Package 'G4SNVHunter'

December 20, 2024

Type Package

Title Evaluating SNV-Induced Disruption of G-Quadruplex Structures

Version 0.99.4

Description G-quadruplexes (G4s) are unique nucleic acid secondary structures predominantly found in guanine-rich regions and have been shown to be involved in various biological regulatory processes. G4SNVHunter is an R package designed to rapidly identify genomic sequences with G4-forming potential and accurately screen user-provided single nucleotide variants (also applicable to single nucleotide polymorphisms) that may destabilize these structures. This enables users to screen key variants for further experimental study, investigating how these variants may influence biological functions, such as gene regulation, by altering G4 formation.

Encoding UTF-8

LazyData false

Depends R (>= 4.3.0)

Imports Biostrings, stats, GenomicRanges, IRanges, Rcpp, RcppRoll, data.table, GenomeInfoDb, S4Vectors, ggplot2, cowplot, progress, ggseqlogo, viridis, ggpointdensity

LinkingTo Rcpp

RoxygenNote 7.3.2

License MIT + file LICENSE

Suggests knitr, BiocStyle, rmarkdown, BiocManager, BSgenome.Hsapiens.UCSC.hg19, DT, rtracklayer, testthat (>= 3.0.0), RBGL

VignetteBuilder knitr

biocViews Epigenetics, SNP

News NEWS.md

URL https://github.com/rongxinzh/G4SNVHunter

BugReports https://github.com/rongxinzh/G4SNVHunter/issues

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/G4SNVHunter

git_branch devel

git_last_commit 73edf5e

git_last_commit_date 2024-12-12

Repository Bioconductor 3.21

Date/Publication 2024-12-20

Author Rongxin Zhang [cre, aut] (ORCID:
<https://orcid.org/0000-0002-1643-1185>)

Maintainer Rongxin Zhang <rongxinzhang@outlook.com>

Contents

`checkSNV` *Check the validity of SNVs*

Description

This function checks whether the user-provided SNVs (or SNPs) are single nucleotide substitutions.

Usage

```
checkSNV(snv_gr = NULL, mode = "wra", ref_col = NULL, alt_col = NULL)
```

Arguments

`snv_gr`	A `GRanges` object containing SNV (or SNP) data.
`mode`	A character string specifying the checks to be performed. `w` for checking if all widths (w) are 1, `r` for checking if all ref (r) values are A, T, C, or G, `a` for checking if all alt (a) values are A, T, C, or G.
`ref_col`	Column name for the ref bases in `snv_gr`. Default is NULL.
`alt_col`	Column name for the alt bases in `snv_gr`. Default is NULL.

Value

A logical value indicating whether the user-provided SNVs (or SNPs) passed all checks.

Examples

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!requireNamespace("BSgenome.Hsapiens.UCSC.hg19", quietly = TRUE)) {
  BiocManager::install("GenomicRanges")
}

library(GenomicRanges)
gr1 <- GRanges("chr1", IRanges(start = 100, width = 1))
# check width ('w')
checkSNV(gr1, mode = "w")

gr2 <- GRanges(
  seqnames = Rle("seq1"),
  ranges = IRanges(c(100, 200, 300), width = 1),
  ref = c("A", "C", "G"),
  alt = c("T", "T", "A")
)

# check width ('w'), ref ('r'), and alt ('a')
checkSNV(gr2, mode = "wra", ref_col = "ref", alt_col = "alt")
# check width ('w') and alt ('a')
checkSNV(gr2, mode = "wa", alt_col = "alt")

gr3 <- GRanges("chr1", IRanges(start = 100, width = 10))
# widths should be all one
checkSNV(gr3, mode = "w")

gr4 <- GRanges(
  seqnames = Rle("seq1"),
  ranges = IRanges(start = 100, width = 1),
  ref = "AG",
  alt = "T"
)

# ref should be all one
checkSNV(gr4, mode = "wr", ref_col = "ref")
```

`filterSNVImpact` *Filter SNV Impact GRanges Object Based on User-Defined Thresholds*

Description

This function filters the SNV Impact GRanges object returned by the SNVImpactG4 function based on user-defined thresholds for the G4.info.score, mut.score, and score.diff parameters. This function filters SNVs that may significantly impact the formation of G4 structures using customizable filtering criteria. You are not required to specify all three threshold parameters. However, at least one threshold parameter must be provided.

Usage

