Package 'TmCalculator'

March 20, 2026

Type Package

Title A Calculator for Melting Temperature of Nucleic Acid Sequences

Version 1.0.4

Date 2026-03-18

Description

A comprehensive R package for calculating melting temperatures of nucleic acid sequences.
Implements three calculation methods:
1. Wallace rule (Thein & Wallace, 1986)
2. Empirical formulas based on GC content (Marmur, 1962; Schildkraut, 2010; Wetmur, 1991; Untergasser, 2012; von Ahsen, 2001)
3. Nearest neighbor thermodynamics (Breslauer, 1986; Sugimoto, 1996; Allawi, 1998; SantaLucia, 2004; Freier, 1986; Xia, 1998; Chen, 2012; Bommarito, 2000; Turner, 2010; Sugimoto, 1995; Allawi, 1997; Santalucia, 2005)

Includes corrections for:
- Salt ions (SantaLucia, 1996, 1998; Owczarzy, 2004, 2008)
- Chemical compounds (dimethyl sulfoxide, formamide)

Supports both direct sequence input and FASTA file input.

BugReports https://github.com/JunhuiLi1017/TmCalculator/issues

License MIT + file LICENSE

Depends R (>= 3.5)

VignetteBuilder knitr

Imports seqinr, BSgenome, Biostrings, GenomicRanges, IRanges, S4Vectors, GenomeInfoDb, Gviz, dplyr, ggbio, ggplot2, karyoploteR, viridis, rlang, plotly

Suggests testthat (>= 3.0.0), knitr, rmarkdown, devtools, roxygen2, BSgenome.Hsapiens.UCSC.hg38

NeedsCompilation no

Repository CRAN

RoxygenNote 7.3.2

LazyData true

Encoding UTF-8

Author Junhui Li [cre, aut] (ORCID: <https://orcid.org/0000-0003-3973-1700>)

Maintainer Junhui Li <ljh.biostat@gmail.com>

Date/Publication 2026-03-20 16:50:30 UTC

Contents

`check_filter_seq` *Filter invalid bases in nucleotide sequences*

Description

This function processes nucleotide sequences by converting characters to uppercase and replacing invalid bases with "". based on the specified method. The function preserves the sequence length and attributes (name and Tm) of each sequence.

Usage

```
check_filter_seq(seq_list, method)
```

Arguments

seq_list	Input sequence in 5' to 3' direction. Must be provided as: - A list of sequences with attributes (name and Tm)
method	Method to determine valid bases:
	TM_Wallace: Valid bases are "A","B","C","D","G","H","I","K","M","N","R","S","T","V","W" and "Y"
	TM_GC: Valid bases are "A","B","C","D","G","H","I","K","M","N","R","S","T","V","W", "X" and "Y"
	TM_NN: Valid bases are "A","C","G","I" and "T"

Value

Returns a list of sequences with the same structure as input, where invalid bases are replaced with ""

Author(s)

Junhui Li

References

```
citation("TmCalculator")
```

---

chem_correct *Corrections of melting temperature with chemical substances*

Description

Apply corrections to melting temperature calculations based on the presence of DMSO and formamide. These corrections are rough approximations and should be used with caution.

Usage

```
chem_correct(
  DMSO = 0,
  formamide_value_unit = list(value = 0, unit = "percent"),
  dmso_factor = 0.75,
  formamide_factor = 0.65,
  pt_gc = NULL
)
```

Arguments

DMSO	Percent DMSO concentration in the reaction mixture. Default: 0 DMSO can lower the melting temperature of nucleic acid duplexes.
formamide_value_unit	A list containing formamide concentration information: - value: numeric value of formamide concentration - unit: character string specifying the unit ("percent" or "molar") Default: list(value=0, unit="percent")
dmso_factor	Coefficient of melting temperature (Tm) decrease per percent DMSO. Default: 0.75 (von Ahsen N, 2001, PMID:11673362) Other published values: 0.5, 0.6, 0.675
formamide_factor	Coefficient of melting temperature (Tm) decrease per percent formamide. Default: 0.65 Literature reports values ranging from 0.6 to 0.72
pt_gc	Percentage of GC content in the sequence (0-100 This is used in molar formamide corrections.

Details

When formamide_value_unit$unit = "percent": Correction = - factor * percentage_of_formamide

When formamide_value_unit$unit = "molar": Correction = (0.453 * GC/100 - 2.88) * formamide

Author(s)

Junhui Li

References

von Ahsen N, Wittwer CT, Schutz E, et al. Oligonucleotide melting temperatures under PCR conditions: deoxynucleotide Triphosphate and Dimethyl sulfoxide concentrations with comparison to alternative empirical formulas. Clin Chem 2001, 47:1956-1961.

Examples

```
# DMSO correction
chem_correct(DMSO = 3)

# Formamide correction (percent)
chem_correct(formamide_value_unit = list(value = 1.25, unit = "percent"), pt_gc = 50)

# Formamide correction (molar)
chem_correct(formamide_value_unit = list(value = 1.25, unit = "molar"), pt_gc = 50)
```

coor_to_genomic_ranges *Convert genomic coordinate strings to GenomicRanges object with sequences*

Description

This function converts genomic coordinate strings in the format "chr:start-end:strand:species[:name]" to a GenomicRanges object containing the corresponding sequences from the specified reference genome.

Usage

```
coor_to_genomic_ranges(input_seq)
```

Arguments

input_seq	A character vector where each element is a string in the format: "chr:start-end:strand:species[:name]" - chr: Chromosome ID (e.g., "chr1", "chrX") - start: Start position (integer) - end: End position (integer) - strand: "+" for positive strand or "-" for negative strand - species: Species name for reference genome (e.g., "hg38") - name: (optional) Custom name for the sequence

Value

A GenomicRanges object containing: - GRanges information (seqnames, ranges, strand) - sequence data from the reference genome - Names either from the optional name parameter or auto-generated as "1", "2", etc.

