments

x  a cobindR configuration object
value  the identifier of the configuration object

Value

id (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin,

Examples

cfg <- cobindRConfiguration()
id(cfg)

location  location of cobindR SeqObj object

Description

location of cobindR seqObj object (e.g. chr1)

Usage

## S4 method for signature 'SeqObj'
location(x)
## S4 replacement method for signature 'SeqObj,character'
location(x) <- value

Arguments

x  a cobindR seqObj object
value  the location description of the sequence

Value

returns location (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid, name, species, location, comment, sequence
Examples

library(Biostrings)
so <- seqObj(DNAString('A'), id='', name='', species='', comment='', location='')
location(so)

Description

biomart of cobindR configuration object. Set to "ensembl" as default

Usage

## S4 method for signature 'configuration'
mart(x)
## S4 replacement method for signature 'configuration,character'
mart(x) <- value

Arguments

x a cobindR configuration object
value name of biomart to retrieve sequence data

Value

mart (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id.experiment_description.sequence_source.sequence_origin.sequence_type.bg_sequence_source.bg_sequen

Examples

cfg <- cobindRConfiguration()
mart(cfg)
max_distance  

max_distance of cobindR configuration object

Description

max_distance of cobindR configuration object.

Usage

## S4 method for signature 'configuration'
max_distance(x)
## S4 replacement method for signature 'configuration,numeric'
max_distance(x) <- value

Arguments

x 
a cobindR configuration object
value  
the maximal distance of two hits to be considered a pair

Value

max_distance (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, path, mart, pValue

Examples

cfg <- cobindRConfiguration()
max_distance(cfg)

name  

name of cobindR SeqObj object

Description

name of cobindR seqObj object.
pairs

Usage

## S4 method for signature 'SeqObj'
name(x)
## S4 method for signature 'cobindR'
name(x)
## S4 replacement method for signature 'SeqObj,character'
name(x) <- value
## S4 replacement method for signature 'cobindR,character'
name(x) <- value

Arguments

x a cobindR seqObj object
value the name describing the sequence object

Value

name (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid,name,species,location,comment,sequence

Examples

library(Biostrings)
so <- seqObj(NAString('A'), id='', name='', species='',comment='',location='')
nname(so)

pairs motif hit pairs in the foreground sequences

Description

motif hit pairs in the foreground sequences

Usage

## S4 method for signature 'configuration'
pairs(x)
## S4 replacement method for signature 'configuration,character'
pairs(x) <- value
## S4 method for signature 'cobindR'
pairs(x)
## S4 replacement method for signature 'cobindR,data.frame'
pairs(x) <- value
pairs_of_interest

Arguments

x a cobindR configuration object
value for a configuration object, pairs is a character specifying the motif pairs which should be considered. for a cobindR object, pairs is a data.frame holding the detected motif pairs.

Value

pairs (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

detected motif pairs.

Examples

cfg <- cobindRConfiguration()
pairs(cfg)

Description

pairs_of_interest of cobindr object.

Usage

## S4 method for signature 'cobindr'
pairs_of_interest(x)
## S4 replacement method for signature 'cobindr,factor'
pairs_of_interest(x) <- value

Arguments

x a cobindr object
value factors specifying the motif pairs that are to be evaluated

Value

pairs_of_interest (factor)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>
path

See Also

uid, name, sequences, bg_sequences, desc, configuration, binding_sites, bg_binding_sites, pfm, pairs, bg_pairs, pairs_of_interest

Examples

cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette_seq.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak pairs_of_interest'
cbr <- cobindr(cfg)
pairs_of_interest(cbr)

path

path of cobindR configuration object

Description

path of cobindR configuration object.

Usage

## S4 method for signature 'configuration'
path(x)
## S4 replacement method for signature 'configuration, character'
path(x) <- value

Arguments

x a cobindR configuration object
value the path of the loaded configuration file

Value

path (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue

Examples

cfg <- cobindRConfiguration()
path(cfg)
pfm

pfm list used in experiment

Description

pfm list used in experiment

Usage

```r
## S4 method for signature 'cobindr'
pfm(x)
## S4 replacement method for signature 'cobindr,list'
pfm(x) <- value
```

Arguments

- `x`: a cobindr object
- `value`: a list of motif matrices

Value

`pfm` (list of motif matrices)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

`uid`, `name`, `sequences`, `bg_sequences`, `desc`, `configuration`, `binding_sites`, `bg_binding_sites`, `pfm`, `pairs`, `bg_pairs`, `pairs_of_interest`

Examples

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sox_oct_example_vignette_seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak pfm'
cbr <- cobindr(cfg)
 pfm(cbr)
```
pfm_path

path to pfms to be used

Description
path to pfms to be used

Usage
## S4 method for signature 'configuration'
pfm_path(x)
## S4 replacement method for signature 'configuration,character'
pfm_path(x) <- value

Arguments
x a cobindR configuration object
value the path to the folder containing the motif matrices to be used

Value
pfm_path (character)

Author(s)
Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also
id,experiment_description,sequence_source,sequence_origin,sequence_type,bg_sequence_source,bg_sequence_origin,
species,downstream,upstream,max_distance,pairs,threshold,fdrThreshold,path,mart,pValue

Examples

cfg <- cobindRConfiguration()
pfm_path(cfg)

plot.detrending

function to plot distances between a pair of PWMs

Description
plot.detrending plots a histograms of observed distances between the specified pair of PWMs in
the foreground set of the sequences as well as the background set of sequences. The distance
distribution for the pair in the background is used for detrending.

Usage
## S4 method for signature 'cobindr'
plot.detrending(x, pwm1, pwm2, bin_length=20, z_value=3, overlap=0,
abs.distance=FALSE)
plot.gc

Arguments

- `x`: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `seq.ids`: list of sequence identifiers, for which the GC (or CpG) content will be plotted.
- `cpg`: logical flag, if cpg=TRUE the CpG content rather than the GC content will be calculated and plotted.
- `wind.size`: integer describing the window size for GC content calculation
- `sig.test`: logical flag, if sig.test=TRUE wilcoxon.test is performed per individual window against all windows in other sequence at the same position. The significance test might be slow for large number of sequences.
- `hm.margin`: optional argument providing the margin widths for the heatmap (if sig.test=FALSE)
- `frac`: determines the overlap between consecutive windows as fraction wind.size/frac
- `n.cpu`: number of CPUs to be used for parallelization. Default value is 'NA' in which case the number of available CPUs is checked and than used.

Author(s)

Yue-Hien Lee

See Also

plot.pairdistribution, plot.pairdistance

Description

plot.gc calculates the GC (or CpG) content based on a window size for each sequence and plots the content for all sequences as a heatmap over position and sequence.

Usage

