Package 'PscanR'

November 13, 2026

Type Package

Title Motif Enrichment Analysis of Regulatory DNA Sequences

Version 0.99.4

Encoding UTF-8

Depends R (>= 4.6.0)

Imports BiocFileCache, BiocParallel, BiocGenerics, Biostrings, ExperimentHub, ggplot2, grDevices, methods, pheatmap, RSQLite, stats, TFBSTools, UCSC.utils, utils

Suggests AnnotationDbi, BiocStyle, BSgenome, BSgenome.Athaliana.TAIR.TAIR9, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Mmusculus.UCSC.mm10, BSgenome.Scerevisiae.UCSC.sacCer3, GenomicFeatures, GenomicRanges, GenomeInfoDb, IRanges, JASPAR2020, JASPAR2022, JASPAR2024, knitr, org.Hs.eg.db, org.Mm.eg.db, org.Sc.sgd.db, PscanRBackgrounds, rmarkdown, testthat, TxDb.Scerevisiae.UCSC.sacCer3.sgdGene, txdbmaker

biocViews SequenceMatching, Transcription, GeneRegulation, MotifAnnotation

Description R implementation of the 'Pscan' algorithm for motif enrichment analysis in regulatory DNA sequences, based on Zambelli et al. (2009) <doi:10.1093/nar/gkp464>.
'PscanR' scans a set of DNA sequences (e.g., promoters) from co-regulated or co-expressed genes, using transcription factor binding motifs from databases such as JASPAR. It identifies significantly enriched motifs, providing thus hints on which transcription factors could be common regulators of the genes studied, together with the location of their candidate binding sites in the sequences.

License GPL-3

VignetteBuilder knitr

URL https://github.com/Federico77z/PscanR

BugReports https://github.com/Federico77z/PscanR/issues

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

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

git_branch devel

git_last_commit be6299a

git_last_commit_date 2026-11-11

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Federico Zambelli [aut, cre] (ORCID:
<https://orcid.org/0000-0003-3487-4331>),
Giulio Pavesi [aut] (ORCID: <https://orcid.org/0000-0001-5705-6249>),
Diana Betelli [ctb],
Italy's National Recovery and Resilience Plan (PNRR) [fnd] (project
ELIXIRNextGenIT, R0000010)

Maintainer Federico Zambelli <federico.zambelli@unimi.it>

Contents

pscan . . . 3
PscanR . . . 6
pscan_filtered . . . 7
pscan_full_bg . . . 9
PSMatrix . . . 10
PSMatrixList . . . 12
ps_available_bg . . . 13
ps_bg_avg . . . 14
ps_bg_avg,PSMatrix-method . . . 15
ps_bg_avg<- . . . 16
ps_bg_avg<-,PSMatrix-method . . . 16
ps_bg_seq_names . . . 17
ps_bg_seq_names,PSMatrix-method . . . 18
ps_bg_size . . . 19
ps_bg_size,PSMatrix-method . . . 19
ps_bg_size<- . . . 20
ps_bg_size<-,PSMatrix-method . . . 21
ps_bg_std_dev . . . 21
ps_bg_std_dev,PSMatrix-method . . . 22
ps_bg_std_dev<- . . . 23
ps_bg_std_dev<-,PSMatrix-method . . . 23
ps_build_bg . . . 24
ps_build_bg_from_table . . . 26
ps_density_plot . . . 27
ps_fg_avg . . . 29
ps_fg_avg,PSMatrix-method . . . 30
ps_fg_size . . . 30
ps_fg_size,PSMatrix-method . . . 31
ps_get_bg_table . . . 32
ps_hitpos_map . . . 33
ps_hits_oligo . . . 34
ps_hits_oligo,PSMatrix-method . . . 35
ps_hits_oligo_bg . . . 36
ps_hits_oligo_bg,PSMatrix-method . . . 37
ps_hits_pos . . . 38

ps_hits_pos,PSMatrix-method . . . 39
ps_hits_pos_bg . . . 39
ps_hits_pos_bg,PSMatrix-method . . . 40
ps_hits_score . . . 41
ps_hits_score,PSMatrix-method . . . 42
ps_hits_score_bg . . . 42
ps_hits_score_bg,PSMatrix-method . . . 43
ps_hits_size . . . 44
ps_hits_size,PSMatrix-method . . . 45
ps_hits_strand . . . 46
ps_hits_strand,PSMatrix-method . . . 46
ps_hits_strand_bg . . . 47
ps_hits_strand_bg,PSMatrix-method . . . 48
ps_hits_table . . . 49
ps_hits_table,PSMatrix-method . . . 50
ps_hits_z . . . 51
ps_hits_z,PSMatrix-method . . . 51
ps_hit_score_plot . . . 52
ps_load_select_transcripts . . . 53
ps_motif_barplot . . . 54
ps_motif_class . . . 56
ps_pvalue . . . 57
ps_pvalue,PSMatrix-method . . . 58
ps_results_table . . . 58
ps_retrieve_bg . . . 60
ps_retrieve_bg_from_file . . . 62
ps_scan . . . 63
ps_scan,PSMatrix-method . . . 64
ps_selection_summary . . . 65
ps_select_promoters . . . 66
ps_seq_names . . . 68
ps_seq_names,PSMatrix-method . . . 69
ps_transcript_legend . . . 70
ps_transcript_legend,PSMatrixList-method . . . 70
ps_write_bg_to_file . . . 71
ps_zscore . . . 72
ps_zscore,PSMatrix-method . . . 73
ps_zscore_heatmap . . . 74
ps_z_table . . . 75
show,PSMatrix-method . . . 76

Index **78**

`pscan` *Executes the Pscan algorithm on a set of regulatory sequences.*

Description

This function computes alignment scores between regulatory sequences (a set of gene promoters, 'x') and position weight matrices, which quantify potential binding affinities of transcription factors in that region. These matrices can be sourced from public databases such as JASPAR. The scanning is performed by the Pscan algorithm.

Usage

```
pscan(x, pfms, BPPARAM = BiocParallel::SerialParam(), BPOPTIONS = bpoptions())
```

Arguments

x	A 'DNAStringSet' object containing the set of regulatory sequences from co-regulated or co-expressed genes (i.e. a set of gene promoters). See the Biostrings package for details. Duplicated sequences are removed. All promoters should have the same length; if they do not, only those of the most common length are kept, with a warning. If the promoters kept are shorter than any motif, the function stops with an error that names those motifs.
pfms	A 'PSMatrixList' object containing PWMs and background statistics. For background statistics we refer to the standard deviation and average of best-hit scores when the supplied background reference set is scanned with the position weight matrices. This is used to assert the statistical enrichment of motif occurrences in co-expressed or co-regulated genes. The background-building functions described in See Also create suitable 'PSMatrixList' objects.
BPPARAM	The BPPARAM used by bplapply. See BiocParallel package. This argument is passed to 'BiocParallel::bplapply'. The default, 'BiocParallel::SerialParam()', uses the current R process. Pass an explicit backend to enable parallel execution over motifs. You can specify BPPARAM = BiocParallel::SnowParam(8) on all operating systems, or BPPARAM = BiocParallel::MulticoreParam(8) on Unix-like systems to use, for example, 8 cores.
BPOPTIONS	The BPOPTIONS used by bplapply. See BiocParallel package. This argument is passed to 'BiocParallel::bplapply'. The default is 'bpoptions()'. Some useful tasks: bpoptions(progressbar = TRUE, log = TRUE). progressbar = TRUE enables a progress bar that can be useful when processing many tasks. log = TRUE enables logging to debug each step of the parallel tasks.

Details

The 'pscan' function performs sequence scanning using the 'ps_scan' method for individual PWMs, accounting for both the forward and reverse complement strands, ensuring no potential binding sites are missed. The method extracts all k-mer substrings of the input sequences with the same length of the motif and evaluates a score for both the forward and reverse strand based on how well the k-mer matches the Transcription Factor Binding Motif provided by the PWM. For each input sequence scanned with individual PWM, only the k-mer with the highest score is selected.

Before scanning, exact duplicate sequences are collapsed. Sequences whose width differs from the maximum input width and sequences containing more than 50 percent 'N' are removed with a warning. The following slots therefore describe the retained unique foreground sequences:

- ps_hits_score: the highest matching value found for each sequence.
- ps_hits_pos: position of the best TFBS match.
- ps_hits_strand: the DNA strand of the TFBS ('+' for forward and '-' for reverse).
- ps_hits_oligo: the sequence of the binding site.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A ‘PSMatrixList‘ with each retained sequence’s best score, start, strand, and oligo stored in every matrix.

Foreground size

The Pscan z-score is ‘sqrt(n)‘ times the standardised difference between the foreground and background mean scores, so it grows with the size of the foreground rather than only with the strength of the signal. And since the background contains the foreground, a foreground of ‘n‘ out of ‘N‘ is compared against a set that partly consists of itself: the statistic’s foreground-versus-rest difference is shrunk by ‘1 - f‘, with ‘f = n / N‘. Both effects are invisible in the reported number, which is why a foreground above a tenth of the background raises a warning rather than being left to the reader.

The threshold is ‘getOption("PscanR.foreground.max_fraction", 0.1)‘; set it to ‘Inf‘ to silence the check, or to a smaller fraction to tighten it.

‘ps_bg_size()‘ is a per-motif slot rather than a run constant. It is ‘NA_integer_‘ for any motif absent from the background table, and a hand-built table can legitimately give different sizes per motif, so ‘N‘ is the median of the usable values and nothing is reported when none are. The Z-test uses the number of non-‘NA‘ foreground scores as ‘n‘, while ‘ps_fg_size()‘ reports the number of retained foreground sequences.

See Also

ps_build_bg, background table builder, and background file reader.

Examples

```
# Load the promoter sequences for hg38 (Homo sapiens), promoter regions:
# -200 +50 bp in respect to the TSS.
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:10]

# Build a small background from bundled files.
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
background <- ps_retrieve_bg_from_file(bg_path, matrices)
motif_ids <- c(
    "MA0506.1", "MA0632.2", "MA0615.1",
    "MA0076.2", "MA0645.1", "MA0838.1"
)
background <- background[motif_ids]

# Execute the PScan algorithm
results <- pscan(prom_seq, background,
    BPPARAM = BiocParallel::SerialParam()
)

ps_results_table(results)
```

PscanR *PscanR: An R Implementation for the 'Pscan' Algorithm*

Description

'PscanR' provides an R implementation of the 'Pscan' algorithm for transcription factor binding site motif analysis. The package is designed to work with Bioconductor data objects: it extends PFMatrix and PFMatrixList from the TFBSTools package for transcription factor binding profiles and employs Biostrings DNAStringSet for promoter sequences. PscanR supports parallel execution through the BiocParallel package. PscanR includes functions to build the background for a set of promoter sequences, to scan a foreground set of promoter sequences, and to visualize and plot the results.

Main functions

- ps_retrieve_bg: Build background matrices from pre-computed files.
- pscan: Pscan algorithm utilization on a set of gene promoters.
- ps_results_table: Data visualization.

Author(s)

Federico Zambelli [aut, cre], Giulio Pavesi [aut]

Maintainer: Federico Zambelli <federico.zambelli@unimi.it>

References

'Pscan' Web: http://www.beaconlab.it/pscan/

Zambelli F, Pesole G, Pavesi G. Pscan: finding over-represented transcription factor binding site motifs in sequences from co-regulated or co-expressed genes. Nucleic Acids Res. 2009 Jul;37(Web Server issue):W247-52. doi: 10.1093/nar/gkp464. Epub 2009 May 31. PMID: 19487240; PMCID: PMC2703934.

See Also

Useful links:

- https://github.com/Federico77z/PscanR
- Report bugs at https://github.com/Federico77z/PscanR/issues

Examples

```
motifs <- readRDS(system.file(
    "extdata", "J2020.rds", package = "PscanR"
))
background <- ps_retrieve_bg_from_file(system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
), motifs)
# NRF1 and CTCF illustrate enriched and background-like motif scores.
background <- background[c("MA0506.1", "MA0139.1")]
# Historical NRF1-example promoters; the original experiment is unknown.
```

```
sequences <- readRDS(system.file(
    "extdata", "prom_seq.rds", package = "PscanR"
))
# Adjusted p-values cover only these two motifs.
ps_results_table(pscan(sequences, background))
```

`pscan_filtered` *Filter and Scan Promoter Regions with Pscan*

Description

This function filters input promoter sequences based on their affinity with a specified JASPAR matrix before applying the Pscan algorithm with the background reference.

Usage

```
pscan_filtered(
  prom_seq,
  Jmatrix,
  n = 1,
  background,
  BPPARAM = BiocParallel::SerialParam(),
  BPOPTIONS = bpoptions()
)
```

Arguments

`prom_seq`	A DNAStringSet object. The promoter sequences to be filtered. Duplicated sequences are removed. All promoters should have the same length; if they do not, only those of the most common length are kept, with a warning. If the promoters kept are shorter than any motif, the function stops with an error that names those motifs.
`Jmatrix`	A PSMatrix object. A JASPAR matrix containing background average and standard deviation statistics, used to filter promoter sequences.
`n`	A numeric value determining the threshold for filtering. The threshold is computed as n*ps_bg_std_dev(Jmatrix) + ps_bg_avg(Jmatrix). Positive values retain sequences with higher affinity, while negative values retain sequences with lower affinity. Default is 1.
`background`	A PSMatrixList object. It contains PWMs and background statistics. For background statistics we refer to the standard deviation and average of hits scores when the background (set of promoters of all the transcript in the organism of study) is scanned with the position weight matrices. This is used to assert the statistical enrichment of motif occurrences in co-expressed or co-regulated genes. See `ps_build_bg`, `ps_retrieve_bg_from_file`, `ps_build_bg_from_table`, `ps_retrieve_bg` for how to create 'PSMatrixList' objects that contain background statistics.
`BPPARAM`	The BPPARAM used by bplapply. See BiocParallel package. This argument is passed to 'BiocParallel::bplapply'. The default, 'BiocParallel::SerialParam()', uses the current R process. Pass an explicit backend to enable parallel execution

over motifs. You can specify BPPARAM = BiocParallel::SnowParam(8) on all operating systems, or BPPARAM = BiocParallel::MulticoreParam(8) on Unix-like systems to use, for example, 8 cores.