Examples

```
## Not run:
# Example with multiple coordinates
coords <- c(
  "chr1:1898000-1898050:+:BSgenome.Hsapiens.UCSC.hg38:exon1",
  "chr2:2563000-2563050:-:BSgenome.Hsapiens.UCSC.hg38:exon2"
)
gr <- coor_to_genomic_ranges(coords)

## End(Not run)
```

fa_to_genomic_ranges *Convert FASTA file to GenomicRanges object*

Description

This function reads sequences from a FASTA file and converts them to a GenomicRanges object. If named with format ">chr2:1-10:[+|-]:[seq_name]", the name will be parsed into GRanges components.

Usage

```
fa_to_genomic_ranges(input_seq)
```

Arguments

input_seq	Path to the input FASTA file

Value

A GenomicRanges object containing: - GRanges information (seqnames, ranges, strand) - sequence data from FASTA file - Complementary sequences (if provided) - Names from FASTA headers

Examples

```
# Example with single FASTA file
input_seq <- system.file("extdata", "example1.fasta", package = "TmCalculator")
gr <- fa_to_genomic_ranges(input_seq)
```

gc *Calculate G and C content of nucleotide sequences*

Description

Calculate G and C content of nucleotide sequences. The function calculates the percentage of G and C bases relative to the total number of A, T, G, and C bases in the sequence.

Usage

```
gc(input_seq, ambiguous = FALSE)
```

Arguments

input_seq	Sequence (5' to 3') of one strand of the nucleic acid duplex. Can be provided as either: - A character string (e.g., "ATGCG") - A path to a FASTA file containing the sequence(s)
ambiguous	Logical. If TRUE, ambiguous bases are taken into account when computing the G and C content. The function handles various ambiguous bases (S, W, M, K, R, Y, V, H, D, B) by proportionally distributing their contribution to GC content based on their possible nucleotide compositions. For example: - S (G or C) contributes fully to GC content - W (A or T) contributes fully to AT content - M (A or C) contributes proportionally based on the ratio of A to C in the sequence - And so on for other ambiguous bases

Value

Content of G and C as a percentage (range from 0 to 100

Author(s)

Junhui Li

Examples

```
# Calculate GC content of a DNA sequence
gc(c("a","t","c","t","g","g","g","c","c","a","g","t","a"))  # 53.85%

# Calculate GC content including ambiguous bases
gc("GCATSWSYK", ambiguous = TRUE)  # 55.56%
```

---

generate_complement *Generate complementary sequence*

Description

Generate the complementary sequence of a nucleic acid sequence, with an option to reverse it.

Usage

```
generate_complement(input_seq, reverse = FALSE)
```

Arguments

input_seq	Input sequence(s) in 5' to 3' direction. Must be provided as either: - A character string (e.g., c("ATGCG", "GCTAG"))
reverse	Logical. If TRUE, the complementary sequence is reversed (3' to 5'). If FALSE (default), the complementary sequence is in the same direction (5' to 3').

Value

Returns the complementary sequence(s) in the specified direction.

Author(s)

Junhui Li

References

```
citation("TmCalculator")
```

Examples

```
# Generate complementary sequence in same direction (5' to 3')
generate_complement("ATGCG", reverse = FALSE)

# Generate complementary sequence in reverse direction (3' to 5')
generate_complement("ATGCG", reverse = TRUE)
```

`plot_tm_genome_tracks` *Plot Tm values as Genome Browser Tracks using Gviz*

Description

This function generates Gviz plots displaying Tm values as DataTracks alongside genome axes and ideograms for specified chromosomes. Tm values are visualized using a heatmap-like color gradient.

Usage

```
plot_tm_genome_tracks(
  gr,
  chromosome_to_plot,
  genome_assembly = NULL,
  tm_track_title = "Melting Temperature (°C)",
  color_palette = c("viridis", "magma", "plasma", "inferno", "cividis"),
  show_ideogram = TRUE,
  zoom = NULL
)
```

Arguments

`gr`	A GRanges object. It MUST contain a metadata column named 'Tm' with numeric melting temperature values.
`chromosome_to_plot`	A character string specifying the chromosome to visualize. These chromosomes must exist in your GRanges object.

genome_assembly
: A character string indicating the genome assembly (e.g., "hg19", "mm10"). This is used by IdeogramTrack for correct ideogram display.

tm_track_title
: A character string for the title of the Tm data track.

color_palette
: A character string specifying the viridis color palette to use. Available options are:

  - "viridis" (default): A perceptually uniform color map that works well for most people
  - "magma": A perceptually uniform color map with a dark purple to bright yellow range
  - "plasma": A perceptually uniform color map with a dark purple to bright yellow range
  - "inferno": A perceptually uniform color map with a dark purple to bright yellow range
  - "cividis": A perceptually uniform color map optimized for color vision deficiency

  All palettes are colorblind-friendly and perceptually uniform.

show_ideogram
: Logical, whether to display the chromosome ideogram tracks.

zoom
: A character string specifying the genomic region to zoom into. If NULL (default), the entire range of each chromosome will be shown. Example: "chr1:1000000-2000000" for zooming into chr1:1000000-2000000

Value

Invisible NULL. The function generates a plot directly.