```r
## S4 method for signature 'cobindr'
plot.gc(x, seq.ids, cpg = F, wind.size = 50,
       sig.test = F, hm.margin = c(4, 10), frac = 10, n.cpu = NA)
```

Arguments

- `x`: an object of the class "cobindr", which will hold all necessary information about the sequences.
- `seq.ids`: list of sequence identifiers, for which the GC (or CpG) content will be plotted.
- `cpg`: logical flag, if cpg=TRUE the CpG content rather than the GC content will be calculated and plotted.
- `wind.size`: integer describing the window size for GC content calculation
- `sig.test`: logical flag, if sig.test=TRUE wilcoxon.test is performed per individual window against all windows in other sequence at the same position. The significance test might be slow for large number of sequences.
- `hm.margin`: optional argument providing the margin widths for the heatmap (if sig.test=FALSE)
- `frac`: determines the overlap between consecutive windows as fraction wind.size/frac
- `n.cpu`: number of CPUs to be used for parallelization. Default value is 'NA' in which case the number of available CPUs is checked and than used.
Author(s)
Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also
testCpG

Examples

```r
library(Biostrings)

n <- 50 # number of input sequences
l <- 100 # length of sequences
bases <- c("A", "C", "G", "T") # alphabet
# generate random input sequences with two groups with differing GC content
seqs <- sapply(1:(3*n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.3,.22,.2,.28)), collapse=""))
seqs <- append(seqs, sapply(1:(n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.25,.25,.25,.25)), collapse="")))
# save sample sequences in fasta file
tmp.file <- tempfile(pattern = "cobindr_sample_seq", tmpdir = tempdir(), fileext = ".fasta")
writeXStringSet(DNAStringSet(seqs), tmp.file)

cfg <- new(quotesingle.Varconfiguration
slot(cfg, 'sequence_type') <- 'fasta'
slot(cfg, 'sequence_source') <- tmp.file
# avoid complaint of validation mechanism
slot(cfg, 'pfm_path') <- system.file('extdata/pfms',package='cobindR')
slot(cfg, 'pairs') <- ''

runObj <- new('cobindr', cfg, 'test')

plot.gc(runObj, cpg = TRUE)

unlink(tmp.file)
```

---

**plot.pairdistance**

function to plot the distance of the pairs in the sequences

**Description**

For a specified pair of PWMs the function creates histogram plot of distances between pairs of TFs as specified by pwm1 and pwm2

**Usage**

```r
## S4 method for signature 'cobindr'
plot.pairdistance(x, pwm1, pwm2, breaks=50, main=NA, xlab=NA, ylab=NA, background=FALSE)
```
plot.pairdistribution

Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- **pwm1**: name of the first PWM
- **pwm2**: name of the second PWM
- **breaks**: number of breaks to separate the distance distribution into
- **main**: figure title
- **xlab**: label for the x-axis of the figure
- **ylab**: label for the y-axis of the figure
- **background**: flag allowing to plot foreground or background distance distribution

Author(s)

Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also

plot.pairdistribution

plot.pairdistribution  function to plot the distribution of the number of pairs in the sequences

Description

For a specified pair of PWMs the function visualizes in how many sequences how many of the pairs can be found.

Usage

```r
## S4 method for signature 'cobindr'
plot.pairdistribution(x, pwm1, pwm2)
```

Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- **pwm1**: name of the first PWM
- **pwm2**: name of the second PWM

Author(s)

Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also

plot.detrending, plot.pairdistance
plot.positionprofile function to plot a profile over the total number of predicted transcription factor binding sites for each PWM.

Description

plot.positionprofile provides position-wise profile plot over total number of predicted TFBS for each PWM over all input sequences. Windowing is used to provide a smoother appearance, the window size can be adjusted with the window parameter.

Usage

```
## S4 method for signature 'cobindr'
plot.positionprofile(x, wind.len = 50)
```

Arguments

- `x`: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `wind.len`: integer, defining the length of the window for counting the hits.

Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

`plot.positions`

plot.positions function to plot hits for each PWM on the individual sequence

Description

plot.positions plots hits for each PWM on the individual sequence. Which sequences to plot can be specified by providing a list of sequence identifiers `seq.ids`. Which PWMs to plot can be specified as list of PWMs. The total height of the plot can be adjusted via argument `height`.

Usage

```
## S4 method for signature 'cobindr'
plot.positions(x, seq.ids, pwms, main, order.seq = FALSE, wind.size = 400, frac = 10)
```
Arguments

- `x`: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `seq.ids`: list of sequence identifiers, for which the positions of TFBS will be plotted.
- `pwms`: list of PWMs, for which the positions will be visualized. If no list is given, all PWMs in runObj are used.
- `main`: title for the plot, if no title is given than 'predicted TFBS positions per sequence' will be used.
- `order.seq`: logical flag, if TRUE similar patterns of TFBS are shown together. This is computationally expensive for large numbers of sequences.
- `wind.size`: integer describing the windows which will be used to enhance clustering of TFBS patterns. Necessary if order.seq=TRUE
- `frac`: integer

Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

### plot.positions.simple

Function to plot hits for each PWM on the individual sequence

Description

plot.positions plots hits for each PWM on the individual sequence. Which sequences to plot can be specified by providing a list of sequence identifiers `seq.ids`. Which PWMs to plot can be specified as list of PWMs. The total height of the plot can be adjusted via argument `height`.

Usage