BPOPTIONS The BPOPTIONS used by bplapply. See BiocParallel package. This argument is passed to 'BiocParallel::bplapply'. The default is 'bpoptions()'. Some useful tasks: bpoptions(progressbar = TRUE, log = TRUE). progressbar = TRUE enables a progress bar that can be useful when processing many tasks. log = TRUE enables logging to debug each step of the parallel tasks.

Details

This function:

- scans promoter sequences using the 'Jmatrix' PWM.
- retains only those that meet the threshold criterion.
- scans the filtered sequences across the background PWMs.

Value

A 'PSMatrixList' object in which the foreground values (the alignment scores) have been computed for each sequence in 'prom_seq', that has passed the threshold, based on the position weight matrices in 'background'. If no sequences pass the filter, a warning is issued, and the function returns 'NULL'.

See Also

pscan, whose "Foreground size" section explains why a foreground that is a large share of the background weakens the statistic.

Examples

```
# Run on a small subset using bundled data (no downloads).
prom_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(prom_path)
set.seed(1)
n_prom <- min(100, length(prom_seq))
prom_seq <- prom_seq[sample(seq_along(prom_seq), n_prom)]

bg_path <- system.file("extdata",
    "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
bg <- ps_retrieve_bg_from_file(bg_path, J2020)

JM <- bg[[1]]
res <- pscan_filtered(prom_seq,
    JM,
    background = bg,
    BPPARAM = BiocParallel::SerialParam()
)
```

pscan_full_bg *Extract Pre-Computed Metrics from a PSMatrixList Object*

Description

This function is a faster alternative to pscan, that works only if a full background PSMatrixList object has already been computed. Instead of re-scanning input sequences, it retrieves pre-computed alignment scores, TFBS positions, strands, and binding oligos for a selected subset of sequences from the complete background.

Usage

```
pscan_full_bg(ID, full_pfms, scheme = "auto", quiet = FALSE)
```

Arguments

ID	A character vector containing sequence identifiers (transcript IDs) corresponding to the sequences of interest. These should match those used when the background was originally computed (e.g., RefSeq transcript IDs).
full_pfms	A complete background object. Create it with the background builder in full-background mode. It stores scores and metadata for the complete promoter reference set.
scheme	Transcript identifier scheme. ‘"auto"‘ (the default) detects it from the background’s transcript identifiers. The choices match those used for promoter selection.
quiet	Logical. Suppress the scheme-detection message.

Details

This is particularly useful when analyzing different subsets of sequences on the same background, reducing computation time.

Internally, the function matches the provided ‘ID‘ values against the ‘transcriptIDLegend‘ slot of the full background PSMatrixList. The transcriptIDLegend slot was introduced to handle cases where duplicate promoter sequences are removed using unique() during background generation. Since in this case only one representative sequence name is retained, the mapping vector ensures that if a user provides a transcript ID corresponding to a removed duplicate, it can still be correctly associated with the retained sequence used in the background.

Matching is scheme-aware. Under the RefSeq scheme a trailing version (‘".1"‘, ‘".2"‘) is a release of the same transcript and is normalised away, so ‘NM_000546.6‘ matches a background built from ‘NM_000546.4‘. Under the TAIR scheme the same punctuation denotes a splice variant, so ‘AT1G01110.2‘ is a different transcript from ‘AT1G01110.1‘ and the suffix is preserved. Requesting an identifier that is ambiguous within the background raises a warning naming the identifier rather than silently returning the first match.

A full background is a whole ‘PSMatrixList‘, hits included, so ‘saveRDS()‘ / ‘readRDS()‘ is the way to keep one between sessions. The ‘.txt‘ format ‘ps_write_bg_to_file()‘ produces carries only the background size, mean and standard deviation of each matrix, which is enough for ‘pscan()‘ and not enough for this function.

Identifiers absent from the background are dropped with a warning. That normally means the promoter was excluded during background construction (high ’N’ content or a length mismatch), but it also covers identifiers that were never part of this background.

Value

A ‘PSMatrixList‘ object in which the alignment scores and related metrics have been retrieved from ‘full_pfms‘ for each sequence in ‘ID‘, based on the background data.

See Also

pscan, whose "Foreground size" section explains why a foreground that is a large share of the background weakens the statistic.

Examples

```
# load a full background PSMatrixList
full_bg_path <- system.file("extdata", "full_pfms.rds", package = "PscanR")
full_pfms <- readRDS(full_bg_path)

IDs <- c(
    "NM_031921.6", "NM_001005484.2", "NR_047525.1", "NR_029639.1",
    "NR_036051.1", "NR_029834.1", "NM_001029885.2", "NR_148357.1",
    "NR_148960.1", "NM_001130413.4"
)

# The bundled background is a 36-promoter toy, so retrieving ten of them is
# a large enough share of it to trip the foreground-fraction warning. A real
# background is tens of thousands of promoters and would not.
old <- options(PscanR.foreground.max_fraction = Inf)

results <- pscan_full_bg(IDs, full_pfms)
ps_results_table(results)

options(old)
```

PSMatrix *An S4 Class for a Single Position-Specific Matrix*

Description

This constructor creates a ‘PSMatrix‘ object, which contains a single Position Frequency Matrix (PFM) and associated metadata related to promoter sequence motif hits, such as background and foreground averages, standard deviation, size, z-scores, and other related information.

Usage

```
PSMatrix(
  pfm,
  ps_bg_avg = as.numeric(NA),
  ps_fg_avg = as.numeric(NA),
  ps_bg_std_dev = as.numeric(NA),
  ps_bg_size = as.integer(NA),
  .PS_PSEUDOCOUNT = 0.01,
  ...
)
```

Arguments

pfm	An object of class 'PFMatrix' from the 'TFBSTools' package, typically a position frequency matrix (PFM) representing the motif for which all the organism's promoter regions will be scanned.
ps_bg_avg	Numeric. The background average value, default = 'NA'. It is the average binding score of promoter regions when scanned with a PWM.
ps_fg_avg	Numeric. The foreground average value, default = 'NA'. Represents the average binding score of a set of promoter regions of co-expressed or co-regulated genes when scanned with a PWM.
ps_bg_std_dev	Numeric. The background standard deviation of PWM scores, default = 'NA'.
ps_bg_size	Integer. The number of sequences in the background reference set, default = 'NA'.
.PS_PSEUDOCOUNT	Numeric. The pseudocount added to avoid division by zero. Default = '0.01'.
...	Reserved for compatibility. Supplying additional arguments is an error, which protects against silently misspelled argument names.

Details

When a 'PSMatrix' object is created or modified, it is automatically validated using an internal function. The validation ensures that the background and foreground statistics are properly formatted, and that the foreground and stored full-background hit vectors have consistent lengths. The class extends 'TFBSTools::PFMatrix' and therefore also contains its motif identifier, name, matrix, strand, tags, and related metadata.

Value

A 'PSMatrix' object containing the provided statistics, initialized with default or provided values.

A 'PSMatrix' object created from the input 'PFMatrix' object.

Slots

ps_bg_avg Mean normalized best-hit score in the background reference set.

ps_fg_avg Mean normalized best-hit score in the retained foreground.

ps_bg_std_dev Standard deviation of normalized background scores.

ps_bg_size Number of sequences in the background reference set.

ps_fg_size Number of retained foreground sequences.

ps_hits_pos Integer positions of foreground best hits.

ps_hits_pos_bg Integer positions of stored full-background best hits.

ps_hits_strand Foreground best-hit strands.

ps_hits_strand_bg Stored full-background best-hit strands.

ps_hits_score Normalized foreground best-hit scores.

ps_hits_score_bg Stored normalized full-background best-hit scores.

ps_hits_oligo Foreground sequence windows at best-hit positions.

ps_hits_oligo_bg Stored full-background sequence windows.

ps_zscore Motif-enrichment Z-statistic.

`ps_pvalue` Upper-tail p-value for ‘ps_zscore‘.

`ps_seq_names` Identifiers of retained foreground sequences.

`ps_bg_seq_names` Identifiers of stored full-background sequences.

`.PS_PSEUDOCOUNT` Pseudocount used to convert frequencies to weights.

`.PS_ALPHABET` Integer encoding of A, C, G, and T.

Examples

```
# Load JASPAR motif matrices for vertebrates
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)

result <- PSMatrix(
    pfm = J2020[[1]],
    ps_bg_avg = 0.25,
    ps_fg_avg = 0.5,
    ps_bg_std_dev = 0.05,
    ps_bg_size = 1000L # number of background sequences
)
print(result)
```

`PSMatrixList` *An S4 Class for Storing Position-Specific Matrices*

Description

This constructor creates a ‘PSMatrixList‘ object, which is a container for managing multiple ‘PSMatrix‘ objects. It is similar to ‘PFMatrixList‘ but extends its functionality.

This method converts a ‘PFMatrixList‘ object into a ‘PSMatrixList‘ object by applying the conversion for each ‘PFMatrix‘ in the list to a ‘PSMatrix‘. The resulting ‘PSMatrixList‘ is created by combining the converted ‘PSMatrix‘ objects into a list and returning the new object.

Usage

```
PSMatrixList(..., transcriptIDLegend = character(), use.names = TRUE)
```

Arguments

`...`	Objects of class ‘PSMatrix‘ to include in the list.
`transcriptIDLegend`	Named character vector. The names correspond to the IDs in the supplied background reference set, whereas corresponding values are the IDs of transcripts retained by the unique() function when multiple identical sequences exist. So, if multiple identical sequences exist (e.g., ID1, ID2, ID3, and ID4), and unique() retains only ID2, the mapping will associate each original name with its unique counterpart (ID1 → ID2, ID2 → ID2, ID3 → ID2, ID4 → ID2). Defaults to an empty vector, which marks the list as an ordinary background rather than a full one. ‘ps_build_bg(fullBG = TRUE)‘ builds this vector from the names of the sequences it is given; pass it here to carry it through an object rebuilt by hand.
`use.names`	Logical. Assert whether to use names from the input objects. Default = ‘TRUE‘

Details

'PSMatrixList' extends 'TFBSTools::PFMatrixList' and adds the 'transcriptIDLegend' used by full-background retrieval.

Value

A 'PSMatrixList' object, which is a list containing 'PSMatrix' objects. Each element in the list corresponds to a 'PSMatrix' object provided as input.

A 'PSMatrixList' object, which is a list containing the 'PSMatrix' objects converted from the 'PFMatrixList'.

Slots

`transcriptIDLegend` Named character vector mapping original background identifiers to identifiers retained after sequence deduplication.

Examples

```
# Load JASPAR motif matrices for vertebrates
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)

PSM1 <- PSMatrix(
    pfm = J2020[[1]],
    ps_bg_avg = 0.25,
    ps_fg_avg = 0.5,
    ps_bg_std_dev = 0.05,
    ps_bg_size = 1000L # number of background sequences
)
PSM2 <- PSMatrix(
    pfm = J2020[[2]],
    ps_bg_avg = 0.25,
    ps_fg_avg = 0.5,
    ps_bg_std_dev = 0.05,
    ps_bg_size = 1000L # number of background sequences
)
result <- PSMatrixList(PSM1, PSM2)
ps_results_table(result)
```

---

ps_available_bg *Get Available Pre-Computed Background Files*

Description

This function retrieves the names of all available .txt files containing background information, that serves for the construction of the background PSMatrixList. If a keyword is specified, the function will return only the file names that contain that string. Note that the filtering parameter works only if you know the nomenclature used to name the file. See the details paragraph for further information.

Usage

```
ps_available_bg(keyword = NULL, details = FALSE)
```

Arguments

keyword	A string to filter the file names. Default is NULL, so the complete list of available file names is returned.
details	Logical. If 'FALSE' (the default), return the filename vector used by earlier PscanR versions. If 'TRUE', return matching rows from the background catalog, including versions, latest status, provenance, and checksums.

Details

Some information for the filtering:

- Type of collection: To filter by JASPAR version, search by year prefixed by 'J', e.g., 'J2020'.
- To filter by genome assembly, use assembly identifiers like 'mm10' or 'mm39'.
- To filter by promoter region, use strings like '500u_0d', where 'u' stands for 'upstream' and 'd' for 'downstream'
- You may also filter by background version, e.g., 'psbg2'.
- Multiple keywords can be combined (e.g., 'hs1.*450u_50d').

Value

A character vector containing available background filenames, or a catalog 'data.frame' when 'details = TRUE'.

See Also

ps_retrieve_bg, ps_retrieve_bg_from_file

Examples

```
# The catalog is bundled with PscanR, so listing needs no network access.
head(ps_available_bg())
ps_available_bg("mm10")
ps_available_bg("hs1.*450u_50d", details = TRUE)
```

---

ps_bg_avg *Retrieve the Background Average Score for Motif Enrichment Analysis (Generic Function)*

Description

'ps_bg_avg' retrieves the average background score stored in a 'PSMatrix'.

Usage

```
ps_bg_avg(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The background average score represents the mean motif score computed over the background sequences. This value is essential for normalizing motif enrichment results.

Value

- If ‘x‘ is a ‘PSMatrix‘ object (Pscan result): a **numeric value** representing the stored background average score.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_avg(pfm1)
```

---

ps_bg_avg,PSMatrix-method

Get Background Average Score

Description

Retrieves the background average score stored in a ‘PSMatrix‘ object. This score represents the average binding score of promoter regions when scanned with a PWM.