Examples

```
## Not run:
library(GenomicRanges)
library(Gviz)
# Example 1: Generate sample data with 150 sequences
set.seed(123)

# Generate 100 sequences for chr1
chr1_starts <- sort(sample(1:249250621, 100))  # chr1 length in hg19
chr1_lengths <- sample(50:200, 100, replace=TRUE)
chr1_ends <- chr1_starts + chr1_lengths
chr1_tms <- runif(100, min=60, max=80)

# Generate 50 sequences for chr2
chr2_starts <- sort(sample(1:243199373, 50))   # chr2 length in hg19
chr2_lengths <- sample(50:200, 50, replace=TRUE)
chr2_ends <- chr2_starts + chr2_lengths
chr2_tms <- runif(50, min=60, max=80)

# Create GRanges object
tm_results <- GRanges(
```

```
  seqnames = Rle(c(rep("chr1", 100), rep("chr2", 50))),
  ranges = IRanges(
    start = c(chr1_starts, chr2_starts),
    end = c(chr1_ends, chr2_ends)
  ),
  strand = Rle(sample(c("+", "-"), 150, replace=TRUE)),
  Tm = c(chr1_tms, chr2_tms)
)

# Plot single chromosome with zoom
plot_tm_genome_tracks(
  gr = tm_results,
  chromosome_to_plot = "chr1",
  genome_assembly = "hg19",
  tm_track_title = "DNA Sequence Tm",
  zoom = "chr1:10062800-20000000"
)

# Example with custom color palette and no zoom
plot_tm_genome_tracks(
  gr = tm_results,
  chromosome_to_plot = "chr2",
  genome_assembly = "hg19",
  color_palette = "plasma"
)

## End(Not run)
```

`plot_tm_heatmap` *Plot Tm values as a heatmap using ggbio*

Description

This function generates a heatmap visualization of Tm values across chromosomes using the ggbio package. It supports both karyogram and faceted plot types.

Usage

```
plot_tm_heatmap(
  gr,
  genome_assembly = NULL,
  chromosome_to_plot = NULL,
  plot_type = c("karyogram", "faceted"),
  color_palette = c("viridis", "magma", "plasma", "inferno", "cividis"),
  title_name = NULL,
  zoom = NULL
)
```

Arguments

gr	A GRanges object. It MUST contain a metadata column named 'Tm' with numeric melting temperature values.
genome_assembly	A character string indicating the genome assembly (e.g., "hg19", "mm10"). This is used by ggbio for correct chromosome display.
chromosome_to_plot	A character vector specifying which chromosomes to visualize. These chromosomes must exist in your GRanges object.
plot_type	A character string specifying the type of plot to generate: • "karyogram" (default): A single plot with all chromosomes arranged in a karyogram • "faceted": Separate plots for each chromosome
color_palette	A character string specifying the viridis color palette to use. Available options are: • "viridis" (default): A perceptually uniform color map that works well for most people • "magma": A perceptually uniform color map with a dark purple to bright yellow range • "plasma": A perceptually uniform color map with a dark purple to bright yellow range • "inferno": A perceptually uniform color map with a dark purple to bright yellow range • "cividis": A perceptually uniform color map optimized for color vision deficiency All palettes are colorblind-friendly and perceptually uniform.
title_name	A character string for the plot title.
zoom	A character string specifying the genomic region to zoom into. If NULL (default), the entire range of each chromosome will be shown. Example: c("chr1:1000000-2000000", "chr2:1000000-2000000") for zooming into chr1:1000000-2000000 and chr2:1000000-2000000

Value

A ggplot object displaying Tm values across genomic coordinates.

Examples

```
## Not run:
# Create example GRanges object
gr_tm <- GenomicRanges::GRanges(
  seqnames = c("chr1", "chr2", "chr1", "chr2", "chr1"),
  ranges = IRanges::IRanges(
    start = c(100, 200, 300, 400, 150),
    end = c(150, 250, 350, 450, 200)
  ),
```

```
  Tm = c(65.5, 68.2, 70.1, 63.8, 72.0)
)

# Plot with ideograms
plot_tm_heatmap(gr_tm, genome_assembly = "hg19", plot_type = "karyogram")

# Faceted plot by chromosome
plot_tm_heatmap(gr_tm, genome_assembly = "hg19", plot_type = "faceted")

# Plot with zoom
plot_tm_heatmap(gr_tm, genome_assembly = "hg19", plot_type = "faceted", zoom = "chr1:100-200")


## End(Not run)
```

`plot_tm_heatmap_interactive` *Convert Tm plots to interactive plotly versions*

Description

These functions convert the standard Tm plots to interactive plotly versions that can be used in Shiny applications or R Markdown documents.

These functions convert the standard Tm karyotype plots to interactive plotly versions that can be used in Shiny applications or R Markdown documents.

These functions convert the standard Tm genome tracks plots to interactive plotly versions that can be used in Shiny applications or R Markdown documents.

Usage

```
plot_tm_heatmap_interactive(
  gr,
  genome_assembly = NULL,
  chromosome_to_plot = NULL,
  plot_type = c("karyogram", "faceted"),
  color_palette = c("viridis", "magma", "plasma", "inferno", "cividis"),
  title_name = NULL,
  zoom = NULL
)

plot_tm_karyotype_interactive(
  gr,
  chromosomes = NULL,
  genome_assembly = NULL,
  colors = NULL,
  shapes = NULL,
```

```
  plot_type = 1,
  point_cex = 1.5,
  xaxis_cex = 0.7,
  yaxis_cex = 0.8,
  chr_cex = 1,
  tick_dist = 1e+07,
  zoom = NULL
)

plot_tm_genome_tracks_interactive(
  gr,
  chromosome_to_plot,
  genome_assembly = NULL,
  tm_track_title = "Melting Temperature (°C)",
  color_palette = c("viridis", "magma", "plasma", "inferno", "cividis"),
  show_ideogram = TRUE,
  zoom = NULL
)
```