```r
## S4 method for signature 'cobindr'
plot.positions.simple(x, seq.ids, pwms, main)
```

Arguments

- `x`: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `seq.ids`: list of sequence identifiers, for which the positions of TFBS will be plotted.
- `pwms`: list of PWMs, for which the positions will be visualized. If no list is given, all PWMs in runObj are used.
- `main`: title for the plot, if no title is given than 'predicted TFBS positions per sequence' will be used.

Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

- `plot.positionprofile`
Description

plot.tfbs.heatmap plots a heatmap of overlaps between all specified PWMs. For each overlap, the significance is determined based on the hypergeometric test. If a file path is specified in pdf.name, the diagram will be written into the specified file.

Usage

```r
## S4 method for signature 'cobindr'
plot.tfbs.heatmap(x, pwms, include.empty.seqs = FALSE)
```

Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- **pwms**: list of PWMs, for which the overlap will be visualized. If no list is given, all PWMs in runObj are used.
- **include.empty.seqs**: logical flag, if include.empty.seqs == TRUE, sequences without hits of the specified PWMs are also included in the diagram.

Details

In this plot for each pair of PWMs the overlap of sequences with hits of the given PWMs is calculated. The number of sequences in each overlap are color-coded in the heatmap. For each overlap the significance is calculated using the hypergeometric test. If the significance is below 0.05 (or below 0.01), the corresponding field is marked with one (or two) *

Warning

- unknown identifier if the list of PWMs contains unknown PWM identifiers a warning is given and the method stops
- no hits if no hits are found in the object, the method gives a warning and stops

Author(s)

Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also

- `plot.tfbs.venndiagram`
**plot.tfbs.venndiagram** function visualize the overlaps of PWM hits over the sequences.

**Description**

The distribution of PWM hits over the sequences is visualized as Venn diagram. If a list of PWM names is provided, only these PWMs are included in the Venn diagram. If `include.empty.seqs == TRUE`, sequences without hits of the specified PWMs are also included in the diagram. If a file path is specified in `pdf.name`, the diagram will be written into the specified file.

**Usage**

```r
## S4 method for signature 'cobindr'
plot.tfbs.venndiagram(x, pwms, include.empty.seqs = FALSE)
```

**Arguments**

- `x` an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- `pwms` list of PWMs, which shall be visualized in the Venn-Diagram. If no list is given, all PWMs in the runObj are used. The package "VennDiagram" only allows Venn plots with up to 4 elements.
- `include.empty.seqs` logical flag, if `include.empty.seqs == TRUE`, sequences without hits of the specified PWMs are also included in the diagram.

**Warning**

- unknown identifier: if the list of PWMs contains unknown PWM identifiers a warning is given and the method stops
- too many PWMs: if more than 4 PWMs are listed a warning is given and the method stops
- no hits: if no hits are found in the object, the method gives a warning and stops

**Author(s)**

Manuela Benary <manuela.benary@cms.hu-berlin.de>

**References**

using the package "VennDiagram" (http://www.biomedcentral.com/1471-2105/12/35/)

**See Also**

`plot.tfbs.heatmap`
plot.tfbslogo

**Description**

plot.tfbslogo produces a sequence logo based on all hits per position weight matrix. If a file path is specified in pdf.name, sequences logos will be written into the specified file.

**Usage**

```r
## S4 method for signature 'cobindr'
plot.tfbslogo(x, pwms)
```

**Arguments**

- `x` Object
- `pwms` vector of names of position weight matrices used for searching the sequences. For each pwm a new sequence logo based on the hits is produced.

**Author(s)**

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

---

predicted2pwm

**Description**

function converts for each input PWM the predicted TFBS hits into a PWM. Function is intended to be used together with the sequence logo creation function 'plot.tfbslogo'.

**Usage**

```r
## S4 method for signature 'cobindr'
predicted2pwm(x, as.pfm=FALSE)
```

**Arguments**

- `x` object of class "cobindr" describing the sequences and the predicted TFBS.
- `as.pfm` logical flag, to indicate whether the function should return a PFM (TRUE) or a PWM (FALSE)

**Value**

- `predPwm` positional frequency matrix based on consensus matrix

**Author(s)**

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>
pseudocount

See Also

plot.tfbslogo

pseudocount of cobindR configuration object

Description

pseudocount of cobindR configuration object. Set to 10 as default

Usage

## S4 method for signature 'configuration'
pseudocount(x)
## S4 replacement method for signature 'configuration,character'
pseudocount(x) <- value

Arguments

x
  a cobindR configuration object

value
  pseudocount for detrending analysis, i.e. the default number in each distance bin.

Value

pseudocount (numeric)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id,experiment_description,sequence_source,sequence_origin,sequence_type.bg_sequence_source.bg_sequence_origin.bg_sequence_type

Examples

cfg <- cobindRConfiguration()
pseudocount(cfg)
**pValue**

**pValue**

---

**pValue** threshold used for motif hit finding

**Description**

pValue threshold used for motif hit finding

**Usage**

```r
## S4 method for signature 'configuration'
pValue(x)
## S4 replacement method for signature 'configuration,numeric'
pValue(x) <- value
```

**Arguments**

- `x`: a cobindR configuration object
- `value`: the p-value threshold used for hit searching

**Value**

pValue threshold (numeric)

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

- `id`, `experiment_description`, `sequence_source`, `sequence_origin`, `sequence_type`, `bg_sequence_source`, `bg_sequence_origin`, `species`, `downstream`, `upstream`, `max_distance`, `pairs`, `pfm_path`, `threshold`, `fdrThreshold`, `path`, `mart`, `pValue`

**Examples**

```r
cfg <- cobindRConfiguration()
pValue(cfg)
```

---

**rtfbs**

---

**rtfbs**

**Description**

function performs TFBS prediction using the package rtfbs

**Usage**

```r
## S4 method for signature 'cobindr'
rtfbs(x, append = F, background_scan = FALSE, n.cpu = NA)
```
Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- **append**: logical flag, if append=TRUE the binding sites will be appended to already existing results
- **background_scan**: logical flag, if background_scan=TRUE the background sequences will be searched for transcription factor binding sites
- **n.cpu**: number of CPUs to be used for parallelization. Default value is 'NA' in which case the number of available CPUs is checked and than used.

Value

- **x**: an object of the class "cobindr" including the predicted transcription factor binding sites

Author(s)

Yue-Hien Lee <>

References

uses the package "rtfbs" (http://cran.r-project.org/web/packages/rtfbs/index.html)

See Also

search.pwm, search.gadem

Examples