Usage

```
## S4 method for signature 'PSMatrix'
ps_bg_avg(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘, but this argument is currently unused as the function returns a single numeric value. It is included for consistency with other matrix-related methods.

Value

The stored numeric background average. It is the mean best-hit PWM score over the sequences in the background reference set.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_avg(pfm1)
```

ps_bg_avg<- *Generic Setter Method for Background Mean Value of an object*

Description

Generic Setter Method for Background Mean Value of an object

Usage

```
ps_bg_avg(x, ...) <- value
```

Arguments

x	An object for which the background average is to be set.
...	Additional arguments.
value	The value for the background average.

Value

An object of the same class of the input with the modified background average value.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_avg<-`(pfm1, value = 0.73473)
```

ps_bg_avg<-,PSMatrix-method *Set Background Average*

Description

This method specifically sets the background average for an object of class 'PSMatrix'.

Usage

```
## S4 replacement method for signature 'PSMatrix'
ps_bg_avg(x) <- value
```

Arguments

x	An object of class 'PSMatrix'.
value	Numeric. The mean background value to be set.

Details

The background average is the mean normalized best-hit score across the background reference set. This method replaces that stored value and then validates the object.

Value

The modified ‘PSMatrix‘ object with the updated background average.

See Also

ps_bg_avg to retrieve the background average value, and `methods::validObject()` for object validation.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_avg<-`(pfm1, value = 0.73473)
```

`ps_bg_seq_names` *Retrieve Sequence Names for the Background Dataset (Generic Function)*

Description

‘ps_bg_seq_names‘ is a **generic function** that extracts the sequence names or identifiers stored for a full-background reference set.

Usage

```
ps_bg_seq_names(x, ...)
```

Arguments

`x`	An object containing sequence information.
`...`	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes storing sequence data.

The ‘ps_bg_seq_names‘ function is particularly relevant for background datasets, which include stored motif scan results for every retained background sequence.

In a PSMatrixList object, the ‘ps_bg_seq_names‘ slot is populated only for full-background matrices. This allows ‘pscan_full_bg()‘ to retrieve selected hits without rescanning DNA.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **character vector** of sequence names corresponding to the analyzed promoter regions.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_bg_seq_names(full_pfm1)
```

`ps_bg_seq_names,PSMatrix-method`
Get Sequence Identifiers of the Background Dataset

Description

Retrieves the names or identifiers of the promoter sequences in a 'PSMatrix' object computed on a background reference set.

Usage

```
## S4 method for signature 'PSMatrix'
ps_bg_seq_names(x, withDimnames = TRUE)
```

Arguments

`x`	A 'PSMatrix' object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Default set to 'TRUE'.

Details

This method is specifically designed for background datasets. The method returns the stored background sequence identifiers.

Value

A character vector of names.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_bg_seq_names(full_pfm1)
```

ps_bg_size *Retrieve the Background Size Value for Motif Enrichment Analysis (Generic Function)*

Description

'ps_bg_size' is a **generic function** that retrieves the background size value stored in a 'PSMatrix'.

Usage

```
ps_bg_size(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The background size is the number of sequences in the background reference set used to calculate the stored statistics.

Value

- If 'x' is a 'PSMatrix' object (Pscan result), an **integer value**.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_size(pfm1)
```

ps_bg_size,PSMatrix-method *Get Background Size*

Description

Retrieves the number of sequences in the background reference set used to compute the background statistics in a 'PSMatrix' object.

Usage

```
## S4 method for signature 'PSMatrix'
ps_bg_size(x, withDimnames = TRUE)
```

Arguments

x	A 'PSMatrix' object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

An integer giving the number of background sequences used for background scoring.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_size(pfm1)
```

---

ps_bg_size<- *Set Background Size (Generic Function)*

Description

'ps_bg_size<-' is a **generic function** that assigns the background size for a given object. The background size typically represents the number of background sequences used for motif enrichment analysis.

Usage

```
ps_bg_size(x, ...) <- value
```

Arguments

x	An object to which the background size will be assigned, typically a 'PSMatrix' object.
...	Additional arguments (not used in the generic function but may be used in specific methods).
value	A numeric value representing the background size (e.g., number of background sequences).

Details

Methods should be implemented for specific object classes that support background size assignment.

Value

The updated object with the assigned background size.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_size<-`(pfm1, value = 25629L)
```

ps_bg_size<-,PSMatrix-method *Set Background Size in a PSMatrix*

Description

This method specifically sets the background size for a 'PSMatrix' object. The background size represents the number of promoter regions used in the background model.

Usage

```
## S4 replacement method for signature 'PSMatrix'
ps_bg_size(x) <- value
```

Arguments

x	A 'PSMatrix' object.
value	An integer representing the new background size.

Value

The modified 'PSMatrix' object with the updated background size.

See Also

ps_bg_size to retrieve the background size, and methods::validObject() for object validation.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_size<-`(pfm1, value = 25629L)
```

ps_bg_std_dev *Retrieve the Background Standard Deviation for Motif Enrichment Analysis (Generic Function)*

Description

'ps_bg_std_dev' retrieves the background standard deviation stored in a 'PSMatrix'.

Usage

```
ps_bg_std_dev(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The background standard deviation value quantifies the variability in the background motif binding scores.

Value

- If ‘x‘ is a ‘PSMatrix‘ object (Pscan result): a **numeric value** representing the stored background standard deviation value.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_std_dev(pfm1)
```

`ps_bg_std_dev,PSMatrix-method`

Get Background Standard Deviation

Description

Retrieves the background standard deviation of PWM scores stored in a ‘PSMatrix‘ object. This value quantifies the variability in the background motif binding scores.

Usage

```
## S4 method for signature 'PSMatrix'
ps_bg_std_dev(x, withDimnames = TRUE)
```

Arguments

`x`	A ‘PSMatrix‘ object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

A numeric value representing the standard deviation of the background PWM scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_bg_std_dev(pfm1)
```

ps_bg_std_dev<- *Set Background Standard Deviation (Generic Function)*

Description

'ps_bg_std_dev<-' is a **generic function** that sets the background standard deviation value for a given object.

Usage

```
ps_bg_std_dev(x, ...) <- value
```

Arguments

x	An object to which the background standard deviation will be assigned, typically a 'PSMatrix' object.
...	Additional arguments (not used in the generic function but may be used in specific methods).
value	A numeric value representing the background standard deviation.

Details

Methods should be implemented for specific object classes that support background standard deviation assignment.

Value

The updated object with the assigned background standard deviation.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_std_dev<-`(pfm1, value = 0.08568925)
```

ps_bg_std_dev<-,PSMatrix-method *Set Background Standard Deviation*

Description

This method updates the background standard deviation stored in a 'PSMatrix' object.

Usage

```
## S4 replacement method for signature 'PSMatrix'
ps_bg_std_dev(x) <- value
```

Arguments

x	A 'PSMatrix' object.
value	Numeric. The value representing the new background standard deviation.

Value

The modified 'PSMatrix' object with the updated background standard deviation.

See Also

ps_bg_std_dev to retrieve the background standard deviation, and `methods::validObject()` for validation.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
`ps_bg_std_dev<-`(pfm1, value = 0.08568925)
```

---

ps_build_bg *Build background matrices in parallel*

Description

This function generates a background probability profile matrix list, used to assess if a transcription factor binding motif is appearing more often than by chance in the pscan() input sequences, using the BiocParallel framework for parallel execution. It computes background scores for each motif matrix, collected, for example, from the JASPAR database, by scanning them against a set of regulatory sequences (e.g., gene promoters) using the 'ps_scan' function. Note that the promoter region analysed may vary.

Usage

```
ps_build_bg(
  x,
  pfms,
  BPPARAM = BiocParallel::SerialParam(),
  BPOPTIONS = bpoptions(),
  fullBG = FALSE
)
```

Arguments

x	A 'DNAStringSet' object (see Biostrings package) containing the regulatory sequences chosen as the background reference set (e.g., gene promoters retrieved from a 'TxDb' object). These sequences are the target for background scanning. Duplicated sequences are removed. All promoters should have the same length; if they do not, only those of the most common length are kept, with a warning. If the promoters kept are shorter than any motif, the function stops with an error that names those motifs.

pfms — A 'PFMatrixList' of transcription-factor binding motifs, for example matrices obtained from JASPAR.

BPPARAM — Parallelization parameter passed to 'bplapply' function from the 'BiocParallel' package. This parameter defines the parallel processing settings, including the number of cores or workers that the function uses for computation. The default, 'BiocParallel::SerialParam()', uses one R process. Pass a backend explicitly to enable parallel execution. See 'BiocParallel' package for more details.

BPOPTIONS — Optional configuration settings passed to 'bplapply' function from the 'BiocParallel' package. This can include additional options to control the behavior of parallel execution. See 'BiocParallel' documentation for more details.

fullBG — Logical. Default is FALSE. When set to TRUE, it creates a mapping between all input sequence names in the background reference set and the corresponding names retained after applying the unique() function. For example, if multiple identical sequences exist (e.g., ID1, ID2, ID3, and ID4), and unique() retains only ID2, the mapping will associate each original name with its unique counterpart (ID1 → ID2, ID2 → ID2, ID3 → ID2, ID4 → ID2). This vector is stored in the transcriptIDLegend slot of the PSMatrixList and helps generate a complete background PSMatrixList, improving computational efficiency for the pscan_full_bg() function. In addition, it retrieves for each PSM in the PSMatrixList output all the background metrics relative to hits score, position, strand, and oligonucleotide sequence of the regulatory sequences scanned with the PWM.

Details

This function validates input types and removes duplicated sequences from 'x' to avoid redundant computations. It also removes sequences whose width differs from the maximum input width and sequences with more than 50 The motif matrices are background scored by the 'ps_scan' function in parallel.

The text exporter writes background size, mean, and standard deviation for each matrix. That is all 'pscan()' needs for enrichment testing. It does not write per-sequence hits or the transcript legend used by full-background retrieval. Importing that text therefore cannot recreate a full background unless those matrices already contain the stored scan.

To persist a full background, use 'saveRDS()' / 'readRDS()' (or 'save()' / 'load()'), which preserve the object whole — hits, positions, strands, oligonucleotides and legend included. Expect a large file: the object holds one hit per promoter per matrix, so it grows with the product of the two.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A 'PSMatrixList' object, containing each motif matrix from 'pfms', background-scored against the sequences in 'x'.

See Also

pscan_full_bg, ps_write_bg_to_file

Examples

```
# Load the example dataset for promoter sequences (hg38 assembly,
# -200 +50 bp in respect to the TSS).
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)
prom_seq <- prom_seq[1:3]

# Use two matrices from the bundled JASPAR2020 collection.
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)[1:2]

# Generate the background-scored motif matrices
bg_matrices <- ps_build_bg(prom_seq, J2020,
    BPPARAM = BiocParallel::SerialParam()
)
bg_matrices
bg_matrices[[1]]

# Example for full-background generation
full_bg_matrices <- ps_build_bg(prom_seq, J2020,
    BPPARAM = BiocParallel::SerialParam(),
    fullBG = TRUE
)

full_bg_matrices[[1]]
```

---

ps_build_bg_from_table

Apply background parameters to position frequency matrices

Description

Associates background parameters from a 'data.frame' with a list of position frequency matrices (PFMs) and returns the updated matrices as a 'PSMatrixList' object.

Usage

```
ps_build_bg_from_table(x, pfms)
```

Arguments

x	A 'data.frame' containing background parameters for each motif. The data frame should include the following columns: • 'BG_SIZE': Background set size. • 'BG_MEAN': Mean score of background hits. • 'BG_STDEV': Standard deviation of background hits score. The row names of 'x' should correspond to the identifiers of the motifs in 'pfms'.
pfms	A 'PFMatrixList' object containing position frequency matrices representing transcription factor binding preferences, obtained, for example, from the JASPAR database. The number of elements in 'pfms' should match the number of rows of 'x', and their identifiers should correspond.

Details

This function:

- Validates the input type. ‘x‘ must be a ‘data.frame‘.
- Converts each element of pfms into ‘PSMatrix‘ objects.
- A warning is issued if the number of elements in ‘pfms‘ doesn’t match the number of rows of ‘x‘.
- If ‘pfms‘ is a full background — one carrying the per-promoter scan — the function stops when ‘BG_SIZE‘ disagrees with the number of hits stored on a matrix. The two are equal by construction, so a disagreement means the table was computed on a different set of promoters from the one those hits describe, and applying it would leave the object with statistics and hits from two different universes.

Note: This function does not compute new background scores — it assigns existing background parameters to each motif, assuming they were previously computed.

This function uses example datasets located in the ‘extdata/‘ directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using ‘system.file()‘ as shown in the examples.

Value

A ‘PSMatrixList‘ object containing each motif from ‘pfms‘, enriched with background scoring parameters from ‘x‘.

Examples

```
# create the `data.frame`
background_data <- data.frame(
    BG_SIZE = c(500, 450, 480),
    BG_MEAN = c(0.3, 0.25, 0.35),
    BG_STDEV = c(0.05, 0.07, 0.06)
)

# Retrieve motif matrices for vertebrates from JASPAR2020
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
j2020_subset <- J2020[1:3] # match the number of rows in `backgound_data`

rownames(background_data) <- c("MA0004.1", "MA0006.1", "MA0019.1")

# Generate background-scored motif matrices
bg_matrices <- ps_build_bg_from_table(background_data, j2020_subset)
bg_matrices
bg_matrices[[1]]
```

`ps_density_plot` *Density Plot of Motif Hits Along Promoter Regions*

Description

This function creates a density plot representing the distribution of hits along the promoter sequences based on their position and score for a specific PWM. It helps visualize where motif occurrences are concentrated.