```
filterSNVImpact(
  gr,
  raw_score_threshold = NULL,
  mut_score_threshold = NULL,
  score_diff_threshold = NULL
)
```

Arguments

gr	A GRanges object returned by the SNVImpactG4 function.
raw_score_threshold	A positive numeric value no greater than 4 used as the threshold for the absolute value of G4.info.score. G4s with an absolute G4Hunter score exceeding this threshold will be retained. If NULL, this threshold is not applied.
mut_score_threshold	A positive numeric value no greater than 4 used as the threshold for the absolute value of mut.score. Mutated G4s with an absolute G4Hunter score below this threshold will be retained. If NULL, this threshold is not applied.
score_diff_threshold	A negative numeric value no less than -4 used as the threshold for score.diff. G4s with a decrease in G4Hunter score greater than this threshold after variation will be retained. If NULL, this threshold is not applied.

Value

A filtered GRanges object, containing only the records that meet the specified threshold criteria.

See Also

SNVImpactG4 for assessing the impact of SNVs on G4 formation.

Examples

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!requireNamespace("GenomicRanges", quietly = TRUE)) {
  BiocManager::install("GenomicRanges")
}
```

```
if (!requireNamespace("BSgenome.Hsapiens.UCSC.hg19", quietly = TRUE)) {
  BiocManager::install("BSgenome.Hsapiens.UCSC.hg19")
}

library(GenomicRanges)
library(BSgenome.Hsapiens.UCSC.hg19)

hg19 <- BSgenome.Hsapiens.UCSC.hg19
chr21_seq <- DNAStringSet(hg19$chr21)
# Chromosome name is needed
names(chr21_seq) <- "chr21"

G4 <- G4HunterDetect(chr21_seq)

data(snp_gr)

res_snp <- SNVImpactG4(G4, snp_gr, alt_col = "alt")
filtered_snv_eff <- filterSNVImpact(res_snp,
  mut_score_threshold = 1.2,
  score_diff_threshold = -0.2
)
print(filtered_snv_eff)
```

`G4HunterDetect` *Detect G4 Sequences Using G4Hunter Algorithm*

Description

This function detects G4 sequences for a given `DNAStringSet` object using the G4Hunter algorithm.

Usage

```
G4HunterDetect(
  sequences = NULL,
  threshold = 1.5,
  window_size = 25,
  include_sequences = TRUE,
  strands = "b"
)
```

Arguments

`sequences`	A `DNAStringSet` object containing the sequences to be analyzed.
`threshold`	A numeric value specifying the threshold for the G4Hunter score (absolute value). Default is 1.5. We do not recommend setting the threshold below 1.2.
`window_size`	An integer specifying the window size (bp) for prediction. Default is 25. Another commonly used window size is 20. However, 25 is generally preferred unless otherwise required.

include_sequences
: A logical value (TRUE/FALSE) indicating whether to include the predicted G4 sequences in the output. Default is TRUE. Setting this parameter to FALSE can reduce the memory overhead of the final output, which may be useful for extremely large genomes. However, we strongly recommend retaining the sequence information in the output, as it is indispensable for subsequent analysis of the impact of variants on G4 formation.

strands
: A character string ("b" for both strands and "p" for positive strand only) indicating which strand to be considered. Default is "b".

Value

A GRanges object containing the predicted G4 sequences. The GRanges object includes the following metadata columns:

score The actual G4Hunter score for the G4 sequence detected (after merging, pruning, etc.).

max_score The maximum G4Hunter score for the window covering this G4.

sequence The G4 sequence after merging, pruning, etc.

If no G4 sequences were detected, it will return an empty GRanges object.

See Also

loadSequence for loading genome sequences into a DNAStringSet object.