```r
############################################################
# use simulated sequences
library(Biostrings)

n <- 400 # number of input sequences
l <- 500 # length of sequences
n.hits <- 250 # number of 'true' binding sites
bases <- c("A","C","G","T") # alphabet
# generate random input sequences with two groups with differing GC content
seqs <- sapply(1:(3*n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.3,.22,.2,.28)), collapse=""))
seqs <- append(seqs, sapply(1:(n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.25,.25,.25,.25)), collapse=""))
path <- system.file("extdata/pfms/myod.tfpfm",package="cobindR")
motif <- read.transfac.pfm(path)[[1]] # get PFM of binding site
# add binding sites with distance specificity
for(position in c(110, 150)) {
  hits <- apply(apply(motif, 2, function(x) sample(x=bases, size=n.hits, prob=x, replace=TRUE)), 1, paste, collapse="")
  pos.hits <- round(rnorm(n.hits, mean=position, sd=8))
  names(pos.hits) <- sample(1:n, n.hits)
  for(i in 1:n.hits) substr(seqs[as.integer(names(pos.hits)[i])],
    start=pos.hits[i], stop=pos.hits[i]+ncol(motif)) <- hits[i]
```
search.gadem

function performs TFBS prediction denovo or based on transfac/jaspar matrices pwms using rGADEM.

Description

function performs TFBS prediction denovo or based on transfac/jaspar matrices pwms using rGADEM. If append=T, predicted hits are appended to the hits in the input object.

Usage

## S4 method for signature 'cobindr'
search.gadem(x, deNovo = FALSE, append = F, background_scan = FALSE)

Arguments

x an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.

deNovo logical flag, if deNOVO=TRUE a denovo search is startet. Otherwise the given PFM are used as seed.

append logical flag, if append=TRUE the binding sites will be appended to already existing results

background_scan logical flag, if background_scan=TRUE the function will search for binding sites in the set of background sequences

Value

x an object of the class "cobindr" including the predicted transcription factor binding sites
Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

References

uses package "rGADEM" (http://www.bioconductor.org/packages/release/bioc/html/rGADEM.html)

See Also

rtfbs, search.pwm

Examples

############################################################
# use simulated sequences
library(Biostrings)

n <- 600 # number of input sequences
l <- 150 # length of sequences
n.hits <- 600 # number of 'true' binding sites
bases <- c("A","C","G","T") # alphabet
# generate random input sequences with two groups with differing GC content
seqs <- sapply(1:(3*n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.3,.22,.2,.28)), collapse=""))
seqs <- append(seqs, sapply(1:(n/4), function(x) paste(sample(bases, l, replace=TRUE, prob=c(.25,.25,.25,.25)), collapse="")))
path <- system.file("extdata/pfms/myod.tfpfm",package="cobindR")
motif <- read.transfac.pfm(path)[[1]] # get PFM of binding site
# add binding sites with distance specificity
for(position in c(70, 90)) {
  hits <- apply(apply(motif, 2, function(x) sample(x=bases, size=n.hits, prob=x, replace=TRUE)), 1, paste, collapse='')
pos.hits <- round(rnorm(n.hits, mean=position, sd=8))
names(pos.hits) <- sample(1:n, n.hits)
  for(i in 1:n.hits) substr(seqs[as.integer(names(pos.hits)[i])], start=pos.hits[i], stop=pos.hits[i]+ncol(motif)) <- hits[i]
}
#save sample sequences in fasta file
tmp.file <- tempfile(pattern = "cobindr_sample_seq", tmpdir = tempdir(), fileext = ".fasta")
writeXStringSet(DNAStringSet(seqs), tmp.file)
#run cobindr
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- tmp.file
sequence_origin(cfg) <- 'artificial sequences'
pfm_path(cfg) <- system.file("extdata/pfms",package="cobindR")
pairs(cfg) <- V$MYOD_01 V$MYOD_01'
runObj <- cobindr(cfg, name="cobindr test using sampled sequences")

# perform tfbs prediction using rGADEM - commented out due to long time required
# runObj.bs <- search.gadem(runObj)
# show results
# plot.positions(runObj.bs)

#clean up
unlink(tmp.file)
**search.pwm**

Function to predict transcription factor binding sites using the method `matchPWM` from package *Biostrings*.

### Description

Function to predict transcription factor binding sites using the method `matchPWM` from package *Biostrings*.

### Usage