Usage

```
ps_density_plot(
  pfm,
  shift = 0,
  st = ps_bg_avg(pfm),
  window = NULL,
  bins = NULL
)
```

Arguments

pfm	A 'PSMatrix' object. It is a selected matrix from the 'PSMatrixList', result of 'pscan' function.
shift	Integer coordinate assigned to the first promoter base. For a promoter beginning 200 bases upstream of the TSS, use '-200'.
st	Score threshold used to filter hits. Can be a numeric value to set the threshold directly, or a character: • 'all': the threshold is set to '0' (All the hits are evaluated). • 'loose': uses the background average score as threshold. • 'strict': uses the background average score together with the background standard deviation as threshold. Default is 'loose' (background average).
window	Optional length-2 numeric giving the promoter window, on the same scale as 'shift' – for promoters extracted 1000 bp upstream of the TSS and plotted with 'shift = -1000', that is 'c(-1000, 0)'. Supplying it corrects the estimate at the edges by reflection; leaving it 'NULL' only clips the curve to the observed range. The reflecting bounds are the positions a hit can be *reported* at, which stop 'ncol(pfm) - 1' short of the end of the window, since a hit is recorded at the start of the motif.
bins	Optional number of bins for a binned profile of the same positions, drawn behind the curve. Bars are on the density scale, so they share the curve's y axis directly. Carrying no bandwidth and no boundary assumption, they show what the smoothing is doing.

Details

The function filters motif hits based on a specified threshold and generates a density plot to show their distribution. The function includes a vertical dashed line marking the estimated kernel-density mode. This smoothed mode need not be an observed position or the most frequent discrete coordinate. At least two finite positions must pass the threshold. The title reports how many promoters passed.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A 'ggplot' object showing the density of hit positions along the promoter region. Nothing is drawn until the object is printed, so it can be modified with further 'ggplot2' layers first.

Examples

```
matrix_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
matrix <- readRDS(matrix_path)
ps_density_plot(matrix, shift = -200)

# The result is a ggplot object, so it can be restyled before drawing.
ps_density_plot(matrix, shift = -200) +
    ggplot2::labs(subtitle = "-200 to +50 around the TSS")
```

`ps_fg_avg` *Retrieve the Foreground Average Score for Motif Enrichment Analysis (Generic Function)*

Description

'ps_fg_avg' retrieves the average foreground score stored in a 'PSMatrix'.

Usage

```
ps_fg_avg(x, ...)
```

Arguments

`x`	An object containing motif scan results.
`...`	Additional arguments passed to specific methods.

Details

The foreground average score represents the mean motif score computed over the input (foreground) sequences. This value is crucial for comparing motif occurrences between the foreground and background models.

Value

- If 'x' is a 'PSMatrix' object (Pscan result): a **numeric value** representing the stored foreground average score.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_fg_avg(pfm1)
```

ps_fg_avg,PSMatrix-method
Get Foreground Average Score

Description

Retrieves the foreground average score stored in a ‘PSMatrix‘ object. This score represents the average binding score of a set of promoter regions of co-expressed or co-regulated genes when scanned with a PWM.

Usage

```
## S4 method for signature 'PSMatrix'
ps_fg_avg(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘, but this argument is currently unused as the function returns a single numeric value. It is included for consistency with other matrix-related methods.

Value

The stored mean of the non-missing foreground best-hit scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_fg_avg(pfm1)
```

ps_fg_size *Retrieve the Foreground Size Value for Motif Enrichment Analysis (Generic Function)*

Description

‘ps_fg_size‘ is a **generic function** that retrieves the foreground size value stored in a ‘PSMatrix‘.

Usage

```
ps_fg_size(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The foreground size is the number of unique sequences retained after input cleaning, including sequences whose best-hit score is ‘NA‘.

Value

- If ‘x‘ is a ‘PSMatrix‘ object (Pscan result), an **integer value**.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_fg_size(pfm1)
```

---

ps_fg_size,PSMatrix-method

Get Foreground Size

Description

Retrieves the number of unique foreground sequences retained after input cleaning. This includes retained sequences whose best-hit score is ‘NA‘.

Usage

```
## S4 method for signature 'PSMatrix'
ps_fg_size(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

An integer representing the retained foreground size.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_fg_size(pfm1)
```

ps_get_bg_table *Extract background statistics from a 'PSMatrixList' object*

Description

Extracts background statistics (size, mean, and standard deviation) from a 'PSMatrixList' of Position Weight Matrices representing transcription factor binding preferences, and returns a data frame containing these metrics.

Usage

```
ps_get_bg_table(pfms)
```

Arguments

pfms	A 'PSMatrixList' of Position Weight Matrices representing transcription factor binding preferences . Each element must be a 'PSMatrix' object or coercible to one.

Details

This function is useful for exporting background-scored matrices to a tabular format. The resulting 'data.frame' can be saved and later reapplied to PFMs using 'ps_build_bg_from_table()'.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A 'data.frame' with one row per matrix (corresponding to a transcription factor) in 'pfms'. The row names will match the matrix identifiers.

Columns:

- 'BG_SIZE': An integer vector containing the background set size for each PSM.
- 'BG_MEAN': A numeric vector containing the mean of the background hits score for each PSM.
- 'BG_STDEV': A numeric vector containing the standard deviation of the background hits score for each PSM.

See Also

ps_build_bg_from_table, ps_retrieve_bg_from_file

Examples

```
# Retrieve motif matrices for vertebrates from JASPAR2020
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
j2020_subset <- J2020[1:3] # match the number of rows in `backgound_data`

# create the `data.frame`
```

```
background_data <- data.frame(
    BG_SIZE = c(500, 450, 480),
    BG_MEAN = c(0.3, 0.25, 0.35),
    BG_STDEV = c(0.05, 0.07, 0.06)
)

rownames(background_data) <- c("MA0004.1", "MA0006.1", "MA0019.1")

# Generate background-scored motif matrices
bg_matrices <- ps_build_bg_from_table(background_data, j2020_subset)
bg_table <- ps_get_bg_table(bg_matrices)
bg_table
```

---

ps_hitpos_map *Heatmap of Motif Hit Positions from Pscan Results*

Description

This function creates a heatmap visualizing the positional distribution of motif hits based on a specified false discovery rate threshold. The heatmap shows each sequence's best-hit start for motifs that pass the motif-level FDR filter. The individual windows are not separately tested for significance.

Usage

```
ps_hitpos_map(pfms, FDR = 0.01, shift = 0, ...)
```

Arguments

pfms	A 'PSMatrixList' object containing multiple PWMs and associated metadata (foreground and background statistics). Typically the output of 'pscan()' or 'pscan_full_bg()' function.
FDR	Numeric. False Discovery Rate (FDR) threshold to select the TFs to be included in the analysis. The default is set to '0.01'.
shift	Integer coordinate assigned to the first sequence base. For a promoter beginning 200 bases upstream of the TSS, use '-200'.
...	Additional user-defined arguments that can be passed to the function (e.g., the color palette) to change the default settings.

Details

The function performs the following steps:

- Extracts the result table from the pscan algorithm result.
- Filters the result table based on the given FDR threshold.
- Creates a positional hits matrix.
- Generates the heatmap using the 'pheatmap' function, with customizable settings.

Default settings (which can be changed) are:

- 'cluster_rows' and 'cluster_cols', set to 'TRUE' by default.

- ‘color‘, which uses a white-yellow-red palette by default.
- The main ‘Pscan Hits Position Heatmap‘.

This function uses example datasets located in the ‘extdata/‘ directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using ‘system.file()‘ as shown in the examples.

Value

Invisibly returns the positional hits matrix used for plotting. The heatmap is drawn as a side effect.

Examples

```
# Load the promoter sequences for hg38 (Homo sapiens), promoter regions:
# -200 +50 bp in respect to the TSS.
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:10]

# Build a small background from bundled files.
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
background <- ps_retrieve_bg_from_file(bg_path, matrices)
motif_ids <- c(
    "MA0506.1", "MA0632.2", "MA0615.1",
    "MA0076.2", "MA0645.1", "MA0838.1"
)
background <- background[motif_ids]

# Execute the Pscan algorithm and view the result table
results <- pscan(prom_seq, background,
    BPPARAM = BiocParallel::SerialParam()
)

ps_hitpos_map(results, shift = -200)
```

---

ps_hits_oligo *Retrieve Matched Oligonucleotide Sequences (Generic Function)*

Description

‘ps_hits_oligo‘ is a **generic function** that extracts the oligonucleotide sequences corresponding to motif hits in a given object.

Usage

```
ps_hits_oligo(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **character vector**, where names correspond to sequence identifiers, and values represent the oligonucleotide sequences (subset of the input promoter sequences) matching the motif.

Strand

The sequence returned is always the **forward strand** of the promoter, as it appears in the input, whatever strand the motif matched on. Reading it against the matrix consensus is therefore misleading for a reverse-strand hit: pair it with ps_hits_strand and reverse-complement the oligo where that reports "-".

This is the behaviour of the original Pscan implementation and is kept for consistency with it. It is also the more useful convention for sequence context, since oligos from different motifs are then all in the same frame of reference and can be compared to each other and to the promoter.

See Also

ps_hits_strand, ps_hits_pos, ps_hits_table

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_oligo(pfm1)

# Oligos are forward-strand, so reverse-strand hits must be flipped before
# they are compared with the motif consensus.
oligos <- ps_hits_oligo(pfm1)
reverse <- ps_hits_strand(pfm1) == "-"
oligos[reverse] <- as.character(Biostrings::reverseComplement(
    Biostrings::DNAStringSet(oligos[reverse])
))
head(oligos)
```

---

ps_hits_oligo,PSMatrix-method

Get Matched Oligonucleotide Sequences

Description

Retrieves the oligonucleotide sequences that match the motif in the foreground sequences, returning a character vector.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_oligo(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object, typically the result of the ‘Pscan‘ algorithm.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Value

A character vector containing the sequences of motif matches (oligonucleotides) in the foreground set.

Strand

The sequence returned is always the **forward strand** of the promoter, whatever strand the motif matched on, following the original Pscan implementation. Use ps_hits_strand to find which hits matched on "-" and reverse-complement those before comparing them with the motif consensus.

See Also

ps_hits_strand, ps_hits_pos, ps_hits_table

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_oligo(pfm1)

# A reverse-strand hit reads as the motif only after flipping it.
table(ps_hits_strand(pfm1))
```

---

ps_hits_oligo_bg *Retrieve Matched Oligonucleotide Sequences from a Background Dataset (Generic Function)*

Description

‘ps_hits_oligo_bg‘ is a **generic function** that extracts the oligonucleotide sequences corresponding to motif hits among a background dataset in a given object.

Usage

```
ps_hits_oligo_bg(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

The ‘ps_hits_oligo_bg‘ function is particularly relevant for background datasets, which include stored motif scan results for every retained background sequence.

In a PSMatrixList object, the ‘ps_hits_oligo_bg‘ slot is populated only for full-background matrices. This allows ‘pscan_full_bg()‘ to retrieve selected hits without rescanning DNA.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **character vector**, where names correspond to sequence identifiers, and values represent the oligonucleotide sequences (subset of the promoter sequences) matching the motif.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_oligo_bg(full_pfm1)
```

---

ps_hits_oligo_bg,PSMatrix-method

Get Matched Oligonucleotide Sequences of a Background Dataset

---

Description

Retrieves the oligonucleotide sequences that match the motif in the background sequences, returning a character vector.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_oligo_bg(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object, typically the result of the ‘Pscan‘ algorithm.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Details

This method is specifically designed for background datasets, where motif hit oligonucleotides are precomputed for every retained background sequence. The returned values are named with their background sequence identifiers.

These stored hits allow pscan_full_bg to retrieve results for selected background identifiers without rescanning DNA.

Value

A character vector containing the sequences of motif matches (oligonucleotides) in the background set.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_oligo_bg(full_pfm1)
```

`ps_hits_pos` *Retrieve Motif Hit Positions (Generic Function)*

Description

'ps_hits_pos' is a **generic function** that retrieves the positions of motif hits in a given object. These positions indicate where motifs are located within promoter sequences. For the 'PSMatrix' method, the optional 'pos_shift' is the coordinate of the first sequence base; stored one-based positions are converted accordingly.

Usage

```
ps_hits_pos(x, ...)
```

Arguments

`x`	An object containing motif scan results.
`...`	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If 'x' is a 'PSMatrix' object: an **integer vector**, where names correspond to promoter sequence identifiers, and values represent motif hit positions (with an optional shift).

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_pos(pfm1)
```

ps_hits_pos,PSMatrix-method *Get Motif Hit Positions*

Description

Retrieves the positions of hits stored in a 'PSMatrix' object. These positions indicate where the motifs are located in each promoter sequence. Stored positions are one-based sequence indices. 'pos_shift' assigns the coordinate of the first sequence base. For a promoter from -200 to +50 relative to the TSS, 'pos_shift = -200' reports the first base as -200.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_pos(x, pos_shift = 0L, withDimnames = TRUE)
```

Arguments

x	A 'PSMatrix' object.
pos_shift	Integer coordinate assigned to the first sequence base. The default '0' reports positions from a zero-valued first base.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to 'TRUE'.

Value

A named integer vector where names correspond to promoter sequence identifiers, and values represent the adjusted motif hit positions.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_pos(pfm1)

ps_hits_pos(pfm1, pos_shift = -200) # in respect to the TSS
```

ps_hits_pos_bg *Retrieve Motif Hit Positions on Background Dataset (Generic Function)*

Description

'ps_hits_pos_bg' is a **generic function** that retrieves the positions of motif hits in a given object. These positions indicate where motifs are located within promoter sequences of a background dataset.