Examples

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!requireNamespace("Biostrings", quietly = TRUE)) {
  BiocManager::install("Biostrings")
}

library(Biostrings)
sequences <- DNAStringSet(c(
  "AGTGAATGGGATGGGAGGAGGGACGGGGTAGTACAGCATAGCATG",
  "TAGGTAGCTACGACACCCTGCCCTACCCCTACCCCTATCTA"
))
names(sequences) <- c("seq1", "seq2")

G4s <- G4HunterDetect(sequences, threshold = 1.5, window_size = 25)
print(G4s)

seq_path <- system.file("extdata", "seq.fa", package = "G4SNVHunter")
G4s <- G4HunterDetect(loadSequence(seq_path = seq_path))
print(G4s)

seq_path <- system.file("extdata", "seq.txt", package = "G4SNVHunter")
G4s <- G4HunterDetect(loadSequence(seq_path = seq_path))
print(G4s)
```

G4HunterScore *Calculate the G4Hunter Score for a Given Sequence*

Description

This function calculates the G4Hunter score for a given nucleotide sequence, which reflects the ability of that sequence to form a G4 structure.

Usage

```
G4HunterScore(seq = NULL)
```

Arguments

seq	A single character string representing the nucleotide sequence. Must contain only the characters A, T, C, G, U, and N. The length of the sequence should not be too short (e.g., less than 10 bp).

Value

A numeric value representing the G4Hunter score for the provided sequence.

See Also

G4HunterDetect for detecting the G4 sequences in a given DNAStringSet object.

Examples

```
sequence <- "GGGTAAGGGATGGGTCGGG"
score <- G4HunterScore(sequence)
print(score)
# A negative value indicates that the G4 sequence
# is located on the reverse strand
sequence <- "GGGTAAGGGATGGGTCGGG"
score <- G4HunterScore(sequence)
print(score)
```

loadSequence *Load Genome Sequences*

Description

This function loads genomic sequences from various sources, including a FASTA file, a text file with sequence identifiers and corresponding sequences, or a data frame object.

Usage

```
loadSequence(genome_seq = NULL, seq_path = NULL)
```

Arguments

genome_seq	A data frame containing sequence identifiers and corresponding sequences.
seq_path	A character string specifying the file path to a FASTA file (suffixed with .fa, .fna or .fasta) or a text file with sequence identifiers and corresponding sequences (file headers should not be provided). Ignored if genome_seq is provided.

Value

A DNAStringSet object containing the genome sequences.

Examples

```
# File path for sequences in fasta format
fa_path <- system.file("extdata", "seq.fa", package = "G4SNVHunter")
seq <- loadSequence(seq_path = fa_path)
print(seq)

# Another example
# Load sequences from data.frame
seq_df <- data.frame(
  chr = c("seq1", "seq2"),
  sequence = c(
    paste0(rep("G", 100), collapse = ""),
    paste0(rep("A", 100), collapse = "")
  )
)
seq <- loadSequence(genome_seq = seq_df)
print(seq)
```

---

plotG4Info *Plot Basic Statistics of G4s Detected by the G4HunterDetect Function*

Description

This function generates a series of plots to visualize basic statistics of G4 sequences predicted by the G4Hunter algorithm. The function produces the following plots:

- Distribution of max scores (absolute values).
- Distribution of max scores, split by strand.
- Distribution of scores (absolute values).
- Distribution of scores, split by strand.
- Distribution of sequence lengths (with lengths greater than 50 bp grouped into a single bin).
- Distribution of sequence lengths, split by strand (with lengths greater than 50 bp grouped into a single bin).

Usage

```
plotG4Info(
  G4,
  p_colors = c("#6B9ECC", "#D91117", "#0E619C", "#58AC7B", "#D91117", "#0E619C",
    "#F39C40", "#D91117", "#0E619C")
)
```

Arguments

G4	A GRanges object containing G4 sequences. The object must include at least the following metadata columns: • score: Numeric vector of scores for G4 sequences. • max_score: Numeric vector of maximum scores for G4 sequences. • sequence: Character vector of G4 sequence strings. The GRanges object must also have a strand information. Whenever possible, it should come from the results returned by the G4HunterDetect function.
p_colors	A vector of colors to be used for the plots. It should contain nine color values for the different plot elements.

Value

A combined plot displays all the generated plots arranged in a grid layout.

See Also

G4HunterDetect for detecting the G4 sequences in a given DNAStringSet object.

Examples

```
seq_path <- system.file("extdata", "seq.txt", package = "G4SNVHunter")
G4 <- G4HunterDetect(loadSequence(seq_path = seq_path))
plotG4Info(G4)
```

---

plotImpactSeq *Visualize the variants in G4 sequence*

Description

This function plot sequence logos to visualize sequence variants caused by SNVs or SNPs, with the location of the variants highlighted by rectangles and arrows.

Usage