Arguments

gr	A GRanges object containing the Tm values.
genome_assembly	A string specifying the genome assembly.
chromosome_to_plot	A string specifying the chromosome to plot.
plot_type	A string specifying the plot type.
color_palette	A string specifying the color palette.
title_name	A string specifying the title name.
zoom	A string specifying the zoom level.
chromosomes	A vector of strings specifying the chromosomes to plot.
colors	A vector of strings specifying the colors for the chromosomes.
shapes	A vector of strings specifying the shapes for the chromosomes.
point_cex	A numeric value specifying the point size.
xaxis_cex	A numeric value specifying the x-axis label size.
yaxis_cex	A numeric value specifying the y-axis label size.
chr_cex	A numeric value specifying the chromosome label size.
tick_dist	A numeric value specifying the tick distance.
tm_track_title	A string specifying the title name.
show_ideogram	A logical value specifying whether to show the ideogram.

Value

A plotly object.

`plot_tm_karyotype` *Plot Tm Values from GRanges with Per-Chromosome Colors and Shapes*

Description

Creates a genome-wide plot of melting temperature (Tm) values from a GRanges object using the karyoploteR package. The x-axis represents genomic positions across chromosomes, and the y-axis represents Tm values. Points are plotted at the midpoints of genomic ranges, with customizable colors and shapes per chromosome.

Usage

```
plot_tm_karyotype(
  gr,
  chromosomes = NULL,
  genome_assembly = NULL,
  colors = NULL,
  shapes = NULL,
  plot_type = 1,
  point_cex = 1.5,
  xaxis_cex = 0.7,
  yaxis_cex = 0.8,
  chr_cex = 1,
  tick_dist = 1e+07,
  zoom = NULL
)
```

Arguments

`gr`	A `GRanges` object with a `Tm` metadata column containing numeric melting temperature values.
`chromosomes`	A character vector specifying chromosomes to plot. If `NULL`, all unique chromosomes in `gr` are plotted. Defaults to `NULL`.
`genome_assembly`	It can be either a UCSC style genome name (hg19, mm10, etc), a BSgenome, a Seqinfo object, a GRanges object with the chromosomes as ranges or in general any genome specification accepted by karyoploteR. If `NULL`, uses default or GRanges seqinfo. Defaults to `NULL`.
`colors`	A named character vector specifying colors for each chromosome (e.g., `c(chr1 = "#FF0000", chr22 = "#00FF00")`). Names must match chromosomes in `gr` or `chromosomes`. If `NULL` or partially specified, unspecified chromosomes use the first viridis color. Defaults to `NULL`.
`shapes`	A named integer vector specifying point shapes (pch values) for each chromosome (e.g., `c(chr1 = 16, chr22 = 17)`). Names must match chromosomes in `gr` or `chromosomes`. If `NULL` or partially specified, unspecified chromosomes use `pch = 16` (filled circles). Defaults to `NULL`.

plot_type	An integer specifying the karyoploteR plot type (e.g., 1 for horizontal chromosomes, 4 or 7 for vertical or grid layouts). See plotKaryotype for details. Defaults to 1.
point_cex	A numeric value for the size of plotted points. Defaults to 1.5.
xaxis_cex	A numeric value for the text size of x-axis labels (base pair positions). Defaults to 0.7.
yaxis_cex	A numeric value for the text size of y-axis labels (Tm values). Defaults to 0.8.
chr_cex	A numeric value for the text size of chromosome names. Defaults to 1.
tick_dist	A numeric value for the distance between tick marks on the x-axis. Defaults to 10000000.
zoom	A GRanges object specifying a genomic region to zoom into. If NULL, the full chromosomes are plotted. Defaults to NULL.

Details

The function validates that gr is a GRanges object with a Tm column. It automatically sets the y-axis limits based on the range of Tm values, with slight padding (floor and ceiling). The plot includes chromosome names (via karyoploteR's default labeling), base pair positions, and a labeled y-axis. Points are colored and shaped according to the colors and shapes parameters, with defaults applied for unspecified chromosomes. The y-axis is placed on the left for plot_type = 1 and on the right for plot_type = 4 or 7, with the label positioned to the right of the y-axis, vertically centered in the middle of the y-axis range. Text sizes for x-axis labels, y-axis labels, and chromosome names can be customized using xaxis_cex, yaxis_cex, and chr_cex, respectively.

Value

Invisibly returns NULL. The function generates a plot as a side effect.

Examples

```
## Not run:
library(GenomicRanges)
# Create a sample GRanges object
gr <- GRanges(
  seqnames = c("chr22", "chr1", "chr14", "chr22"),
  ranges = IRanges(
    start = c(13209021, 1, 13200, 13209150),
    end = c(13209099, 76, 13222, 13209200)
  ),
  strand = c("+", "*", "*", "-"),
  Tm = c(69.1147, 71.1160, 50.7169, 65.5000)
)
genome(seqinfo(gr)) <- "hg19"

# Plot with default settings (plot_type=1)
plot_tm_karyotype(gr)

# Plot with partial color and shape specification (plot_type=4)
plot_tm_karyotype(
```

```
  gr,
  genome_assembly="hg19",
  colors = c(chr1 = "#FF0000"),  # chr14, chr22 use default color
  shapes = c(chr1 = 16, chr14 = 17),  # chr22 uses pch=16
  plot_type = 4,
  xaxis_cex = 0.6,
  yaxis_cex = 0.9,
  chr_cex = 1.2
)

# Plot with full color and shape specification (plot_type=7)
plot_tm_karyotype(
  gr,
  genome_assembly="hg38",
  colors = c(chr1 = "#FF0000", chr14 = "#00FF00", chr22 = "#0000FF"),
  shapes = c(chr1 = 16, chr14 = 17, chr22 = 16),
  plot_type = 5,
  xaxis_cex = 0.8,
  yaxis_cex = 0.7,
  chr_cex = 0.8
)

# Plot with zoom into chr22
zoom_region <- GRanges("chr22:13200000-13220000")
plot_tm_karyotype(
  gr,
  genome_assembly="hg38",
  chromosomes = "chr22",
  zoom = zoom_region,
  xaxis_cex = 0.5,
  yaxis_cex = 1,
  chr_cex = 1.5
)

## End(Not run)
```

`print.TmCalculator` *Prints melting temperature from a* `TmCalculator` *object*

Description

`print.TmCalculator` prints to console the melting temperature value from an object of class `TmCalculator`.