```r
## S4 method for signature 'cobindr'
search.pwm(x, min.score = "80\%", append = FALSE, background_scan = FALSE, n.cpu = NA)
```

### Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
- **min.score**: minimal score to define threshold for hits (default = .8)
- **append**: logical flag, if `append=TRUE` the binding sites will be appended to already existing results
- **background_scan**: logical flag, if `background_scan=TRUE` the background sequences will be searched for transcription factor binding sites
- **n.cpu**: number of CPUs to be used for parallelization. Default value is 'NA' in which case the number of available CPUs is checked and than used.

### Value

- **x**: an object of the class "cobindr" including the predicted transcription factor binding sites

### Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

### References

- Uses `matchPWM` from package "Biostrings" (http://www.bioconductor.org/packages/release/bioc/html/Biostrings.html)

### See Also

- *rtfbs*, *search.gadem*
Examples

# use simulated sequences
library(Biostrings)

n <- 400 # number of input sequences
l <- 500 # length of sequences
n.hits <- 250 # number of 'true' binding sites
bases <- c("A","C","G","T") # alphabet

# generate random input sequences with two groups with differing GC content
seqs <- sapply(1:(3*n/4), function(x) paste(sample(bases, l, replace=TRUE,
prob=c(.3,.22,.2,.28)), collapse=""))
seqs <- append(seqs, sapply(1:(n/4), function(x) paste(sample(bases, l, replace=TRUE,
prob=c(.25,.25,.25,.25)), collapse=""))

path <- system.file("extdata/pfms/myod.tfpfm",package="cobindR")
motif <- read.transfac.pfm(path)[[1]] # get PFM of binding site

# add binding sites with distance specificity
for(position in c(110, 150)) {
  hits <- apply(apply(motif, 2, function(x) sample(x=bases, size=n.hits, prob=x,
replace=TRUE)), 1, paste, collapse="")
  pos.hits <- round(rnorm(n.hits, mean=position, sd=8))
  names(pos.hits) <- sample(1:n, n.hits)
  for(i in 1:n.hits) substr(seqs[as.integer(names(pos.hits)[i])], start=pos.hits[i],
stop=pos.hits[i]+ncol(motif)) <- hits[i]
}

# save sample sequences in fasta file
tmp.file <- tempFile(pattern = "cobindr_sample_seq", tmpdir = tempdir(), fileext = ".fasta")
writeXStringSet(DNAStringSet(seqs), tmp.file)

# run cobindr
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- tmp.file
sequence_origin(cfg) <- 'artificial sequences'
pfm_path(cfg) <- system.file("extdata/pfms",package="cobindR")
pairs(cfg) <- "$MYOD_01 $MYOD_01"
runObj <- cobindr(cfg, name="cobindr test using sampled sequences")

# perform tfbs prediction using matchPWM
runObj.bs <- search.pwm(runObj, min.score = 90)

# show results
plot.positionprofile(runObj.bs)

# clean up
unlink(tmp.file)

---

**seqObj**

*cobindR SeqObj object constructor*

**Description**

cobindR SeqObj object constructor

**Usage**

## S4 method for signature

## 'DNAString,character,character,character,character,character,character'

seqObj(seq,id,name,species,comment,location)
**SeqObj-class**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq</td>
<td>DNAString object holding the sequence</td>
</tr>
<tr>
<td>id</td>
<td>id (character)</td>
</tr>
<tr>
<td>name</td>
<td>id (character)</td>
</tr>
<tr>
<td>species</td>
<td>id (character)</td>
</tr>
<tr>
<td>comment</td>
<td>id (character)</td>
</tr>
<tr>
<td>location</td>
<td>id (character)</td>
</tr>
</tbody>
</table>

**Value**

cobindR SeqObj object

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

cobindRConfiguration

**Examples**

```r
library(Biostrings)
so <- seqObj(DNAString('A'), id='', name='', species='', comment='', location='')
sequence(so)
```

---

**Class "SeqObj"**

**Description**

Container for DNA sequence and its meta-data.

**Objects from the Class**

Objects can be created by calls of the form `new("SeqObj", seq, id, species, name, comment, location)`.

**Slots**

- `uid`: Object of class "character" -- unique id for internal representation
- `name`: Object of class "character" -- biological reference name, if available
- `species`: Object of class "character" -- reference species
- `location`: Object of class "character" -- location on the reference genome
- `comment`: Object of class "character" -- comments and notes
- `sequence`: Object of class "DNAString" -- the sequence
Methods

initialize signature(.Object = "SeqObj"): ...
rtfbs.intern signature(object = "SeqObj"): ...
write.fasta signature(sequences = "SeqObj"): ...

Author(s)

Manuela Benary <manuela.benary@cms.hu-berlin.de>

See Also
cobindr configuration

Examples

showClass("SeqObj")

sequence

returns sequence of cobindR SeqObj object

Description

returns sequence of cobindR seqObj object.

Usage

## S4 method for signature 'SeqObj'
sequence(x)
## S4 replacement method for signature 'SeqObj,DNAString'
sequence(x) <- value

Arguments

x a cobindR seqObj object
value DNAString of the actual DNA sequence in this SeqObj

Value

sequence (DNAString)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

cobindr configuration

Examples

library(Biostrings)
so <- seqObj(DNAString('A'), id='', name='', species='', comment='',location='')
sequence(so)
sequences

Description

sequences of cobindr object.

Usage

```r
## S4 method for signature 'cobindr'
sequences(x)
## S4 replacement method for signature 'cobindr,list'
sequences(x) <- value
```

Arguments

- `x` a cobindr object
- `value` the list of input sequences of type SeqObj

Value

sequences (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

uid, name, sequences, bg_sequences, desc, configuration, binding_sites, bg_binding_sites, pfm, pairs, bg_pairs, p

Examples

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/sup_example_vignette_seqs.fasta', package='cobindR')
sequence_origin(cfg) <- 'Mouse Embryonic Stem Cell Example ChIP-Seq Oct4 Peak Sequences'
cbr <- cobindr(cfg)
length(sequences(cbr))
```
sequence_origin returns sequence_origin of cobindR configuration object

Description

returns sequence_origin of cobindR configuration object.

Usage

## S4 method for signature 'configuration'
sequence_origin(x)
## S4 replacement method for signature 'configuration,character'
sequence_origin(x) <- value

Arguments

x a cobindR configuration object
value the origin of the sequence

Value

sequence_origin (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue

Examples

cfg <- cobindRConfiguration()
sequence_origin(cfg)

sequence_source returns sequence_source of cobindR configuration object

Description

returns sequence_source of cobindR configuration object.

Usage

## S4 method for signature 'configuration'
sequence_source(x)
## S4 replacement method for signature 'configuration,character'
sequence_source(x) <- value
sequence_type

Arguments

x a cobindR configuration object
value the source of which the sequence is retrieved

Value

sequence_source (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue

Examples

cfg <- cobindRConfiguration()
sequence_source(cfg)

sequence_type sequence type of cobindR configuration object

Description

sequence type of cobindR configuration object

Usage

## S4 method for signature 'configuration'
sequence_type(x)
## S4 replacement method for signature 'configuration,character'
sequence_type(x) <- value

Arguments

x a cobindR configuration object
value the type of the sequence used in this experiment (e.g. promotor)

Value

sequence_type (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin, species, downstream, upstream, max_distance, pairs, pfm_path, threshold, fdrThreshold, path, mart, pValue
Examples
cfg <- cobindRConfiguration()
sequence_type(cfg)

species

species of cobindR configuration or SeqObj

Description

species of cobindR configuration or SeqObj

Usage

## S4 method for signature 'configuration'
species(object)
## S4 replacement method for signature 'configuration'
species(object) <- value
## S4 method for signature 'SeqObj'
species(object)
## S4 replacement method for signature 'SeqObj'
species(object) <- value

Arguments

object a cobindR configuration object
value name of species in this experiment or SeqObj

Value

sequence / experiment species (character)

Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

See Also

id, experiment_description, sequence_source, sequence_origin, sequence_type, bg_sequence_source, bg_sequence_origin,

Examples

cfg <- cobindRConfiguration()
species(cfg)
testCpG

function to cluster sequences based on their CpG and GC content

Description

diagnostical function - GC content and CpG content are clustered using 2D gaussian models (Mclust). FALSE is returned if > max.clust (default=1) subgroups are found using the bayesian information criterion (BIC). If do.plot=TRUE, the results are visualized.