Usage

```
ps_hits_pos_bg(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

The ‘ps_hits_pos_bg‘ function is particularly relevant for background datasets, which include stored motif scan results for every retained background sequence.

In a PSMatrixList object, the ‘ps_hits_pos_bg‘ slot is populated only for full-background matrices. This allows ‘pscan_full_bg()‘ to retrieve selected hits without rescanning DNA.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: an **integer vector**, where names correspond to promoter sequence identifiers, and values represent motif hit positions along the background dataset.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_pos_bg(full_pfm1)
```

---

ps_hits_pos_bg,PSMatrix-method

Get Motif Hit Positions in a Background Dataset

Description

Retrieves the positions of hits stored in a ‘PSMatrix‘ object computed on a background reference set. These positions indicate where the motifs are located in each promoter sequence.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_pos_bg(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Details

This method is specifically designed for background datasets, where motif hit positions are precomputed for every retained background sequence. The function extracts the position values from the ps_hits_pos_bg slot and, if applicable, assigns sequence names using .ps_bg_seq_names().

These stored hits allow pscan_full_bg to retrieve results for selected background identifiers without rescanning DNA.

Value

A named integer vector where names correspond to promoter sequence identifiers, and values represent the motif hit positions.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_pos_bg(full_pfm1)
```

---

ps_hits_score *Retrieve Motif Hit Scores (Generic Function)*

Description

‘ps_hits_score‘ is a **generic function** that retrieves the motif hit scores for each promoter sequence in an object. These scores represent the binding affinity or enrichment level of promoter sequences when scanned with a Position Weight Matrix (PWM).

Usage

```
ps_hits_score(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **named numeric vector**, where names correspond to promoter sequence identifiers and values represent their respective motif hit scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_score(pfm1)
```

`ps_hits_score,PSMatrix-method` *Get Hits Score*

Description

Retrieves the motif hit scores for each promoter sequence in a 'PSMatrix' object. These scores represent the binding affinity or enrichment level of promoter sequences when scanned with a Position Weight Matrix (PWM).

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_score(x, withDimnames = TRUE)
```

Arguments

`x`	A 'PSMatrix' object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Default set to 'TRUE'.

Value

A named numeric vector where names correspond to promoter sequence identifiers and values represent their respective motif hit scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_score(pfm1)
```

`ps_hits_score_bg` *Retrieve Motif Hit Scores from a Background Dataset (Generic Function)*

Description

ps_hits_score_bg is a generic function designed to extract motif hit scores from a background dataset. These scores quantify the binding affinity or enrichment level of promoter sequences when scanned using a Position Weight Matrix (PWM). This function is specifically useful for analyzing background matrices, which serve as reference datasets in pscan analyses.

Usage

```
ps_hits_score_bg(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

The ‘ps_hits_score_bg‘ function is particularly relevant for background datasets, which include stored motif scan results for every retained background sequence. These scores provide a reference for comparison in motif enrichment analyses, helping to assess the significance of observed motif occurrences.

In a PSMatrixList object, the ‘ps_hits_score_bg‘ slot is populated only for full-background matrices. This allows ‘pscan_full_bg()‘ to retrieve selected hits without rescanning DNA.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **named numeric vector**, where names correspond to promoter sequence identifiers and values represent their respective motif hit scores.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_score_bg(full_pfm1)
```

---

ps_hits_score_bg,PSMatrix-method

Get Hits Score for Background Dataset

Description

Retrieves the motif hit scores for each promoter sequence in a ‘PSMatrix‘ object computed on a background reference set. These scores represent the binding affinity or enrichment level of promoter sequences when scanned with a Position Weight Matrix (PWM).

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_score_bg(x, withDimnames = TRUE)
```

Arguments

x	A 'PSMatrix' object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to 'TRUE'.

Details

This method is specifically designed for background datasets, where motif hit scores are precomputed for every retained background sequence. The function extracts the scores from the ps_hits_score_bg slot and, if applicable, assigns sequence names using .ps_bg_seq_names().

These stored hits allow pscan_full_bg to retrieve results for selected background identifiers without rescanning DNA.

Value

A named numeric vector where names correspond to promoter sequence identifiers and values represent their respective motif hit scores.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_score_bg(full_pfm1)
```

---

ps_hits_size *Retrieve the Hits Size Value for Motif Enrichment Analysis (Generic Function)*

Description

'ps_hits_size' is a **generic function** that retrieves the number of hits for motif enrichment analysis. Methods should be implemented for specific object classes that store scan results.

Usage

```
ps_hits_size(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The hits size is the number of retained per-sequence best windows. It does not apply a score or significance threshold.

Value

- If ‘x‘ is a ‘PSMatrix‘ object (Pscan result), an **integer value**.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_size(pfm1)
```

`ps_hits_size,PSMatrix-method`
Compute Hits Size

Description

Retrieves the number of retained best-scoring motif windows in a ‘PSMatrix‘ object. Pscan retains one best window per scanned sequence; this count does not apply a score or significance threshold.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_size(x, withDimnames = TRUE)
```

Arguments

`x`	A ‘PSMatrix‘ object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

An integer representing the number of retained per-sequence best hits.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_size(pfm1)
```

`ps_hits_strand` *Retrieve Motif Hit Strand Information (Generic Function)*

Description

'ps_hits_strand' is a **generic function** that retrieves the strand information ('+' or '-') of motif hits within an object. This indicates whether a motif was detected on the forward ('+') or reverse ('-') strand of a promoter sequence.

Usage

```
ps_hits_strand(x, ...)
```

Arguments

`x`	An object containing motif scan results.
`...`	Additional arguments passed to specific methods.

Details

The Pscan algorithm scans both strands of promoter sequences to ensure that no potential transcription factor binding sites are missed.

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If 'x' is a 'PSMatrix' object: a **character vector**, where names correspond to promoter sequence identifiers, and values represent the strand ('+' or '-') on which the motif was detected.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_strand(pfm1)
```

`ps_hits_strand,PSMatrix-method`

Get Motif Hit Strand Information

Description

Retrieves the strand information ('+' or '-') of motif hits in a 'PSMatrix' object. This indicates whether a motif was detected on the forward ('+') or reverse ('-') strand of the promoter sequence. The Pscan algorithm scans both the strands of the promoter sequences to ensure that no potential binding sites are missed.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_strand(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Value

A named character vector where names correspond to promoter sequence identifiers, and values represent the strand (‘+‘ or ‘-‘) on which the motif was detected.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_strand(pfm1)
```

---

ps_hits_strand_bg *Retrieve Motif Hit Strand Information for the Background Dataset (Generic Function)*

Description

‘ps_hits_strand_bg‘ is a **generic function** that retrieves the strand information (‘+‘ or ‘-‘) of motif hits within an object. This indicates whether a motif was detected on the forward (‘+‘) or reverse (‘-‘) strand of a promoter sequence of the background dataset.

Usage

```
ps_hits_strand_bg(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The Pscan algorithm scans both strands of promoter sequences to ensure that no potential transcription factor binding sites are missed.

Methods should be implemented for specific object classes that store motif scanning results.

The ‘ps_hits_strand_bg‘ function is particularly relevant for background datasets, which include stored motif scan results for every retained background sequence.

In a PSMatrixList object, the ‘ps_hits_strand_bg‘ slot is populated only for full-background matrices. This allows ‘pscan_full_bg()‘ to retrieve selected hits without rescanning DNA.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **character vector**, where names correspond to promoter sequence identifiers, and values represent the strand (‘+‘ or ‘-‘) on which the motif was detected.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_strand_bg(full_pfm1)
```

`ps_hits_strand_bg,PSMatrix-method`

Get Motif Hit Strand Information of a Background Dataset

Description

Retrieves the strand information (‘+‘ or ‘-‘) of motif hits in a ‘PSMatrix‘ object computed on a background reference set. This indicates whether a motif was detected on the forward (‘+‘) or reverse (‘-‘) strand of the promoter sequence. The Pscan algorithm scans both the strands of the promoter sequences to ensure that no potential binding sites are missed.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_strand_bg(x, withDimnames = TRUE)
```

Arguments

`x`	A ‘PSMatrix‘ object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Details

This method is specifically designed for background datasets, where motif hit strands are precomputed for every retained background sequence. The function extracts the strand values from the `ps_hits_strand_bg` slot and, if applicable, assigns sequence names using `.ps_bg_seq_names()`.

These stored hits allow `pscan_full_bg` to retrieve results for selected background identifiers without rescanning DNA.

Value

A named character vector where names correspond to promoter sequence identifiers, and values represent the strand (‘+‘ or ‘-‘) on which the motif was detected.

Examples

```
full_pfm1_path <- system.file("extdata",
    "full_pfm1.rds",
    package = "PscanR"
)
full_pfm1 <- readRDS(full_pfm1_path)
ps_hits_strand_bg(full_pfm1)
```

`ps_hits_table` *Retrieve a Summary Table of Motif Hits (Generic Function)*

Description

'ps_hits_table' is a **generic function** that extracts motif hit information from an object and organizes it into a structured table. The table includes motif hit scores, strand orientation, positions, and the matched oligonucleotide sequences.

Usage

```
ps_hits_table(x, ...)
```

Arguments

`x`	An object containing motif scan results.
`...`	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If 'x' is a 'PSMatrix' object: a **data frame** with the following columns:

  - 'SCORE': The motif hit score.
  - 'POS': The position of the motif hit.
  - 'STRAND': The strand orientation ('+' or '-').
  - 'OLIGO': The matched oligonucleotide sequence.

Rows correspond to sequence names and are **sorted by decreasing score**.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_table(pfm1)
```

`ps_hits_table,PSMatrix-method` *Get Table of Motif Hits*

Description

Creates a data frame summarizing the motif hits from a ‘PSMatrix‘ object, including the motif hit score, position, strand, and corresponding oligo sequence. The resulting table is sorted by motif score (descending). When the score is the same between two sequences, they are sorted by ascending position.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_table(x, pos_shift = 0L, withDimnames = TRUE)
```

Arguments

`x`	An object of class ‘PSMatrix‘. Should contain the following slots: • ‘ps_hits_score‘: a numeric vector of hit scores. • ‘ps_hits_strand‘: a character vector of strand information (‘-‘ and ‘+‘). • ‘ps_hits_oligo‘: a character vector of oligo sequences.
`pos_shift`	Integer coordinate assigned to the first sequence base. For a promoter beginning 200 bases upstream of the TSS, use ‘-200‘. The default ‘0‘ assigns coordinate zero to the first base.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Value

A ‘data.frame‘ ordered by decreasing score value, with the following columns:

- ‘SCORE‘: the motif hit score
- ‘POS‘: the position of the motif hit
- ‘STRAND‘: the strand orientation for each hit
- ‘OLIGO‘: the oligo sequence corresponding to each hit, stored as a character vector.

Row names correspond to sequence names when ‘withDimnames = TRUE‘.

Strand

‘OLIGO‘ is the forward strand of the promoter whatever ‘STRAND‘ says, as in the original Pscan. A row with ‘STRAND == "-"‘ matches the motif only after its oligo is reverse-complemented. See `ps_hits_oligo`.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_table(pfm1)
```

ps_hits_z *Compute Motif Hit Z-Scores (Generic Function)*

Description

‘ps_hits_z‘ is a **generic function** that computes Z-scores for motif hit scores in an object. The Z-score indicates how unusual a motif score is compared with scores in the supplied background reference set. Higher Z-scores suggest stronger motif enrichment, which may indicate regulatory significance.

Usage

```
ps_hits_z(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that store motif scanning results.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **named numeric vector**, where names correspond to promoter sequence identifiers and values represent their respective Z-scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_z(pfm1)
```

ps_hits_z,PSMatrix-method *Get Motif Hit Z-score*

Description

Computes the Z-Scores for motif hit scores in a ‘PSMatrix‘ object. The Z-score measures how unusual a motif score is compared to background sequences in the background reference set. Higher Z-scores mean stronger motif enrichment, suggesting potential regulatory significance.

Usage

```
## S4 method for signature 'PSMatrix'
ps_hits_z(x, withDimnames = TRUE)
```

Arguments

x	A ‘PSMatrix‘ object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to ‘TRUE‘.

Value

A named numeric vector where names correspond to promoter sequence identifiers and values represent their respective Z-scores.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_hits_z(pfm1)
```

---

ps_hit_score_plot *Hit Position Against Hit Score*

Description

Plots every scorable promoter's best hit as a point positioned by where the site sits and by how well it scores, so position and strength are read together. A positional profile alone cannot show whether a concentration is made of strong sites or weak ones.

Usage

```
ps_hit_score_plot(x, shift = 0, alpha = NULL, size = NULL)
```

Arguments

x	A ‘PSMatrix‘, or a ‘PSMatrixList‘ in which case the motifs are drawn as panels sharing both axes.
shift	Integer coordinate assigned to the first promoter base. For a promoter beginning 200 bases upstream of the TSS, use ‘-200‘.
alpha, size	Point opacity and point size. Left ‘NULL‘, both are scaled from the number of promoters so that a large scan stays readable; supply either to override that.

Details

Two horizontal references are drawn per motif: the background mean, and one standard deviation above it. They are the same thresholds ‘st‘ names elsewhere in the package, and they split the points into three bands, each drawn in its own colour: below the mean, between the mean and one standard deviation above it, and beyond that. A point in the top band is a site that ‘st = "strict"‘ keeps; the top two bands together are what ‘"loose"‘ keeps. Colouring the middle band separately rather than lumping it with the weak hits shows how much of a positional pattern rests on marginal sites.

Each promoter with a non-missing score contributes exactly one point; unscorable promoters are omitted. Vertical structure is the score distribution and horizontal structure is the positional one. A motif whose strong sites are positionally constrained shows points banking into one region above the upper line while the weak sites stay spread out.

Every scorable promoter is drawn, whatever the size of the scan: nothing is subsampled and nothing is binned. Opacity and point size are scaled from the number of promoters, interpolated between the values that suit a few hundred points and ones that survive tens of thousands, so that a large scan reads as density rather than as solid blocks of colour. ‘alpha‘ and ‘size‘ override that scaling.

Value

A ‘ggplot‘ object.