Usage

```
## S3 method for class 'TmCalculator'
print(x, ...)
```

Arguments

x	An object of class TmCalculator.
...	Unused

Value

The melting temperature value.

---

salt_correction *Corrections of melting temperature with salt concentration*

Description

Apply corrections to melting temperature calculations based on salt concentrations. Different correction methods are available for various experimental conditions.

Usage

```
salt_correction(
  Na = 0,
  K = 0,
  Tris = 0,
  Mg = 0,
  dNTPs = 0,
 method = c("Schildkraut2010", "Wetmur1991", "SantaLucia1996", "SantaLucia1998-1",
    "SantaLucia1998-2", "Owczarzy2004", "Owczarzy2008"),
  input_seq,
  ambiguous = FALSE
)
```

Arguments

Na	Millimolar concentration of sodium ions. Default: 0
K	Millimolar concentration of potassium ions. Default: 0
Tris	Millimolar concentration of Tris buffer. Default: 0
Mg	Millimolar concentration of magnesium ions. Default: 0
dNTPs	Millimolar concentration of deoxynucleotide triphosphates. Default: 0
method	Method for calculating salt concentration corrections to the melting temperature. Available options: - "Schildkraut2010": Updated salt correction method - "Wetmur1991": Classic salt correction method - "SantaLucia1996": DNA-specific salt correction - "SantaLucia1998-1": Improved DNA salt correction - "SantaLucia1998-2": Alternative DNA salt correction (requires input_seq) - "Owczarzy2004": Comprehensive salt correction (requires input_seq) - "Owczarzy2008": Updated comprehensive salt correction (requires input_seq) Note: Setting to NA disables salt correction

`input_seq` Sequence (5' to 3') of one strand of the nucleic acid duplex. Can be provided as either: - A character string (e.g., "ATGCG") - A path to a FASTA file containing the sequence(s) Required for methods: "SantaLucia1998-2", "Owczarzy2004", and "Owczarzy2008"

`ambiguous` Logical. If TRUE, ambiguous bases are taken into account when computing the G and C content. The function handles various ambiguous bases (S, W, M, K, R, Y, V, H, D, B) by proportionally distributing their contribution to GC content based on their possible nucleotide compositions.

Details

Different correction methods are available for various experimental conditions:

- Schildkraut2010: Updated salt correction method that accounts for monovalent and divalent cations - Wetmur1991: Classic salt correction method for monovalent cations - SantaLucia1996: DNA-specific salt correction - SantaLucia1998-1: Improved DNA salt correction - SantaLucia1998-2: Alternative DNA salt correction (requires sequence information) - Owczarzy2004: Comprehensive salt correction including effects of divalent cations (requires sequence information) - Owczarzy2008: Updated comprehensive salt correction (requires sequence information)

Author(s)

Junhui Li

References

Schildkraut C, Lifson S. Dependence of the melting temperature of DNA on salt concentration. Biopolymers. 1965;3(2):195-208.

Wetmur JG. DNA Probes: Applications of the Principles of Nucleic Acid Hybridization. Critical Reviews in Biochemistry and Molecular Biology. 1991;26(3-4):227-259.

SantaLucia J. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. Proceedings of the National Academy of Sciences. 1998;95(4):1460-1465.

Owczarzy R, Moreira BG, Manthey JA, et al. Predicting stability of DNA duplexes in solutions containing magnesium and monovalent cations. Biochemistry. 2008;47(19):5336-5353.

Examples

```
salt_correction(Na = 50, Mg = 1.5, method = "Owczarzy2008",
               input_seq = "ATGCGATGCG")
```

thermodynamic_gc_params

Thermodynamic parameters for GC-based Tm calculation methods

Description

A data frame containing coefficients and parameters for different GC-based Tm calculation methods. Each row represents a different method with its specific coefficients (A, B, C, D) and salt correction method.

Usage

```
thermodynamic_gc_params
```

Format

A data frame with 8 rows and 5 columns:

A Intercept coefficient

B GC content coefficient

C Length correction coefficient

D Mismatch coefficient

salt_correction Associated salt correction method

Details

The methods included are: - Chester1993: Tm = 69.3 + 0.41(Percentage_GC) - 650/N - QuikChange: Tm = 81.5 + 0.41(Percentage_GC) - 675/N - Percentage_mismatch - Schildkraut1965: Tm = 81.5 + 0.41(Percentage_GC) - 675/N + 16.6 x log[Na+] - Wetmur1991_MELTING: Tm = 81.5 + 0.41(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Wetmur1991_RNA: Tm = 78 + 0.7(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Wetmur1991_RNA/DNA: Tm = 67 + 0.8(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Primer3Plus: Tm = 81.5 + 0.41(Percentage_GC) - 600/N + 16.6 x log[Na+] - vonAhsen2001: Tm = 77.1 + 0.41(Percentage_GC) - 528/N + 11.7 x log[Na+]

thermodynamic_nn_params

Thermodynamic Tables for Nucleic Acid Hybridization

Description

A comprehensive collection of thermodynamic parameters used for calculating melting temperatures of nucleic acid duplexes. The dataset includes parameters for DNA/DNA, RNA/RNA, and RNA/DNA hybridizations, as well as parameters for mismatches and dangling ends.