Usage

## S4 method for signature 'cobindr'
testCpG(x, max.clust = 4, do.plot = F, n.cpu = NA)

Arguments

x  an object of the class "cobindr", which will hold all necessary information about the sequences and the hits.
max.clust  integer describing the maximal number of clusters which are used for separating the data.
do.plot  logical flag, if do.plot=TRUE a scatterplot for the GC and CpG content for each sequence is produced and the clusters are color coded.
n.cpu  number of CPUs to be used for parallelization. Default value is 'NA' in which case the number of available CPUs is checked and than used.

Value

result  logical flag, FALSE is returned if more than one subgroups are found using the bayesian information criterion (BIC)
gc  matrix with rows corresponding to sequences and columns corresponding to GC and CpG content

Author(s)

Robert Lehmann <r.lehmann@biologie.hu-berlin.de>

References

the method uses clustering functions from the package "mclust" (http://www.stat.washington.edu/mclust/)

See Also

plot.gc
Examples

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/example.fasta', package='cobindR')
# avoid complaint of validation mechanism
pfm_path(cfg) <- system.file('extdata/pfms', package='cobindR')
pairs(cfg) <- ''
runObj <- cobindr(cfg)
testCpG(runObj, max.clust = 2, do.plot = TRUE)
```

---

### threshold

#### threshold used in motif hit finding

**Description**

threshold used in motif hit finding

**Usage**

```r
## S4 method for signature 'configuration'
threshold(x)
## S4 replacement method for signature 'configuration,numeric'
threshold(x) <- value
```

**Arguments**

- `x` a cobindR configuration object
- `value` the hit threshold

**Value**

threshold (numeric)

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

- `id`
- `experiment_description`
- `sequence_source`
- `sequence_origin`
- `sequence_type`
- `bg_sequence_source`
- `bg_sequence_origin`
- `bg_sequence_type`
- `species`
- `downstream`
- `upstream`
- `max_distance`
- `pairs`
- `pfm_path`
- `threshold`
- `fdrThreshold`
- `path`
- `mart`
- `pValue`

**Examples**

```r
cfg <- cobindRConfiguration()
threshold(cfg)
```
**uid**

uid of cobindR SeqObj object

---

**Description**

uid of cobindR seqObj object.

**Usage**

```r
## S4 method for signature 'SeqObj'
uid(x)
## S4 method for signature 'cobindr'
uid(x)
## S4 replacement method for signature 'SeqObj,character'
uid(x) <- value
## S4 replacement method for signature 'cobindr,character'
uid(x) <- value
```

**Arguments**

- `x` a cobindR seqObj object
- `value` the unique id of the sequence or cobindr object

**Value**

uid (character)

**Author(s)**

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

**See Also**

- `uid`
- `name`
- `species`
- `location`
- `comment`
- `sequence`

**Examples**

```r
library(Biostrings)
so <- seqObj(DNAString('A'), id='', name='', species='', comment='', location='')
uid(so)
```
### Description

upstream range [bp] used in experiment

### Usage

```r
## S4 method for signature 'configuration'
upstream(x)
## S4 replacement method for signature 'configuration,numeric'
upstream(x) <- value
```

### Arguments

- `x`: a cobindR configuration object
- `value`: upstream distance [bp] of feature to be included (numeric)

### Value

considered upstream range [bp]

### Author(s)

Rob Lehmann <r.lehmann@biologie.hu-berlin.de>

### See Also

`id`, `experiment_description`, `sequence_source`, `sequence_origin`, `sequence_type`, `bg_sequence_source`, `bg_sequence_origin`, `species`, `downstream`, `upstream`, `max_distance`, `pairs`, `pfm_path`, `threshold`, `fdrThreshold`, `path`, `mart`, `pValue`

### Examples

```r
cfg <- cobindRConfiguration()
upstream(cfg)
```

### Description

writes predicted binding sites as a BED file.

### Usage

```r
## S4 method for signature 'cobindr'
write.bindingsites(x, file = NULL, background = FALSE)
```
**write.bindingsites.table**

### Arguments

- **x**
  - An object of the class "cobindr", which holds all necessary information about the sequences and the predicted binding sites.

- **file**
  - Path to file. If filename is "NULL" a filename is generated based on the name of the object of class "cobindr".

- **background**
  - Logical flag. If background is 'TRUE' the binding sites found in the background sequences are used.

### Note

At the moment write.bindingsites() only works for sequences based on gene ids. Otherwise please use write.bindingsites.table().

### Author(s)

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

### See Also

write.bindingsites.table, write.pairs, write.sequences, write

---

**write.bindingsites.table**

*Function to write predicted TFBS into a tab-separated file.*

### Description

Function to write predicted TFBS into a tab-separated file.

### Usage