See Also

ps_density_plot, ps_hitpos_map

Examples

```
prom_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(prom_path)[1:10]
J2020 <- readRDS(system.file("extdata", "J2020.rds", package = "PscanR"))
bg_path <- system.file("extdata",
    "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
bg <- ps_retrieve_bg_from_file(bg_path, J2020)
bg <- bg[c("MA0506.1", "MA0632.2")]
results <- pscan(prom_seq, bg, BPPARAM = BiocParallel::SerialParam())

ps_hit_score_plot(results, shift = -200)
```

---

ps_load_select_transcripts

Load MANE Select and RefSeq Select Transcript Metadata

Description

Downloads and caches the human MANE Select and RefSeq Select transcript tables from UCSC hg38. The resulting table can be passed to ps_select_promoters to choose one representative RefSeq promoter per gene.

Usage

```
ps_load_select_transcripts(
  organism = "hg38",
  cache = TRUE,
  cache_dir = tools::R_user_dir("PscanR", "cache")
)
```

Arguments

organism	Character. Currently only "hg38" is supported.
cache	Logical. If TRUE, read/write a cached RDS file under cache_dir.
cache_dir	Directory used for cached transcript metadata.

Details

MANE Select is preferred over RefSeq Select. The returned table can be cached, inspected, and reused in multiple analyses.

The automatic metadata download provided by this function is human-specific and applies only to the RefSeq transcript scheme. It currently uses the UCSC hg38 MANE and RefSeq Select tracks, retrieved with `UCSC.utils::fetch_UCSC_track_data()`. For every other reference, use representative mode or supply species-specific selection metadata.

Value

A `data.frame` containing transcript identifiers, versionless accessions, gene symbols, selection sources, and selection priorities.

See Also

`ps_select_promoters`

Examples

```
cache_dir <- tempfile("pscanr-select-")
dir.create(cache_dir)
cache_file <- file.path(
    cache_dir,
    "hg38_mane_refseq_select_transcripts.rds"
)
example_tx <- data.frame(
    transcript_id = c("NM_000546.6", "NM_000492.4"),
    transcript_base = c("NM_000546", "NM_000492"),
    gene_symbol = c("TP53", "CFTR"),
    selection_source = c("MANE Select", "RefSeq Select"),
    selection_priority = c(1L, 2L)
)
saveRDS(example_tx, cache_file)

select_tx <- ps_load_select_transcripts(cache_dir = cache_dir)
head(select_tx)
unlink(cache_dir, recursive = TRUE)

if (interactive()) {
    current_select_tx <- ps_load_select_transcripts()
    head(current_select_tx)
}
```

---

ps_motif_barplot *Bar Plot of the Most Enriched Motifs*

Description

Plots the highest ranking motifs of a scan as horizontal bars, optionally coloured by a grouping such as the structural class of the factor.

Usage

```
ps_motif_barplot(
  pfms,
  n = 20,
  statistic = c("ZSCORE", "P.VALUE", "FDR", "FG_AVG", "BG_AVG"),
  group = NULL,
  FDR = NULL
)
```

Arguments

pfms	A scan result, or a table from ‘ps_results_table()‘. A table carries no motif metadata, so ‘group‘ must be supplied explicitly when one is used.
n	Number of motifs to show. If the collection holds fewer than ‘n‘ motifs, all of them are shown.
statistic	Column to rank and plot. One of ‘"ZSCORE"‘ (the default), ‘"P.VALUE"‘, ‘"FDR"‘, ‘"FG_AVG"‘ or ‘"BG_AVG"‘.
group	Optional grouping mapped to bar colour. ‘NULL‘ (the default) draws every bar in one colour. ‘"class"‘ derives the structural class with ps_motif_class; ‘"family"‘ uses each matrix’s JASPAR family tag, with missing values labelled ‘"Unclassified"‘. Alternatively supply a character or factor vector, either one entry per motif or named by matrix identifier.
FDR	Optional significance cutoff applied before ranking. ‘NULL‘, the default, keeps every motif.

Details

Ranking follows from ‘statistic‘ rather than being controlled separately. ‘"ZSCORE"‘, ‘"FG_AVG"‘ and ‘"BG_AVG"‘ rank from the largest value down. ‘"P.VALUE"‘ and ‘"FDR"‘ rank from the smallest up, and are plotted as $-\log_{10}$, since a bar chart of raw p-values spanning many orders of magnitude is unreadable.

Bars start at zero, which is meaningful for a z-score and for the $-\log_{10}$ transforms alike.

Up to eight groups are drawn from a fixed palette chosen so that adjacent colours remain distinguishable under the common forms of colour-vision deficiency; beyond eight the ‘ggplot2‘ default applies. Motif names are always shown on the axis, so the grouping is never carried by colour alone.

Value

A ‘ggplot‘ object. Nothing is drawn until the object is printed, so it can be modified with further ‘ggplot2‘ layers first.

See Also

ps_motif_class, ps_results_table

Examples

```
prom_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(prom_path)[1:10]
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
bg_path <- system.file("extdata",
```

```
    "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
bg <- ps_retrieve_bg_from_file(bg_path, J2020)
bg <- bg[c(
    "MA0506.1", "MA0632.2", "MA0611.1",
    "MA0685.1", "MA0698.1", "MA0699.1"
)]
results <- pscan(prom_seq, bg, BPPARAM = BiocParallel::SerialParam())

ps_motif_barplot(results, n = 6)

# Colour by the structural class of the factor.
ps_motif_barplot(results, n = 6, group = "class")

# Or by the JASPAR family tag.
ps_motif_barplot(results, n = 6, group = "family")
```

`ps_motif_class` *Structural Class of Each Motif*

Description

Returns the structural class of the transcription factor behind each motif, for example ‘"Basic leucine zipper factors (bZIP)"‘ or ‘"WRKY"‘.

Usage

```
ps_motif_class(pfms)
```

Arguments

`pfms`	A ‘PSMatrixList‘ object, or any ‘PFMatrixList‘.

Details

The class is read from the ‘matrixClass‘ slot with `TFBSTools::matrixClass()`. It is worth knowing where it lives, because ‘tags(x)$class‘ is ‘NULL‘ for every JASPAR matrix in every release: ‘TFBSTools‘ moves the class out of the tag list and into its own slot when the matrix is built. Since ‘PSMatrix‘ extends ‘PFMatrix‘, the slot survives scanning, so the class is available from a scan result without consulting JASPAR again.

A small number of JASPAR matrices carry two classes; only the first is reported, so the result is always one label per motif and is safe to use as a grouping variable. Surrounding whitespace is stripped, because JASPAR ships a handful of classes with a trailing space — in JASPAR 2020 the IRF3, IRF4 and IRF5 matrices carry ‘"Tryptophan cluster factors "‘, which would otherwise group and colour separately from the other 38 members of that class.

Class is preferred over the ‘family‘ tag for grouping. Family coverage is incomplete in some collections, and its vocabulary changes between JASPAR releases, so it does not give a stable grouping across analyses.

Value

A character vector of class labels, named by JASPAR matrix identifier. Motifs with no class in JASPAR are reported as '"Unclassified"'.

See Also

ps_motif_barplot

Examples

```
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
head(ps_motif_class(J2020))
```

---

ps_pvalue *Retrieve the Motif-Enrichment P-Value (Generic Function)*

Description

'ps_pvalue' retrieves the upper-tail p-value stored in a 'PSMatrix' after scanning.

Usage

```
ps_pvalue(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The p-value quantifies the statistical significance of motif enrichment in the input sequences compared to the background model. A lower p-value is stronger evidence against the background null model.

Value

- If 'x' is a 'PSMatrix' object (Pscan result): a **numeric value** representing the stored p-value.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_pvalue(pfm1)
```

ps_pvalue,PSMatrix-method *Get P-Value*

Description

Retrieves the P-Value stored in a 'PSMatrix' object. The P-value quantifies the statistical significance of motif enrichment in the input sequences in respect to the background.

Usage

```
## S4 method for signature 'PSMatrix'
ps_pvalue(x, withDimnames = TRUE)
```

Arguments

x	A 'PSMatrix' object, typically the result of the 'Pscan' algorithm.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

A numeric value representing the statistical significance of motif enrichment. Lower values suggest stronger motif enrichment in the foreground set.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_pvalue(pfm1)
```

ps_results_table *Create a Summary Table of PscanR Results*

Description

This function generates a table summarizing the statistical analysis of motif enrichment, stored in a 'PSMatrixList' object. It retrieves pre-computed background and foreground statistics (the alignment scores between Pscan input regulatory sequences and position weight matrices) for each PWM. Then it compares motif occurrences in the foreground (e.g., coexpressed/coregulated promoter sequences) with the supplied background reference set using Z-scores, p-values, and FDR correction. Returns a table ordered by decreasing 'ZSCORE' and increasing 'P.VALUE'.

Usage

```
ps_results_table(pfms, FDR = 1)
```

Arguments

pfms	A 'PSMatrixList' object containing multiple PWMs and associated metadata (foreground and background statistics). Typically the output of 'pscan()' or 'pscan_full_bg()' functions.
FDR	A numeric value indicating the maximum false discovery rate (FDR) allowed for filtering the result table. Only rows with FDR <= 'FDR' will be retained. Default is 1, so all the results are displayed.

Details

This summary table is useful for the identification of transcription factors that could be common regulators of a set of co-expressed or co-regulated genes.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A data.frame with matrices ordered by increasing P.VALUE and decreasing ZSCORE, prioritizing significant and strongly enriched motifs. It contains the following columns:

- "NAME": The motif or transcription-factor name. Matrix identifiers are stored as row names.
- "BG_AVG": The average background score of TFBSs for each PWM.
- "BG_STDEV": The standard deviation of the background scores of TFBSs for each PWM.
- "FG_AVG": The average foreground score (the TFBS scores found by scanning the chosen subset of regulatory regions) for each PWM.
- "ZSCORE": The one-sample Z-statistic '(FG_AVG - BG_AVG) / (BG_STDEV / sqrt(n))', where 'n' is the number of non-missing foreground scores.
- "P.VALUE": The p-value for each PWM.
- "FDR": The adjusted p-value, representing the False Discovery Rate, is calculated using the Benjamini-Hochberg correction.

See Also

pscan, whose "Foreground size" section explains why a foreground that is a large share of the background weakens the statistic.

Examples

```
# Load the promoter sequences for hg38 (Homo sapiens), promoter regions:
# -200 +50 bp in respect to the TSS.
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:10]

# Build a small background from bundled files.
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
background <- ps_retrieve_bg_from_file(bg_path, matrices)
```

```
motif_ids <- c(
    "MA0506.1", "MA0632.2", "MA0615.1",
    "MA0076.2", "MA0645.1", "MA0838.1"
)
background <- background[motif_ids]

# Execute the PScan algorithm and view the result table
results <- pscan(prom_seq, background,
    BPPARAM = BiocParallel::SerialParam()
)

ps_results_table(results, FDR = 0.1)
```

`ps_retrieve_bg` *Generate PSMatrixList from Background data and JASPAR Matrix*

Description

This function retrieves background data for a specified organism and promoter region, and then fetches a corresponding matrix set from a specified JASPAR version. It returns a 'PSMatrixList' object with background statistics.

Usage

```
ps_retrieve_bg(
  jaspar_release,
  org,
  prom_reg,
  assembly = character(),
  version = "latest",
  destfile = NULL,
  source = c("experimenthub", "zenodo")
)
```

Arguments

`jaspar_release`	A character string specifying the JASPAR database version. You can choose between 'JASPAR2020', 'JASPAR2022', or 'JASPAR2024' (non case sensitive).
`org`	Organism acronym: • 'hs': Homo sapiens • 'mm': Mus musculus • 'at': Arabidopsis thaliana • 'sc': Saccharomyces cerevisiae • 'dm': Drosophila melanogaster
`prom_reg`	A numeric vector of two integers as '[upstream, downstream]' representing the promoter region relative to the TSS. Allowed combinations: • 200 base pairs upstream and 50 downstream base pair from the TSS (c(-200, 50)) • 450 bp upstream and 50 downstream (c(-450, 50))

- 500 bp upstream and 0 downstream (c(-500, 0))
- 950 bp upstream and 50 downstream (c(-950, 50))
- 1000 bp upstream and 0 downstream (c(-1000, 0))

`assembly` A string representing the assembly version for Human or mouse. For ‘"hs"‘ you can choose between ‘"hg38"‘ or the latest ‘"hs1"‘. For ‘"mm"‘ you can specify ‘"mm10"‘ or ‘"mm39"‘. Required for ‘"hs"‘ and ‘"mm"‘; ignored for other organisms.

`version` A string indicating the desired immutable background version, or ‘"latest"‘ to retrieve the latest validated version in the background catalog. The default is ‘"latest"‘. Use an explicit positive integer such as ‘"2"‘ when an analysis must remain pinned to a specific background. Version 2 is currently the only version distributed.

`destfile` A string indicating the path where the downloaded background .txt file should be saved. This tab-separated file contains the matrix identifiers, background size, average, and standard deviation. See the file-reader function in See Also. By default the file is not saved.

`source` Background retrieval backend. The default, ‘"experimenthub"‘, retrieves the background from the PscanRBackgrounds ExperimentHub package and, if Hub retrieval is unavailable, falls back to Zenodo with a warning. Use ‘"zenodo"‘ to download the file directly from the Zenodo record of its JASPAR release (2020: doi:10.5281/zenodo.23183695; 2022: doi:10.5281/zenodo.23183713; 2024: doi:10.5281/zenodo.23183720). If that download fails, the file is taken from the immutable record that archives all backgrounds in a single ZIP (doi:10.5281/zenodo.21821764).

Details

Each precomputed background is a separate ExperimentHub resource of the PscanRBackgrounds package; the files are hosted on Zenodo. This function looks up the requested background in the catalog bundled with PscanR, retrieves it (downloads are cached), checks it against its recorded SHA-256 value, and combines it with the specified JASPAR matrix collection to create a ‘PSMatrixList‘ with background statistics.

Value

A ‘PSMatrixList‘ object with background-scored motif matrices.

See Also

ps_retrieve_bg_from_file

Examples