```
plotImpactSeq(filtered_gr, ncol = 1)
```

Arguments

filtered_gr	A GRanges object containing sequence data and G4Hunter scores. The object must have metadata columns named G4.info.score, mut.score, G4.info.sequence, and mut.G4.seq.
ncol	An integer specifying the number of columns in the output plot grid. Default is 1.

Value

A plot that displays the grid of sequence logos, showing the differences between the original and mutated sequences.

See Also

SNVImpactG4 for evaluating the impact of SNVs on G4 formation, and filterSNVImpact for filtering G4s that are significantly affected by SNVs.

Examples

```
if (!requireNamespace("GenomicRanges", quietly = TRUE)) {
  BiocManager::install("GenomicRanges")
}

library(GenomicRanges)

seq <- data.frame(chr = c("seq1", "seq2"),
                  seq = c("ATTTGGGAGGGAGGGAGGGATGATGAAAATTTTATTTATTTTATTTA",
                          "TTTATACTATTCCCTTACCCTCCCATCCCCATACGGCATCTAGATC"))
seq_gr <- loadSequence(seq)
G4 <- G4HunterDetect(seq_gr)

snv_gr <- GRanges(seqnames = c("seq1", "seq2"),
                  ranges = IRanges(start = c(18, 23), width = 1),
                  ref = c("G", "C"),
                  alt = c("C", "G"))

effect <- SNVImpactG4(G4, snv_gr, alt_col = "alt")
plotImpactSeq(effect, ncol = 2)
```

---

plotSNVImpact *Plot the Impact of SNVs on G4Hunter Scores*

Description

This function generates two plots for visualizing the impact of SNVs on G4 formation: 1. A scatter plot with density shading comparing the original G4Hunter score and the mutant G4Hunter score. 2. A density plot showing the distribution of score changes between the original and mutant G4 sequences.

Usage

```
plotSNVImpact(gr, p_colors = c("#b22d2d", "#6ca4d6", "#2d69b0", "#1f77b4"))
```

Arguments

gr	A GRanges object containing the G4Hunter scores and associated metadata. The object should include the following columns: • G4.info.score: Numeric vector of the original G4Hunter scores. • mut.score: Numeric vector of the G4Hunter scores after mutation. • score.diff: Numeric vector of the differences between the original and mutant G4Hunter scores.
p_colors	A vector of four colors used for plotting.

Value

A combined plot (using plot_grid) containing two subplots: - Density scatter plot comparing original vs mutant G4Hunter scores. - Density plot of the G4Hunter score differences.

See Also

SNVImpactG4 for evaluating the impact of SNVs on G4 formation, and filterSNVImpact for filtering G4s that are significantly affected by SNVs.

Examples

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!requireNamespace("GenomicRanges", quietly = TRUE)) {
  BiocManager::install("GenomicRanges")
}

if (!requireNamespace("BSgenome.Hsapiens.UCSC.hg19", quietly = TRUE)) {
  BiocManager::install("BSgenome.Hsapiens.UCSC.hg19")
}

library(GenomicRanges)
library(BSgenome.Hsapiens.UCSC.hg19)

hg19 <- BSgenome.Hsapiens.UCSC.hg19
chr21_seq <- DNAStringSet(hg19$chr21)
# Chromosome name is needed
names(chr21_seq) <- "chr21"

G4 <- G4HunterDetect(chr21_seq)

# Load SNVs
data(snp_gr)
```

```
res_snp <- SNVImpactG4(G4, snp_gr, alt_col = "alt")
plotSNVImpact(res_snp)
```

---

snp_gr *Single Nucleotide Polymorphisms GRanges Object*

Description

This dataset contains a GRanges object storing single nucleotide polymorphisms (SNPs).

Usage

```
data(snp_gr)
```

Format

A GRanges object with 30,000 SNPs.

SNVImpactG4 *Evaluate the Impact of SNVs on G4 Sequences*

Description

This function evaluates the impact of SNVs on G4 formation.