```r
## S4 method for signature 'cobindr'
write.bindingsites.table(x, file = NULL)
```

### Arguments

- **x**
  - An object of the class "cobindr", which will hold all necessary information about the sequences and the predicted binding sites.

- **file**
  - Path to file. If filename is 'NULL' a filename is generated based on the name of the object of class "cobindr".

### Author(s)

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

### See Also

write.pairs, write.bindingsites, write.sequences, write
write.pairs

function to write output of findPairs into file

Description

Function writes the results of findPairs() as a tab-separated file. The file consists of 6 columns, namely

- a unique id for each pair,
- the unique id of the sequence, where the pair was found,
- the names of the corresponding PFMs,
- the unique id for each PFM, and
- the distance window in which the pair occurs.

Usage

## S4 method for signature 'cobindr'
write.pairs(x, file = NULL, background = FALSE)

Arguments

x an object of the class "cobindr", which holds all necessary information about the sequences and the predicted binding sites.

file path to file. If filename is 'NULL' a filename is generated based on the name of the object of class "cobindr".

background logical flag. If background is 'TRUE' the pairs found in the background sequences are used.

Author(s)

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

See Also

write.bindingsites.table, write.bindingsites, write.sequences, write

write.sequences

writes the sequences of a cobindr-object into a fasta file.

Description

writes the sequences of a cobindr-object into a fasta file.

Usage

## S4 method for signature 'cobindr'
write.sequences(x, slotname = "sequences", file = NULL)
write.sequences

Arguments

- **x**: an object of the class "cobindr", which will hold all necessary information about the sequences.
- **slotname**: string, describing whether to use foreground sequences (default) or background sequences.
- **file**: path to file. If filename is 'NULL' a filename is generated based on the name of the object of class "cobindr".

Author(s)

Stefan Kroeger <kroeger@informatik.hu-berlin.de>

See Also

write.bindingsites.table, write.bindingsites, write.pairs, write

Examples