Usage

```
thermodynamic_nn_params
```

Format

A list containing 12 matrices of thermodynamic parameters:

DNA_NN_Breslauer_1986 DNA/DNA nearest neighbor parameters from Breslauer et al. (1986)

DNA_NN_Sugimoto_1996 DNA/DNA nearest neighbor parameters from Sugimoto et al. (1996)

DNA_NN_Allawi_1998 DNA/DNA nearest neighbor parameters from Allawi et al. (1998)

DNA_NN_SantaLucia_2004 DNA/DNA nearest neighbor parameters from SantaLucia (2004)

RNA_NN_Freier_1986 RNA/RNA nearest neighbor parameters from Freier et al. (1986)

RNA_NN_Xia_1998 RNA/RNA nearest neighbor parameters from Xia et al. (1998)

RNA_NN_Chen_2012 RNA/RNA nearest neighbor parameters from Chen et al. (2012)

RNA_DNA_NN_Sugimoto_1995 RNA/DNA nearest neighbor parameters from Sugimoto et al. (1995)

DNA_IMM_Peyret_1999 DNA internal mismatch parameters from Peyret et al. (1999)

DNA_TMM_Bommarito_2000 DNA terminal mismatch parameters from Bommarito et al. (2000)

DNA_DE_Bommarito_2000 DNA dangling end parameters from Bommarito et al. (2000)

RNA_DE_Turner_2010 RNA dangling end parameters from Turner et al. (2010)

Each matrix contains thermodynamic parameters (enthalpy and entropy) for different nucleic acid interactions. The parameters are used in the nearest neighbor model for calculating melting temperatures of nucleic acid duplexes.

Source

Various publications as cited in the references

References

Breslauer K J (1986) <doi:10.1073/pnas.83.11.3746> Sugimoto N (1996) <doi:10.1093/nar/24.22.4501> Allawi H (1998) <doi:10.1093/nar/26.11.2694> SantaLucia J (2004) <doi:10.1146/annurev.biophys.32.110601.141800> Freier S (1986) <doi:10.1073/pnas.83.24.9373> Xia T (1998) <doi:10.1021/bi9809425> Chen JL (2012) <doi:10.1021/bi3002709> Sugimoto N (1995) <doi:10.1016/S0048-9697(98)00088-6> Bommarito S (2000) <doi:10.1093/nar/28.9.1929> Peyret N (1999) <doi:10.1021/bi9825091> Allawi H T (1997) <doi:10.1021/bi962590c> Santalucia N (2005) <doi:10.1093/nar/gki918> Turner D H (2010) <doi:10.1093/nar/gkp892>

Examples

```
# Access DNA/DNA nearest neighbor parameters
thermodynamic_nn_params$DNA_NN_SantaLucia_2004

# Access RNA/RNA nearest neighbor parameters
thermodynamic_nn_params$RNA_NN_Chen_2012
```

```
# Access DNA internal mismatch parameters
thermodynamic_nn_params$DNA_IMM_Peyret_1999
```

`tm_calculate` *Calculate melting temperature using multiple methods*

Description

Calculates melting temperature using multiple methods: - Nearest Neighbor thermodynamics (tm_nn) - GC content-based method (tm_gc) - Wallace rule (tm_wallace)

Usage

```
tm_calculate(
  input_seq,
  method = c("tm_nn", "tm_gc", "tm_wallace"),
  complement_seq = NULL,
  ambiguous = FALSE,
  shift = 0,
 nn_table = c("DNA_NN_SantaLucia_2004", "DNA_NN_Breslauer_1986", "DNA_NN_Sugimoto_1996",
   "DNA_NN_Allawi_1998", "RNA_NN_Freier_1986", "RNA_NN_Xia_1998", "RNA_NN_Chen_2012",
     "RNA_DNA_NN_Sugimoto_1995"),
  tmm_table = "DNA_TMM_Bommarito_2000",
  imm_table = "DNA_IMM_Peyret_1999",
  de_table = c("DNA_DE_Bommarito_2000", "RNA_DE_Turner_2010"),
  dnac_high = 25,
  dnac_low = 25,
  self_comp = FALSE,
  variant = c("Primer3Plus", "Chester1993", "QuikChange", "Schildkraut1965",
   "Wetmur1991_MELTING", "Wetmur1991_RNA", "Wetmur1991_RNA/DNA", "vonAhsen2001"),
  userset = NULL,
  Na = 50,
  K = 0,
  Tris = 0,
  Mg = 0,
  dNTPs = 0,
  salt_corr_method = c("Schildkraut2010", "Wetmur1991", "SantaLucia1996",
    "SantaLucia1998-1", "Owczarzy2004", "Owczarzy2008"),
  DMSO = 0,
  formamide_value_unit = list(value = 0, unit = "percent"),
  dmso_factor = 0.75,
  formamide_factor = 0.65,
  mismatch = TRUE
)
```