```
# The online helper downloads its matching background and motif collection.
if (interactive()) {
    bg_matrices <- ps_retrieve_bg(
        "Jaspar2020", "hs",
        c(-200, 50), "hg38"
    )
    bg_matrices[[4]]
}

# The equivalent bundled files provide a fast, offline example.
```

```
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
local_bg_matrices <- ps_retrieve_bg_from_file(bg_path, matrices)
local_bg_matrices[[4]]
```

`ps_retrieve_bg_from_file` *Import background-scored matrices from a file*

Description

Reconstructs a 'PSMatrixList' object containing background-scored motif matrices, using a previously computed background dataset stored in a file. The input file must have been generated by the 'ps_write_bg_to_file()' function from a background 'PSMatrixList' object.

Usage

```
ps_retrieve_bg_from_file(file, pfms)
```

Arguments

`file`	A character string with the path to the input file. This file contains the score distribution statistics for the frequency matrices computed on a background set of regulatory sequences. The file should be in tabular format, where the first column contains the frequency matrix identifiers (row names) and subsequent columns contain the background statistics (e.g., `BG_SIZE`, `BG_MEAN`, `BG_STDEV`). Background statistics are generated with the 'ps_build_bg' function and can be written to a file with 'ps_write_bg_to_file'.
`pfms`	A 'PFMatrixList' object containing position frequency matrices representing transcription factor binding preferences, obtained, for example, from the JASPAR database. The provided list must correspond to the same frequency matrix list used to build the background stored in 'file'.

Details

This function:

- Validates the input type: must be a character string.
- Reads the input file into a 'data.frame', with row names taken from the first column.
- Calls 'ps_build_bg_from_table' with the created 'data.frame' and 'pfms' as input.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Precomputed backgrounds for other organisms, promoter windows and JASPAR releases are distributed through ExperimentHub by the PscanRBackgrounds package; see `ps_available_bg` and `ps_retrieve_bg`. See the vignettes for further details on the backgrounds available.

Value

A ‘PSMatrixList‘ object, containing each motif matrix from ‘pfms‘, background-scored using the values provided in ‘file‘. See ‘ps_build_bg_from_table‘ for more details.

See Also

ps_build_bg, ps_write_bg_to_file, ps_build_bg_from_table

ps_build_bg_from_table

Examples

```
# Load a background information file
file_path <- system.file("extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)

# Load the example dataset for JASPAR2020 matrices collection for
# vertebrates.
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)

# Generate the background-scored motif matrices from file
bg_matrices <- ps_retrieve_bg_from_file(file_path, J2020)
bg_matrices
bg_matrices[[2]]
```

---

ps_scan *Perform Motif Scanning (Generic Function)*

Description

‘ps_scan‘ is a **generic function** that scans DNA sequences for motif occurrences using an object containing motif scoring data. The scan is typically performed on both the forward and reverse complement strands to detect all potential transcription factor binding sites.

Usage

```
ps_scan(x, ...)
```

Arguments

x	An object containing motif information, typically a ‘PSMatrix‘ object.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes that support motif scanning.

Value

- If ‘x‘ is a ‘PSMatrix‘ object: a **PSMatrix object** with updated motif hit information, including positions, strands, scores, and matched sequences.

Examples

```
matrix_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
matrix <- readRDS(matrix_path)
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:5]

scanned_result <- ps_scan(matrix, prom_seq)
scanned_result
```

---

ps_scan,PSMatrix-method

Perform a Scan of DNA Sequences Using a 'PSMatrix' Object

Description

This method performs a scan of DNA sequences using a 'PSMatrix' object to identify potential hits based on the matrix score. The scan is performed on both strands (the direct and reverse complement strands) of the sequences to ensure that no potential binding sites are missed.

Usage

```
## S4 method for signature 'PSMatrix'
ps_scan(x, seqs, BG = FALSE,
  use_full_bg = FALSE, fullBG = FALSE, encoded = NULL)
```

Arguments

x	A 'PSMatrix' object representing the motif and related statistics to be used for scanning DNA sequences.
seqs	A 'DNAStringSet' object representing the sequences to be scanned for motif occurrences.
BG	A logical value indicating whether to calculate background statistics. Default is set to 'FALSE'.
use_full_bg	A logical value (default is 'FALSE'). If 'TRUE', the method assumes that the PSMatrix represents a special "full-background" set. In this case, the 'seqs' parameter should be a vector of sequence names (instead of actual DNA sequences). The function handles this by matching sequence names against precomputed background data in the 'PSMatrix'.
fullBG	Logical, default is 'FALSE'. If 'TRUE' it computes a series of computation for the generation of a "full-background", a special case of background that retains all the background hits score, position, strand, and oligonucleotide sequence for each regulatory sequence scanned with a PWM.
encoded	Internal. An optional pre-encoded integer matrix of the sequences (one column per sequence) produced by the calling functions to avoid re-encoding the same sequences for every motif. Users should leave this as the default 'NULL'.

Details

This method is designed to handle different types of sequence scanning scenarios. It can handle both regular DNA sequences (using a 'DNAStringSet') and specialized background sequence sets (either using a background flag or a full-background flag).

If 'use_full_bg' is set to 'TRUE', the function assumes that 'seqs' contains sequence identifiers rather than the sequences themselves. In this case, the method matches the sequence names to those in the 'PSMatrix''s background hit data and retrieves the corresponding binding information (score, strand, position, oligo).

When 'use_full_bg' is set to 'FALSE', the function scans both the forward and reverse complement strands of the sequences to ensure all potential binding sites are detected. Optionally, the background statistics (background average and standard deviation) can be computed and used during scanning when 'BG' is set to 'TRUE'.

Each motif position is scored against every window of the sequence; the highest-scoring window on either strand is reported. Bases are matched case-insensitively, so soft-masked (lower-case) sequence is scored as its upper-case equivalent. Any window containing a character outside A/C/G/T (for instance 'N' or an IUPAC ambiguity code) is skipped. If a sequence is shorter than the motif, or contains no scorable window at all (e.g. a fully 'N'-masked sequence), its score is reported as 'NA' so that it does not affect the foreground and background averages.

Value

A 'PSMatrix' object, with updated information about the motif hits in the sequences. This includes the positions, strands, scores, and oligos (sequence motifs) where the hits occurred.

Examples

```
matrix_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
matrix <- readRDS(matrix_path)
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:5]

scanned_result <- ps_scan(matrix, prom_seq)
scanned_result
```

---

ps_selection_summary *Summarise a Promoter Selection*

Description

Reports how a mapping produced by `ps_select_promoters` was arrived at: which scheme was used, which rule selected each promoter, how many genes had a real choice to make, and how many were decided by the tie-break rather than by the ranking.

Usage

```
ps_selection_summary(mapping)
```

Arguments

`mapping`	A mapping `data.frame` returned by `ps_select_promoters`, or the list returned with `return = "both"`.

Value

A list with two tables:

- `overall`: counts of requested, mapped, and unmapped genes, ambiguous choices, tie-break decisions, and excluded proteins.
- `by_source`: one row per selection source.

See Also

`ps_select_promoters`

Examples

```
annotation <- data.frame(
    GENE = c("AT1G01110", "AT1G01110", "AT1G01160"),
    TX = c("AT1G01110.2", "AT1G01110.1", "AT1G01160.1")
)
mapping <- ps_select_promoters(
    c("AT1G01110", "AT1G01160"),
    promoter_ids = c("AT1G01110.2", "AT1G01110.1", "AT1G01160.1"),
    annotation = annotation,
    gene_col = "GENE", transcript_col = "TX",
    mode = "representative"
)
ps_selection_summary(mapping)
```

---

ps_select_promoters *Select One Representative Promoter per Gene*

Description

Maps genes to transcript promoters and selects one promoter per gene using a rule appropriate to the transcript identifier scheme in use. The scheme is detected automatically and reported, so the choice is always auditable.

Usage

```
ps_select_promoters(
  genes,
  promoter_sequences = NULL,
  promoter_ids = NULL,
  annotation = NULL,
  org_db = NULL,
  keytype = "SYMBOL",
  gene_col = keytype,
  transcript_col = "REFSEQ",
  select_transcripts = NULL,
  scheme = "auto",
  mode = c("select", "representative", "all_transcripts"),
  fallback = TRUE,
  return = c("mapping", "sequences", "both"),
  sequence_names = c("gene_transcript", "transcript"),
  quiet = FALSE
)
```

Arguments

genes	Character vector of gene symbols, or values in keytype if annotation is not supplied.
promoter_sequences	Optional DNAStringSet of promoter sequences, named by transcript identifier.
promoter_ids	Optional character vector of available promoter transcript IDs. Ignored when promoter_sequences is supplied.
annotation	Optional gene-to-transcript mapping table. If omitted, the selected organism database is queried.
org_db	Optional organism annotation package object. If omitted and annotation is omitted, org.Hs.eg.db::org.Hs.eg.db is used.
keytype	Key type used for genes when querying org_db. Defaults to "SYMBOL".
gene_col	Column in annotation containing gene identifiers.
transcript_col	Column in annotation containing transcript identifiers.
select_transcripts	Optional table from ps_load_select_transcripts. If omitted and mode = "select", the table is downloaded or loaded from cache.
scheme	Transcript identifier scheme. "auto" detects and reports the scheme. The explicit choices are "refseq", "tair", "sgd", and "generic". See Details.
mode	Character. "select" uses MANE Select, then RefSeq Select, then the scheme's own ranking; it requires the RefSeq scheme. "representative" uses only the scheme's own ranking. "all_transcripts" returns every mapped transcript instead of one per gene.
fallback	Logical. If TRUE, genes without MANE/RefSeq Select transcripts fall back to the scheme's own ranking.
return	Character. "mapping" returns a mapping table, "sequences" returns selected promoter sequences, and "both" returns a list with both.
sequence_names	Character. How returned sequences should be named: "gene_transcript" uses GENE|TRANSCRIPT; "transcript" uses the selected promoter identifier alone.
quiet	Logical. Suppress the scheme-detection message.

Details

Transcript identifiers do not all mean the same thing, and the difference matters. In RefSeq, NM_000546.6 carries a release *version*: two versions name the same transcript, so the suffix is normalised away before matching. In TAIR, AT1G01110.2 carries a *splice variant*: the suffix is part of the transcript's identity and is never stripped, because variants of the same gene frequently have different transcription start sites. SGD systematic names such as YAL068W-A carry no suffix at all.

Detected schemes and their ranking rules:

"refseq" Human, mouse and Drosophila backgrounds. Prefers NM_, then NR_, then XM_, then XR_, then anything else. Protein accessions (NP_, XP_) are not transcripts and are excluded.

"tair" Arabidopsis. Prefers the lowest splice-variant number, which is the primary gene model.

"sgd" Saccharomyces cerevisiae, which has one transcript per ORF.

"generic" Anything unrecognised. Identifiers are matched whole, nothing is stripped, and all transcripts rank equally. This is the safe fallback and is chosen with a warning.

Ties are broken by a natural ordering of the promoter identifier, in which embedded numbers compare numerically, so `.10` follows `.9`. Because promoter identifiers are unique this is a total order, and the result therefore does not depend on the order in which promoters were supplied.

Automatic MANE Select and RefSeq Select metadata are available only for human hg38, so `mode = "select"` requires the RefSeq scheme and is an error otherwise. Elsewhere, use representative mode or provide a table containing a transcript identifier, selection source, and priority.

Value

A mapping table, a `DNAStringSet`, or a list containing both, as selected by `return`. The mapping has one row per promoter and records its gene, transcript, scheme, selection source and priority, candidate count, decision rule, and input order. Unmapped genes are retained as an attribute.

See Also

`ps_selection_summary`, `ps_load_select_transcripts`

Examples

```
annotation <- data.frame(
    SYMBOL = c("GENE1", "GENE1", "GENE2"),
    REFSEQ = c("NM_000001.1", "NM_000002.1", "NM_000003.1")
)
select_tx <- data.frame(
    transcript_id = "NM_000002.1",
    transcript_base = "NM_000002",
    gene_symbol = "GENE1",
    selection_source = "MANE Select",
    selection_priority = 1L
)
ps_select_promoters(
    c("GENE1", "GENE2"),
    annotation = annotation,
    select_transcripts = select_tx
)

# Arabidopsis splice variants are preserved, and the primary model wins.
at_annotation <- data.frame(
    GENE = c("AT1G01110", "AT1G01110"),
    TX = c("AT1G01110.2", "AT1G01110.1")
)
ps_select_promoters(
    "AT1G01110",
    promoter_ids = c("AT1G01110.2", "AT1G01110.1"),
    annotation = at_annotation,
    gene_col = "GENE", transcript_col = "TX",
    mode = "representative"
)
```

`ps_seq_names` *Retrieve Sequence Names (Generic Function)*

Description

'ps_seq_names' is a **generic function** that extracts the sequence names or identifiers from an object containing promoter sequence data.

Usage

```
ps_seq_names(x, ...)
```

Arguments

x	An object containing sequence information.
...	Additional arguments passed to specific methods.

Details

Methods should be implemented for specific object classes storing sequence data.

Value

- If 'x' is a 'PSMatrix' object: a **character vector** of sequence names corresponding to the analyzed promoter regions.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_seq_names(pfm1)
```

---

ps_seq_names,PSMatrix-method

Get Sequence Names

Description

Retrieves the names or identifiers of the promoter sequences in a 'PSMatrix' object. These names correspond to the promoter regions that were scanned by the Pscan algorithm.

Usage

```
## S4 method for signature 'PSMatrix'
ps_seq_names(x, withDimnames = TRUE)
```

Arguments

x	A 'PSMatrix' object.
withDimnames	Logical, whether to include dimension names in the output, if they exist in the object. Default set to 'TRUE'.

