Package ‘pqsfinder’

November 21, 2024

Type Package

Title Identification of potential quadruplex forming sequences

Version 2.22.0

Date 2021-11-21

URL https://pqsfinder.fi.muni.cz

Author Jiri Hon, Dominika Labudova, Matej Lexa and Tomas Martinek

Maintainer Jiri Hon <`jiri.hon@gmail.com`>

Description Pqsfinder detects
DNA and RNA sequence patterns that are likely to fold into an
intramolecular G-quadruplex (G4). Unlike many other approaches,
pqsfinder is able to detect G4s
folded from imperfect G-runs containing bulges or mismatches
or G4s having long loops. Pqsfinder also assigns an integer score
to each hit that was fitted on G4 sequencing data and corresponds
to expected stability of the folded G4.

License BSD_2_clause + file LICENSE

biocViews MotifDiscovery, SequenceMatching, GeneRegulation

LazyData TRUE

Depends Biostrings

Imports Rcpp (>= 0.12.3), GenomicRanges, IRanges, S4Vectors, methods

Suggests BiocStyle, knitr, rmarkdown, Gviz, rtracklayer, ggplot2,
BSgenome.Hsapiens.UCSC.hg38, testthat, stringr, stringi

LinkingTo Rcpp, BH (>= 1.78.0)

SystemRequirements GNU make, C++11

VignetteBuilder knitr

RoxygenNote 7.1.2

Encoding UTF-8

NeedsCompilation yes

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

git_branch RELEASE_3_20

git_last_commit 7baec2f

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-11-20

Contents

`as.character,PQSViews-method`

Coerce to character vector

Description

Coerce to character vector

Usage

```
## S4 method for signature 'PQSViews'
as.character(x)
```

Arguments

`x`	PQSViews object.

Value

Character vector representing PQS.

`density,PQSViews-method`

Get density vector

Description

Desity vector represents numbers of PQS (potential quadruplex forming sequences) overlapping at each position in input sequence.

Usage

```
## S4 method for signature 'PQSViews'
density(x)
```

Arguments

`x`	PQSViews object.

Value

Density vector.

Examples

```
pqs <- pqsfinder(DNAString("CCCCCCGGGTGGGTGGGTGGGAAAA"))
density(pqs)
```

`maxScores` *Get vector of maximal scores*

Description

Get vector of maximal scores for a given object.

Usage

```
maxScores(x, ...)
```

Arguments

`x`	An object.
`...`	Additional arguments, for use in specific methods.

Value

Vector of maximal scores.

Examples

```
showMethods("maxScores")
```

`maxScores,PQSViews-method` *Get vector of maximal scores*

Description

For each sequence position it gives the maximal score of all PQS conformations which overlap that position.

Usage

```
## S4 method for signature 'PQSViews'
maxScores(x)
```

Arguments

`x`	PQSViews object.

Value

Vector of maximal scores.

Examples

```
pqs <- pqsfinder(DNAString("CCCCCCGGGTGGGTGGGTGGGAAAA"))
maxScores(pqs)
```

`pqsfinder` *Identify potential quadruplex forming sequences.*

Description

Function for identification of all potential intramolecular quadruplex patterns (PQS) in DNA or RNA sequence.

Usage

```
pqsfinder(
  subject,
  strand = "*",
  overlapping = FALSE,
  max_len = 50L,
  min_score = 47L,
  run_min_len = 2L,
  run_max_len = 11L,
  loop_min_len = 0L,
  loop_max_len = 30L,
  max_bulges = 3L,
  max_mismatches = 3L,
```

```
  max_defects = 3L,
  tetrad_bonus = 40L,
  mismatch_penalty = 28L,
  bulge_penalty = 20L,
  bulge_len_factor = 0.2,
  bulge_len_exponent = 1,
  loop_mean_factor = 6.6,
  loop_mean_exponent = 0.8,
  run_re = "G{1,10}.{0,9}G{1,10}",
  custom_scoring_fn = NULL,
  use_default_scoring = TRUE,
  deep = FALSE,
  verbose = FALSE
)
```

Arguments

`subject`	DNAString or RNAString object.
`strand`	Strand specification. Allowed values are "+", "-" or "*", where the last one represents both strands. Implicitly, the input DNAString object is assumed to encode the "+" strand.
`overlapping`	If true, than all overlapping PQS will be reported.
`max_len`	Maximal lenth of PQS.
`min_score`	Minimal PQS score. The default value 52 shows the best balanced accuracy on G4 sequencing data provided by Chambers et al. 2015.
`run_min_len`	Minimal length of quadruplex run.
`run_max_len`	Maximal length of quadruplex run.
`loop_min_len`	Minimal length of quadruplex loop. Unless the default scoring system is disabled, at most one loop can have zero length.
`loop_max_len`	Maxmimal length of quadruplex loop.
`max_bulges`	Maximal number of runs with bulge.
`max_mismatches`	Maximal number of runs with mismatch.
`max_defects`	Maximum number of defects in total (`max_bulges` + `max_mismatches`).
`tetrad_bonus`	Score bonus for one complete G tetrade.
`mismatch_penalty`	Penalization for a mismatch in tetrad.
`bulge_penalty`	Penalization for a bulge in quadruplex run.
`bulge_len_factor`	Penalization factor for a bulge length.
`bulge_len_exponent`	Exponent of bulge length.
`loop_mean_factor`	Penalization factor of loop length mean.
`loop_mean_exponent`	Exponent of loop length mean.
`run_re`	Regular expression specifying one run of quadruplex.