```r
cfg <- cobindRConfiguration()
sequence_type(cfg) <- 'fasta'
sequence_source(cfg) <- system.file('extdata/example.fasta', package='cobindR')
# avoid complaint of validation mechanism
pfm_path(cfg) <- system.file('extdata/pfms', package='cobindR')
pairs(cfg) <- ''
runObj <- cobindr(cfg)
write.sequences(runObj, file = file.path(tempfile("example.txt", tempdir())))
```
## Index

*Topic **IO**
  - get.pairs, 18
  - get.significant.pairs, 19
  - write.bindingsites, 52
  - write.bindingsites.table, 53
  - write.pairs, 54
  - write.sequences, 54

*Topic **SeqObj**
  - comment, 11
  - location, 20
  - name, 22
  - sequence, 44
  - uid, 51

*Topic **accessor**
  - bg_binding_sites, 3
  - bg_pairs, 4
  - bg_sequence_origin, 6
  - bg_sequence_source, 6
  - bg_sequence_type, 7
  - bg_sequences, 5
  - binding_sites, 8
  - comment, 11
  - configuration, 12
  - downstream, 14
  - experiment_description, 15
  - fdrThreshold, 16
  - id, 19
  - location, 20
  - mart, 21
  - max_distance, 22
  - name, 22
  - pairs, 23
  - pairs_of_interest, 24
  - path, 25
  - pfm, 26
  - pfm_path, 27
  - pseudocount, 36
  - pValue, 37
  - sequence, 44
  - sequence_origin, 46
  - sequence_source, 46
  - sequence_type, 47
  - sequences, 45
  - species, 48
  - threshold, 50
  - uid, 51
  - upstream, 52

*Topic **classes**
  - cobindr-class, 9
  - configuration-class, 13
  - SeqObj-class, 43

*Topic **clust**
  - testCpG, 49

*Topic **cobindR**
  - bg_binding_sites, 3
  - bg_pairs, 4
  - bg_sequence_origin, 6
  - bg_sequence_source, 6
  - bg_sequence_type, 7
  - bg_sequences, 5
  - binding_sites, 8
  - cobindRConfiguration, 10
  - comment, 11
  - configuration, 12
  - downstream, 14
  - experiment_description, 15
  - fdrThreshold, 16
  - id, 19
  - location, 20
  - mart, 21
  - max_distance, 22
  - name, 22
  - pairs, 23
  - pairs_of_interest, 24
  - path, 25
  - pfm, 26
  - pfm_path, 27
  - pseudocount, 36
  - pValue, 37
  - seqObj, 42
  - sequence, 44
  - sequence_origin, 46
  - sequence_source, 46
  - sequence_type, 47
  - sequences, 45
  - species, 48
detrending, cobindr-method
(cobindr-class), 9
downstream, 6, 7, 14, 15, 16, 20–22, 24,
25, 27, 36, 37, 46–48, 50, 52
downstream<-(downstream), 14
downstream<-, configuration, numeric-method
(downstream), 14
experiment_description, 6, 7, 14, 15, 16, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 52
experiment_description, cobindr-method
(experiment_description), 15
experiment_description, configuration-method
(experiment_description), 15
experiment_description<-(experiment_description), 15
experiment_description<-, cobindr, character-method
(experiment_description), 15
experiment_description<-, configuration, character-method
(experiment_description), 15
fdrThreshold, 6, 7, 14–16, 16, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 52
fdrThreshold, configuration-method
(fdrThreshold), 16
fdrThreshold<-(fdrThreshold), 16
fdrThreshold<-, configuration, numeric-method
(fdrThreshold), 16
find.pairs, 3, 16, 19
find.pairs, cobindr-method
(cobindr-class), 9
generate.background, cobindr-method
(cobindr-class), 9
generate.bindingsite.ranges, 17
generate.bindingsite.ranges, cobindr-method
(cobindr-class), 9
get.pairs, 18, 19
get.pairs, cobindr-method (get.pairs), 18
get.pairs-method (get.pairs), 18
get.significant.pairs, 18, 19
generate.significant.pairs, cobindr-method
(get.significant.pairs), 19
generate.significant.pairs-method
(get.significant.pairs), 19
id, 6, 7, 14–16, 19, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 52
id, configuration-method (id), 19
id<-(id), 19
id<-,configuration,character-method
   (id), 19
initialize, cobindr-method
   (cobindr-class), 9
initialize, configuration-method
   (configuration-class), 13
initialize, SeqObj-method
   (SeqObj-class), 43
input.pwm, cobindr-method
   (cobindr-class), 9
location, 11, 20, 20, 23, 44, 51
location<-, (location), 20
location<-, SeqObj-method (location), 20
mart, 6, 7, 14–16, 20, 21, 21, 22, 24, 25, 27, 26, 36, 37, 46–48, 50, 52
mart, configuration-method (mart), 21
mart<-, configuration, character-method (mart), 21
max_distance, 6, 7, 14–16, 20–22, 22, 24, 25, 27, 26, 36, 37, 46–48, 50, 52
max_distance, configuration-method (max_distance), 22
max_distance<-, configuration, character-method (max_distance), 22
max_distance<-, (max_distance), 22
max_distance<-, configuration, numeric-method (max_distance), 22
name, 4, 5, 8, 11, 12, 20, 22, 23, 25, 26, 44, 45, 51
name, cobindr-method (name), 22
name, SeqObj-method (name), 22
name<-, (name), 22
name<-, cobindr, character-method (name), 22
name<-, SeqObj, character-method (name), 22
pairs, 4–8, 12, 14–16, 20–22, 23, 24–27, 36, 37, 45–48, 50, 52
pairs, cobindr-method (pairs), 23
pairs, configuration-method (pairs), 23
pairs<-, (pairs), 23
pairs<-, cobindr, data.frame-method (pairs), 23
pairs<-, configuration, character-method (pairs), 23
pairs_of_interest, 4, 5, 8, 12, 24, 25, 26, 45
pairs_of_interest, cobindr-method (pairs_of_interest), 24
pairs_of_interest<-, (pairs_of_interest), 24
pairs_of_interest<-, cobindr, factor-method (pairs_of_interest), 24
path, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 52
path, configuration-method (path), 25
path<-, (path), 25
path<-, configuration, character-method (path), 25
pfm, 4, 5, 8, 12, 25, 26, 26, 45
pfm, cobindr-method (pfm), 26
pfm<-, (pfm), 26
pfm<-, cobindr, list-method (pfm), 26
pfm_path, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 52
pfm_path, configuration-method (pfm_path), 27
pfm_path<-, (pfm_path), 27
pfm_path<-, configuration, character-method (pfm_path), 27
plot.detrending, 17, 19, 27, 30
plot.detrending, cobindr-method (plot.detrending), 27
plot.detrending-method (plot.detrending), 27
plot.gc, 28, 49
plot.gc, cobindr-method (plot.gc), 28
plot.gc-method (plot.gc), 28
plot.pairdistance, 28, 29, 30
plot.pairdistance, cobindr-method (plot.pairdistance), 29
plot.pairdistance-method (plot.pairdistance), 29
plot.pairdistribution, 28, 30, 30
plot.pairdistribution, cobindr-method (plot.pairdistribution), 30
plot.pairdistribution-method (plot.pairdistribution), 30
plot.positionprofile, 31, 32
plot.positionprofile, cobindr-method (plot.positionprofile), 31
plot.positionprofile-method (plot.positionprofile), 31
plot.positions, 31, 31
plot.positions, cobindr-method (plot.positions), 31
plot.positions-method (plot.positions), 31
plot.positions.simple, 32
plot.positions.simple, cobindr-method (plot.positions.simple), 32
SeqObj, 10, 14
seqObj, 11, 42
seqObj, DNAString, character, character, character, character, character
(pseudocount), 36
SeqObj-class, 43
seqObj-method (seqObj), 42
sequence, 11, 20, 23, 44, 45, 51
sequence, SeqObj-method (sequence), 44
sequence<-, (sequence), 44
sequence<-, SeqObj, DNAString-method (sequence), 44
sequence_origin, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46, 47, 48, 50, 52
sequence_origin, configuration-method (sequence_origin), 46
sequence_origin<-, (sequence_origin), 46
sequence_origin<-, configuration, character-method (sequence_origin), 46
sequence_source, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46, 47, 48, 50, 52
sequence_source, configuration-method (sequence_source), 46
sequence_source<-, (sequence_source), 46
sequence_source<-, configuration, character-method (sequence_source), 46
sequence_type, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46, 47, 48, 50, 52
sequence_type, configuration-method (sequence_type), 47
sequence_type<-, (sequence_type), 47
sequence_type<-, configuration, character-method (sequence_type), 47
sequences, 4, 8, 12, 25, 26, 45, 45
sequences, cobindr-method (sequences), 45
sequences<-, (sequences), 45
sequences<-, cobindr, list-method (sequences), 45
species, 6, 7, 11, 14–16, 20–25, 27, 36, 37, 44, 46–48, 48, 50–52
species, configuration-method (species), 48
species, SeqObj-method (species), 48
species<-, (species), 48
species<-, configuration-method (species), 48
species<-, SeqObj-method (species), 48
testCpG, 29, 49
testCpG, cobindr-method (testCpG), 49
testCpG-method (testCpG), 49
threshold, 6, 7, 14–16, 20–22, 24, 25, 27, 36, 37, 46–48, 50, 50, 52
INDEX

threshold, configuration-method
(threshold), 50
threshold<- (threshold), 50
threshold<-, configuration, numeric-method
(threshold), 50

uid, 4, 5, 8, 11, 12, 20, 23, 25, 26, 44, 45, 51,
51
uid, cobindr-method (uid), 51
uid, SeqObj-method (uid), 51
uid<- (uid), 51
uid<-, cobindr, character-method (uid), 51
uid<-, SeqObj, character-method (uid), 51
upstream, 6, 7, 14–16, 20–22, 24, 25, 27, 36,
37, 46–48, 50, 52, 52
upstream, configuration-method
(upstream), 52
upstream<- (upstream), 52
upstream<-, configuration, numeric-method
(upstream), 52

write, 18, 53–55
write, cobindr, character-method
(cobindr-class), 9
write, configuration, character-method
(configuration-class), 13
write.bindingsites, 18, 52, 53–55
write.bindingsites, cobindr-method
(write.bindingsites), 52
write.bindingsites-method
(write.bindingsites), 52
write.bindingsites.table, 53, 53, 54, 55
write.bindingsites.table, cobindr-method
(write.bindingsites.table), 53
write.bindingsites.table-method
(write.bindingsites.table), 53
write.fasta, SeqObj-method
(SeqObj-class), 43
write.pairs, 53, 54, 55
write.pairs, cobindr-method
(write.pairs), 54
write.pairs-method (write.pairs), 54
write.sequences, 18, 53, 54, 54
write.sequences, cobindr-method
(write.sequences), 54
write.sequences-method
(write.sequences), 54