Arguments

input_seq	Input sequence(s) in 5' to 3' direction. Can be provided as either: - A character string (e.g., "ATGCG") - A path to a FASTA file containing the sequence(s) - A character vector where each element is a string in the format "chr:start-end:strand:species" #' (e.g., "chr1:100-200:+:hg38"). Strand is "+" for positive (default if not provided) or "-" for negative. - chr: Chromosome ID - start: Start position - end: End position - strand: positive or negtive strand - species: Species name for reference genome (e.g., "hg38"), the function will attempt to load the appropriate BSgenome from genome package.
method	Method(s) to use for Tm calculation. Can be one or more of: - "tm_nn": Nearest Neighbor thermodynamics (default) - "tm_gc": GC content-based method - "tm_wallace": Wallace rule Default: c("tm_nn", "tm_gc", "tm_wallace")
complement_seq	Complementary sequence(s) in 3' to 5' direction. If not provided, the function will automatically generate it from input_seq. This is the template/target sequence that the input sequence will hybridize with. Can be provided as input_seq format besides A NULL value(default)
ambiguous	Logical. If TRUE, ambiguous bases are taken into account when computing the G and C content. The function handles various ambiguous bases (S, W, M, K, R, Y, V, H, D, B) by proportionally distributing their contribution to GC content based on their possible nucleotide compositions. Default: FALSE
shift	Integer value controlling the alignment offset between primer and template sequences. Only applicable for the NN method. Default: 0
nn_table	Thermodynamic nearest-neighbor parameters for different nucleic acid hybridizations. Only applicable for the NN method. Default: "DNA_NN_SantaLucia_2004"
tmm_table	Thermodynamic parameters for terminal mismatches. Only applicable for the NN method. Default: "DNA_TMM_Bommarito_2000"
imm_table	Thermodynamic parameters for internal mismatches. Only applicable for the NN method. Default: "DNA_IMM_Peyret_1999"
de_table	Thermodynamic parameters for dangling ends. Only applicable for the NN method. Default: "DNA_DE_Bommarito_2000"
dnac_high	Concentration of the higher concentrated strand in nM. Only applicable for the NN method. Default: 25
dnac_low	Concentration of the lower concentrated strand in nM. Only applicable for the NN method. Default: 25
self_comp	Logical value indicating if the sequence is self-complementary. Only applicable for the NN method. Default: FALSE
variant	Empirical constants coefficient for GC method. Only applicable for the GC method. Default: "Primer3Plus"
userset	A vector of four coefficient values for GC method. Only applicable for the GC method. Usersets override value sets. Default: NULL
Na	Millimolar concentration of sodium ions. Default: 50
K	Millimolar concentration of potassium ions. Default: 0
Tris	Millimolar concentration of Tris buffer. Default: 0

`Mg` Millimolar concentration of magnesium ions. Default: 0

`dNTPs` Millimolar concentration of deoxynucleotide triphosphates. Default: 0

`salt_corr_method`
Method for calculating salt concentration corrections to the melting temperature. Available options: - "Schildkraut2010": Updated salt correction method - "Wetmur1991": Classic salt correction method - "SantaLucia1996": DNA-specific salt correction - "SantaLucia1998-1": Improved DNA salt correction - "SantaLucia1998-2": Alternative DNA salt correction - "Owczarzy2004": Comprehensive salt correction - "Owczarzy2008": Updated comprehensive salt correction Default: "Schildkraut2010"

`DMSO` Percent DMSO concentration in the reaction mixture. Default: 0

`formamide_value_unit`
List containing formamide concentration value and unit. Default: list(value = 0, unit = "percent") - value: Numeric value of formamide concentration - unit: Either "percent" or "molar"

`dmso_factor` Coefficient of Tm decreases per percent DMSO. Default: 0.75 Other published values are 0.5, 0.6 and 0.675.

`formamide_factor`
Coefficient of Tm decrease per percent formamide. Default: 0.65 Several papers report factors between 0.6 and 0.72.

`mismatch` Logical. If TRUE, every '.' in the sequence is counted as a mismatch. Only applicable for the GC method. Default: TRUE

Details

The function calculates melting temperature using the specified method(s). For each method: - NN: Uses nearest neighbor thermodynamics with detailed sequence analysis - GC: Uses GC content-based calculation with various empirical formulas - Wallace: Uses the simple Wallace rule (2°C per A/T, 4°C per G/C)

The function processes the input sequence once and applies it to all selected methods, making it more efficient than calling each method separately.

Value

A list containing Tm values and options for each method used. The structure includes: - Tm: A list of sequences with updated Tm attributes - Options: A list containing calculation parameters and method information

Available Options

Method Selection:

- `method`: c("tm_nn", "tm_gc", "tm_wallace")

Nearest Neighbor (NN) Method Options:

- `nn_table`:
  - "DNA_NN_Breslauer_1986"

  - "DNA_NN_Sugimoto_1996"
  - "DNA_NN_Allawi_1998"
  - "DNA_NN_SantaLucia_2004" (default)
  - "RNA_NN_Freier_1986"
  - "RNA_NN_Xia_1998"
  - "RNA_NN_Chen_2012"
  - "RNA_DNA_NN_Sugimoto_1995"
- `tmm_table` (Terminal Mismatches):
  - "DNA_TMM_Bommarito_2000" (default)
- `imm_table` (Internal Mismatches):
  - "DNA_IMM_Peyret_1999" (default)
- `de_table` (Dangling Ends):
  - "DNA_DE_Bommarito_2000" (default)
  - "RNA_DE_Turner_2010"

GC Method Options:

- `variant`:
  - "Primer3Plus" (default)
  - "Chester1993"
  - "QuikChange"
  - "Schildkraut1965"
  - "Wetmur1991_MELTING"
  - "Wetmur1991_RNA"
  - "Wetmur1991_RNA/DNA"
  - "vonAhsen2001"

Salt Correction Options:

- `salt_corr_method`:
  - "Schildkraut2010" (default)
  - "Wetmur1991"
  - "SantaLucia1996"
  - "SantaLucia1998-1"
  - "SantaLucia1998-2"
  - "Owczarzy2004"
  - "Owczarzy2008"

Formamide Unit Options:

- `formamide_value_unit$unit`:
  - "percent" (default)
  - "molar"