`custom_scoring_fn`	Custom quadruplex scoring function. It takes the following 10 arguments: `subject` - Input DNAString or RNAString object, `score` - implicit PQS score, `start` - PQS start position, `width` - PQS width, `loop_1` - start pos. of loop #1, `run_2` - start pos. of run #2, `loop_2` - start pos. of loop #2, `run_3` - start pos. of run #3, `loop_3` - start pos. of loop #3, `run_4` - start pos. of run #4. Return value of the function has to be new score represented as a single integer value. Please note that if `use_default_scoring` is enabled, the custom scoring function is evaluated AFTER the default scoring system but ONLY IF the default scoring system resulted in non-zero score (for performance reasons). On the other hand, when `use_default_scoring` is disabled, custom scoring function is evaluated on every PQS.
`use_default_scoring`	Enables default internal scoring system. This option is particularly useful in case you intend to radically change the default behavior and specify your own scoring function. By disabling the default scoring you will get a full control above the underlying detection algorithm.
`deep`	Perform deep search. With this option enabled, `maxScores` and `density` vectors are computed. Deep search is much more computationaly demanding.
`verbose`	Enables detailed output. Turn it on if you want to see all possible PQS found at each positions and not just the best one. It is highly recommended to use this option for debugging custom quadruplex scoring function. Each PQS is reported on separate row in the following format: `start cnt pqs_sequence score`, where `start` is the PQS starting position, `pqs_sequence` shows the PQS sequence structure with each run surrounded by square brackets and `score` is the score assigned to the particular PQS by all applied scoring functions.

Details

Use `elementMetadata` function to get extra PQS features like number of tetrads (nt), bulges (nb), mismatches (nm) or loop lengths (ll1, ll2, ll3).

Value

`PQSViews` object

Examples

```
pv <- pqsfinder(DNAString("CCCCCCGGGTGGGTGGGTGGGTAAAA"))
pv
elementMetadata(pv)
```

`PQSViews` *PQSViews class constructor*

Description

User friendly constructor for PQSViews class representing potential quadruplex forming sequences (PQS). PQSViews is a subclass of `XStringViews` class and adds two more slots to store PQS density and PQS score distribution.

Usage

```
PQSViews(
  subject,
  start,
  width,
  strand,
  score,
  density,
  max_scores,
  nt,
  nb,
  nm,
  rl1,
  rl2,
  rl3,
  ll1,
  ll2,
  ll3
)
```

Arguments

`subject`	DNAString or RNAString object.
`start`	Start positions.
`width`	Lengths.
`strand`	Strand specifications.
`score`	Scores.
`density`	Numbers of PQS overlapping at each position in `subject`.
`max_scores`	Score of the best PQS found at each position.
`nt`	Tetrad numbers.
`nb`	Bulge counts.
`nm`	Mismatch counts.
`rl1`	Run 1 lengths.
`rl2`	Run 2 lengths.
`rl3`	Run 3 lengths.
`ll1`	Loop 1 lengths.
`ll2`	Loop 2 lengths.
`ll3`	Loop 3 lengths.

Details

Use `elementMetadata` function to get extra PQS features like number of tetrads, bulges, mismatches or loop lengths.

Value

PQSViews object.

Examples

```
pv <- PQSViews(DNAString("GGTGGTGGTGG"), 1, 11, "+", 33, as.integer(rep(1, 11)),
               as.integer(rep(33, 11)), 2, 0, 0, 2, 2, 2, 1, 1, 1)
start(pv)
width(pv)
strand(pv)
score(pv)
density(pv)
maxScores(pv)
elementMetadata(pv)
```

`PQSViews-class` *An S4 class to represent potential quadruplex forming sequences*

Description

Represents potential quadruplex forming sequences found by `pqsfinder` function. This is a subclass of `XStringViews-class` class and adds one more slot.

Slots

`density` Numbers of PQS (potential quadruplex forming sequences) overlapping at each position in input sequence.

`max_scores` Score of the best PQS found at each position.

`score,PQSViews-method` *Get PQS score vector*

Description

Get PQS score vector

Usage

```
## S4 method for signature 'PQSViews'
score(x)
```

Arguments

`x`	PQSViews object.

Value

Score vector.

Examples

```
pqs <- pqsfinder(DNAString("CCCCCCGGGTGGGTGGGTGGGAAAA"))
score(pqs)
```

`show,PQSViews-method` *Show method*

Description

Show method

Usage

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

Arguments

`object`	PQSViews object.

Value

PQSViews object printed.

`strand,PQSViews-method` *Get PQS strand vector*

Description

Get PQS strand vector

Usage

```
## S4 method for signature 'PQSViews'
strand(x)
```

Arguments

`x`	PQSViews object.

Value

Strand vector.

Examples

```
pqs <- pqsfinder(DNAString("CCCCCCGGGTGGGTGGGTGGGAAAA"))
strand(pqs)
```

Index