Value

A character vector of names.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_seq_names(pfm1)
```

---

ps_transcript_legend *Transcript ID Legend (Generic Function)*

Description

'ps_transcript_legend' is a **generic function** that retrieves the 'transcriptIDLegend' slot from a 'PSMatrixList' object. This slot contains a character vector mapping transcript IDs used in the dataset.

Usage

```
ps_transcript_legend(x, ...)
```

Arguments

x	A 'PSMatrixList' object.
...	Additional arguments (not used in this method but included for extensibility).

Value

A character vector containing transcript ID mappings.

Examples

```
full_pfms_path <- system.file("extdata", "full_pfms.rds", package = "PscanR")
full_pfms <- readRDS(full_pfms_path)
head(ps_transcript_legend(full_pfms))
```

---

ps_transcript_legend,PSMatrixList-method *Get the Transcript ID Legend from a PSMatrixList object*

Description

This method retrieves the value stored in the 'transcriptIDLegend' slot of a 'PSMatrixList' object.

Usage

```
## S4 method for signature 'PSMatrixList'
ps_transcript_legend(x)
```

Arguments

x	An object of class 'PSMatrixList'.

Value

A named character vector containing transcript ID mappings where values are the sequence identifiers maintained during the generation of the background by the unique() function, while the names are the original sequence identifiers for the organism of study.

Examples

```
full_pfms_path <- system.file("extdata",
    "full_pfms.rds",
    package = "PscanR"
)
full_pfms <- readRDS(full_pfms_path)
head(ps_transcript_legend(full_pfms))
```

---

ps_write_bg_to_file *Save background statistics from a 'PSMatrixList' object to a file*

Description

Saves background statistics (such as size, mean, and standard deviation) for each Position Weight Matrix in a 'PSMatrixList' object to a specified file.

Usage

```
ps_write_bg_to_file(pfms, file)
```

Arguments

pfms	A 'PSMatrixList' object of Position Weight Matrices representing transcription factor binding preferences, obtained, for example, from the JASPAR database. Each matrix should already include background statistics computed from the chosen background reference set. Each element must be a 'PSMatrix' object or coercible to one.
file	A character string specifying the path to the output file where the background statistics should be saved.

Details

This function first extracts background statistics from the input 'pfms' using 'ps_get_bg_table()'. It then writes the result to the specified file in a tab-delimited format. The output file begins with a header line: '[SHORT TFBS MATRIX]'. This header is used internally for compatibility with 'ps_retrieve_bg_from_file()', which expects it when reading.

The output does not include column names; the first column contains matrix identifiers, followed by 'BG_SIZE', 'BG_MEAN', and 'BG_STDEV'.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

'NULL' (invisibly). This function is called for its side effect of writing the given background statistics to the specified file in a tab-delimited format.

See Also

ps_get_bg_table

Examples

```
# Load the example dataset for JASPAR2020 matrices collection
# for vertebrates.
j2020_path <- system.file("extdata", "J2020.rds", package = "PscanR")
J2020 <- readRDS(j2020_path)
# File path to save the result (temporary file)
file_path <- tempfile(fileext = ".txt")

PSM1 <- PSMatrix(
    pfm = J2020[[1]],
    ps_bg_avg = 0.25,
    ps_fg_avg = 0.5,
    ps_bg_std_dev = 0.05,
    ps_bg_size = 1000L # number of background sequences
)
PSM2 <- PSMatrix(
    pfm = J2020[[2]],
    ps_bg_avg = 0.25,
    ps_fg_avg = 0.5,
    ps_bg_std_dev = 0.05,
    ps_bg_size = 1000L # number of background sequences
)

PSMatrixList_J2020 <- PSMatrixList(PSM1, PSM2)

ps_write_bg_to_file(PSMatrixList_J2020, file_path)
unlink(file_path)
```

---

ps_zscore *Retrieve the Motif-Enrichment Z-Score (Generic Function)*

Description

'ps_zscore' retrieves the Z-statistic stored in a 'PSMatrix' after scanning.

Usage

```
ps_zscore(x, ...)
```

Arguments

x	An object containing motif scan results.
...	Additional arguments passed to specific methods.

Details

The Z-score represents how significantly a motif is enriched in the input sequences compared to the background model.

Value

- If ‘x‘ is a ‘PSMatrix‘ object (Pscan result): a **numeric value** representing the stored Z-score.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_zscore(pfm1)
```

`ps_zscore,PSMatrix-method`
Get Z-Score

Description

Retrieves the Z-Score stored in a ‘PSMatrix‘ object.

Usage

```
## S4 method for signature 'PSMatrix'
ps_zscore(x, withDimnames = TRUE)
```

Arguments

`x`	A ‘PSMatrix‘ object.
`withDimnames`	Logical, whether to include dimension names in the output, if they exist in the object. Included for accessor consistency; it has no effect on this scalar value.

Value

A numeric value representing the ‘PSMatrix‘ Z-score (z-statistic) computed during motif scanning.

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
ps_zscore(pfm1)
```

`ps_zscore_heatmap` *Heatmap of Per-Sequence Motif Z-Scores*

Description

This function displays standardized best-hit scores for each retained sequence and motif selected by a motif-level FDR threshold.

Usage

```
ps_zscore_heatmap(pfms, FDR = 0.01, ...)
```

Arguments

`pfms`	A 'PSMatrixList' object containing multiple PWMs and associated metadata (foreground and background statistics). Typically the output of 'pscan()' or 'pscan_full_bg()' functions.
`FDR`	Numeric. False Discovery Rate (FDR) threshold to select the TFs to include in the analysis. The default is '0.01'.
`...`	Additional user-defined arguments to customize the heatmap settings, such as color palettes or clustering objects.

Details

The cells contain the per-sequence values returned by 'ps_hits_z()'; they are not correlations. Correlation distance is used only for clustering rows and columns when both dimensions contain at least two entries. The function performs the following steps:

- Extracts the result table and the z-score table from the pscan algorithm result.
- Filters the result table based on the given FDR threshold.
- Uses the filtered result table to create the z-table for the selected TFs.
- Generates the heatmap using the 'pheatmap' function, with customizable settings.

Default settings (which can be changed) are:

- 'cluster_rows' and 'cluster_cols', set to 'TRUE' when the selected matrix dimensions permit clustering.
- 'color', which uses a blue-white-red palette.
- The main 'Pscan Per-Sequence Z-Score Heatmap'.
- TF names are shown as column labels.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

Invisibly returns the Z-score submatrix used for plotting. The heatmap is drawn as a side effect.

Examples

```
# Load the promoter sequences for hg38 (Homo sapiens), promoter regions:
# -200 +50 bp in respect to the TSS.
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:10]

# Build a small background from bundled files.
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
background <- ps_retrieve_bg_from_file(bg_path, matrices)
motif_ids <- c(
    "MA0506.1", "MA0632.2", "MA0615.1",
    "MA0076.2", "MA0645.1", "MA0838.1"
)
background <- background[motif_ids]

# Execute the Pscan algorithm and view the result table
results <- pscan(prom_seq, background,
    BPPARAM = BiocParallel::SerialParam()
)

ps_zscore_heatmap(results, FDR = 0.05)
```

`ps_z_table` *Generate a Z-score Matrix of Motif Enrichment per Sequence*

Description

This function creates a matrix of Z-scores for motif occurrences across multiple PSMatrix contained in a 'PSMatrixList' object. Each row corresponds to a sequence, and each column represents a motif.

Usage

```
ps_z_table(pfms)
```

Arguments

`pfms`	A 'PSMatrixList' object containing multiple Position Weight Matrices and associated metadata (foreground and background statistics). This object is the output of 'pscan()' function.

Details

Z-score represents the statistical significance of the alignment scores for regulatory sequences relative to background expectation. A high Z-score suggests strong motif enrichment in the foreground compared to the background.

This function uses example datasets located in the 'extdata/' directory for demonstration purposes only. These files are not part of the core data used by the function. They can be accessed using 'system.file()' as shown in the examples.

Value

A matrix in which each column corresponds to a motif in the 'PSMatrixList' object, and each row to the sequence identifiers. The matrix contains z-score values for each sequence-motif pair, indicating how much the computed scores deviate from the expected background distribution. Higher Z-score values represent stronger motif enrichment.

Examples

```
# Load the promoter sequences for hg38 (Homo sapiens), promoter regions:
# -200 +50 bp in respect to the TSS.
file_path <- system.file("extdata", "prom_seq.rds", package = "PscanR")
prom_seq <- readRDS(file_path)[1:10]

# Build a small background from bundled files.
matrix_path <- system.file("extdata", "J2020.rds", package = "PscanR")
matrices <- readRDS(matrix_path)
bg_path <- system.file(
    "extdata", "J2020_hg38_200u_50d_UCSC.psbg.txt",
    package = "PscanR"
)
background <- ps_retrieve_bg_from_file(bg_path, matrices)
motif_ids <- c(
    "MA0506.1", "MA0632.2", "MA0615.1",
    "MA0076.2", "MA0645.1", "MA0838.1"
)
background <- background[motif_ids]

# Execute the Pscan algorithm and view the result table
results <- pscan(prom_seq, background,
    BPPARAM = BiocParallel::SerialParam()
)

ps_z_table(results)
```

---

show,PSMatrix-method *Display Details of a 'PSMatrix' object*

Description

This method displays a summary of a 'PSMatrix' object, including background and foreground statistics, standard deviation, background and foreground size, z-scores and P-Value.

Usage

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

Arguments

`object`	An object of class 'PSMatrix'.

Value

Prints a summary of the 'PSMatrix' object, including:

- Pscan Background Average
- Pscan Foreground (the sample) Average
- Pscan Background Standard Deviation
- Pscan Background Size
- Pscan Foreground (the sample) Size
- Pscan Z-score
- Pscan p-value (Z-test)

See Also

`ps_bg_avg`, `ps_fg_avg`, `ps_bg_std_dev`, `ps_bg_size`, `ps_fg_size`, `ps_zscore`, `ps_pvalue`

Examples

```
pfm1_path <- system.file("extdata", "pfm1.rds", package = "PscanR")
pfm1 <- readRDS(pfm1_path)
show(pfm1)
```

Index

∗ **internal**
PscanR, 6

ps_available_bg, 13, *62*
ps_bg_avg, 14, *17*, *77*
ps_bg_avg,PSMatrix-method, 15
ps_bg_avg<-, 16
ps_bg_avg<-,PSMatrix-method, 16
ps_bg_seq_names, 17
ps_bg_seq_names,PSMatrix-method, 18
ps_bg_size, 19, *21*, *77*
ps_bg_size,PSMatrix-method, 19
ps_bg_size<-, 20
ps_bg_size<-,PSMatrix-method, 21
ps_bg_std_dev, 21, *24*, *77*
ps_bg_std_dev,PSMatrix-method, 22
ps_bg_std_dev<-, 23
ps_bg_std_dev<-,PSMatrix-method, 23
ps_build_bg, *5*, *7*, 24, *63*
ps_build_bg_from_table, *7*, 26, *32*, *63*
ps_density_plot, 27, *53*
ps_fg_avg, 29, *77*
ps_fg_avg,PSMatrix-method, 30
ps_fg_size, 30, *77*
ps_fg_size,PSMatrix-method, 31
ps_get_bg_table, 32, *72*
ps_hit_score_plot, 52
ps_hitpos_map, 33, *53*
ps_hits_oligo, 34, *50*
ps_hits_oligo,PSMatrix-method, 35
ps_hits_oligo_bg, 36
ps_hits_oligo_bg,PSMatrix-method, 37
ps_hits_pos, *35*, *36*, 38
ps_hits_pos,PSMatrix-method, 39
ps_hits_pos_bg, 39
ps_hits_pos_bg,PSMatrix-method, 40
ps_hits_score, 41
ps_hits_score,PSMatrix-method, 42
ps_hits_score_bg, 42
ps_hits_score_bg,PSMatrix-method, 43
ps_hits_size, 44
ps_hits_size,PSMatrix-method, 45
ps_hits_strand, *35*, *36*, 46
ps_hits_strand,PSMatrix-method, 46
ps_hits_strand_bg, 47
ps_hits_strand_bg,PSMatrix-method, 48
ps_hits_table, *35*, *36*, 49
ps_hits_table,PSMatrix-method, 50
ps_hits_z, 51
ps_hits_z,PSMatrix-method, 51
ps_load_select_transcripts, 53, *67*, *68*
ps_motif_barplot, 54, *57*
ps_motif_class, *55*, 56
ps_pvalue, 57, *77*
ps_pvalue,PSMatrix-method, 58
ps_results_table, *6*, *55*, 58
ps_retrieve_bg, *6*, *7*, *14*, 60, *62*
ps_retrieve_bg_from_file, *7*, *14*, *32*, *61*, 62
ps_scan, 63
ps_scan,PSMatrix-method, 64
ps_select_promoters, *53*, *54*, *65*, *66*, 66
ps_selection_summary, 65, *68*
ps_seq_names, 68
ps_seq_names,PSMatrix-method, 69
ps_transcript_legend, 70
ps_transcript_legend,PSMatrixList-method, 70
ps_write_bg_to_file, *25*, *63*, 71
ps_z_table, 75
ps_zscore, 72, *77*
ps_zscore,PSMatrix-method, 73
ps_zscore_heatmap, 74
pscan, 3, *6*, *8–10*, *59*
pscan_filtered, 7
pscan_full_bg, 9, *25*, *37*, *41*, *44*, *48*
PscanR, 6
PscanR-package *(*PscanR*)*, 6
PSMatrix, 10
PSMatrixList, 12

setAs,PFMatrix,PSMatrix-method *(*PSMatrix*)*, 10
setAs,PFMatrixList,PSMatrixList-method *(*PSMatrixList*)*, 12
show,PSMatrix-method, 76