Usage

```
SNVImpactG4(
  G4 = NULL,
  snvs = NULL,
  alt_col = "alt",
  mode = "s",
  sampleid_col = NULL,
  snvid_col = NULL
)
```

Arguments

G4	A GRanges object representing the G4 regions. This object must include metadata columns for G4 sequence and G4Hunter scores. Whenever possible, it should come from the results returned by the G4HunterDetect function.
snvs	A GRanges object representing the SNVs. This object must include metadata columns for SNV alternative alleles and, optionally, SNV IDs.
alt_col	A character string specifying the name of the column in snvs that contains the alternative alleles for the SNVs. The default is "alt".

mode	A character string indicating the mode of operation. Set to "s" to evaluate the impact of individual SNVs on G4 regions one at a time. Set to "m" to assess the combined impact of multiple SNVs that overlap the same G4 region within a sample. If using "m" mode, you must specify the sampleid_col and snvid_col parameters.
sampleid_col	A character string specifying the name of the column in snvs that contains the sample IDs. This parameter is required when mode is "m", and is ignored if mode is "s".
snvid_col	A character string specifying the column name in snvs that contains SNV IDs. Required when mode is "m". Ignored if mode is "s".

Value

A GRanges object depending on the mode parameter:

Mode "s": For mode = "s", the returned GRanges object includes the following metadata columns:

- seqnames Identifiers for SNVs.
- ranges Position of the SNVs.
- strand Strand of the SNVs.
- SNV.info.* Metadata columns related to each SNV.
- G4.info.* Metadata columns from the original G4 object.
- mut.G4.seq The mutated G4 sequence after applying the SNV change.
- mut.G4.anno.seq The mutated G4 sequence, with the mutated bases annotated using square brackets.
- mut.score The G4Hunter score of the mutated sequence.
- score.diff The difference between the mutated G4Hunter score and the original G4Hunter score. The value is calculated as the absolute value of the mutated G4Hunter score minus the absolute value of the original G4Hunter score.

Mode "m": For mode = "m", the returned GRanges object includes the following metadata columns:

- seqnames Identifiers for G4 sequences.
- ranges Position of the G4 sequences (start and end).
- strand Strand of the G4 sequences.
- G4.info.* Metadata columns from the original G4 object.
- snv.ids Concatenated SNV IDs for all SNVs affecting the G4 region.
- sample.ids A semicolon-separated list of sample IDs overlapping each G4 region.
- mut.G4.seq The mutated G4 sequence after applying the combined SNV changes.
- mut.G4.anno.seq The mutated G4 sequence, with the mutated bases annotated using square brackets.
- mut.score The G4Hunter score of the mutated sequence.
- score.diff The difference between the mutated G4Hunter score and the original G4Hunter score. This value is calculated as the absolute value of the mutated G4Hunter score minus the absolute value of the original G4Hunter score.

See Also

G4HunterDetect for detecting the G4 sequences in a given DNAStringSet object. G4HunterScore for calculating the G4Hunter scores for a given sequence. filterSNVImpact for filtering out SNVs with significant impact.

Examples

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}

if (!requireNamespace("GenomicRanges", quietly = TRUE)) {
  BiocManager::install("GenomicRanges")
}

if (!requireNamespace("BSgenome.Hsapiens.UCSC.hg19", quietly = TRUE)) {
  BiocManager::install("BSgenome.Hsapiens.UCSC.hg19")
}

library(GenomicRanges)
library(BSgenome.Hsapiens.UCSC.hg19)

# Load sequence for chromosome 21 (hg19)
hg19 <- BSgenome.Hsapiens.UCSC.hg19
chr21_seq <- DNAStringSet(hg19$chr21)
# Chromosome name is needed
names(chr21_seq) <- "chr21"

# Detect G4s in human chromosome 21
G4 <- G4HunterDetect(chr21_seq)

# 's' mode
# Load SNPs
data(snp_gr)

# Obtain SNPs that overlap with G4 regions and assess their impact on G4.
# In variant-centric mode ('s'), evaluating each SNP individually.
res_snp <- SNVImpactG4(G4, snp_gr, alt_col = "alt")
print(res_snp)

# 'm' mode
# Load SNVs
data(snv_gr)

# Obtain SNVs that overlap with G4 regions and assess their impact on G4.
# In sample-centric mode ('m'), evaluate the combined impact of SNVs on G4s.
# Grouped by the sample IDs specified in 'sampleid_col'.
res_snv <- SNVImpactG4(G4, snv_gr,
  alt_col = "alt",
  mode = "m", sampleid_col = "sampleid", snvid_col = "snv_id"
)
print(res_snv)
```

snv_gr *Single Nucleotide Variant GRanges Object*

Description

This dataset contains a GRanges object storing single nucleotide variants (SNVs).

Usage

```
data(snv_gr)
```

Format

A GRanges object with 50,000 SNVs.

Index