Other Parameters:

- ambiguous: TRUE/FALSE (default: FALSE)
- shift: Integer value (default: 0)
- dnac_high: Numeric value in nM (default: 25)
- dnac_low: Numeric value in nM (default: 25)
- self_comp: TRUE/FALSE (default: FALSE)
- Na: Millimolar concentration (default: 50)
- K: Millimolar concentration (default: 0)
- Tris: Millimolar concentration (default: 0)
- Mg: Millimolar concentration (default: 0)
- dNTPs: Millimolar concentration (default: 0)
- DMSO: Percent concentration (default: 0)
- dmso_factor: Numeric value (default: 0.75)
- formamide_factor: Numeric value (default: 0.65)
- mismatch: TRUE/FALSE (default: TRUE)

Examples

```
# Calculate Tm using all methods
input_seq <- c("ATGCGATGCG")

# Calculate Tm with specific method parameters
result <- tm_calculate(
  input_seq,
  method = "tm_nn",
  nn_table = "DNA_NN_SantaLucia_2004",
  salt_corr_method = "Owczarzy2008"
)
```

---

tm_gc *Calculate the melting temperature using empirical formulas based on GC content*

Description

Calculate the melting temperature using empirical formulas based on GC content with different options. The function returns a list of sequences with updated Tm attributes and calculation options.

Usage

```
tm_gc(
  gr_seq,
  ambiguous = FALSE,
  userset = NULL,
  variant = c("Primer3Plus", "Chester1993", "QuikChange", "Schildkraut1965",
   "Wetmur1991_MELTING", "Wetmur1991_RNA", "Wetmur1991_RNA/DNA", "vonAhsen2001"),
  Na = 50,
  K = 0,
  Tris = 0,
  Mg = 0,
  dNTPs = 0,
  salt_corr_method = c("Schildkraut2010", "Wetmur1991", "SantaLucia1996",
    "SantaLucia1998-1", "Owczarzy2004", "Owczarzy2008"),
  mismatch = TRUE,
  DMSO = 0,
  formamide_value_unit = list(value = 0, unit = "percent"),
  dmso_factor = 0.75,
  formamide_factor = 0.65
)
```

Arguments

gr_seq	Pre-processed sequence(s) in 5' to 3' direction. This should be the output from to_genomic_ranges() function.
ambiguous	Logical. If TRUE, ambiguous bases are taken into account when computing the G and C content. The function handles various ambiguous bases (S, W, M, K, R, Y, V, H, D, B) by proportionally distributing their contribution to GC content based on their possible nucleotide compositions.
userset	A vector of four coefficient values. Usersets override value sets.
variant	Empirical constants coefficient with 8 variants: - Chester1993: Tm = 69.3 + 0.41(Percentage_GC) - 650/N - QuikChange: Tm = 81.5 + 0.41(Percentage_GC) - 675/N - Percentage_mismatch - Schildkraut1965: Tm = 81.5 + 0.41(Percentage_GC) - 675/N + 16.6 x log[Na+] - Wetmur1991_MELTING: Tm = 81.5 + 0.41(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Wetmur1991_RNA: Tm = 78 + 0.7(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Wetmur1991_RNA/DNA: Tm = 67 + 0.8(Percentage_GC) - 500/N + 16.6 x log([Na+]/(1.0 + 0.7 x [Na+])) - Percentage_mismatch - Primer3Plus: Tm = 81.5 + 0.41(Percentage_GC) - 600/N + 16.6 x log[Na+] - vonAhsen2001: Tm = 77.1 + 0.41(Percentage_GC) - 528/N + 11.7 x log[Na+]
Na	Millimolar concentration of sodium ions. Default: 50
K	Millimolar concentration of potassium ions. Default: 0
Tris	Millimolar concentration of Tris buffer. Default: 0
Mg	Millimolar concentration of magnesium ions. Default: 0
dNTPs	Millimolar concentration of deoxynucleotide triphosphates. Default: 0

`salt_corr_method`
: Salt correction method. Options are: - "Schildkraut2010": Schildkraut & Lifson 1965 - "Wetmur1991": Wetmur 1991 - "SantaLucia1996": SantaLucia 1996 - "SantaLucia1998-1": SantaLucia 1998 (Method 1) - "Owczarzy2004": Owczarzy 2004 - "Owczarzy2008": Owczarzy 2008 Note: "SantaLucia1998-2" is not available for this function.

`mismatch`
: Logical. If TRUE (default), every '.' in the sequence is counted as a mismatch

`DMSO`
: Percent DMSO concentration in the reaction mixture. Default: 0

`formamide_value_unit`
: List containing formamide concentration value and unit. Default: list(value = 0, unit = "percent") - value: Numeric value of formamide concentration - unit: Either "percent" or "molar"

`dmso_factor`
: Coefficient of Tm decreases per percent DMSO. Default: 0.75 (von Ahsen et al. 2001) Other published values are 0.5, 0.6 and 0.675.

`formamide_factor`
: Coefficient of Tm decrease per percent formamide. Default: 0.65 Several papers report factors between 0.6 and 0.72.

Value

Returns a list with two components: - Tm: A list of sequences with updated Tm attributes - Options: A list containing calculation parameters and method information

Author(s)

Junhui Li

References

Marmur J, Doty P. Determination of the base composition of deoxyribonucleic acid from its thermal denaturation temperature. Journal of Molecular Biology, 1962, 5(1):109-118.

Schildkraut C. Dependence of the melting temperature of DNA on salt concentration. Biopolymers, 2010, 3(2):195-208.

Wetmur JG. DNA Probes: Applications of the Principles of Nucleic Acid Hybridization. CRC Critical Reviews in Biochemistry, 1991, 26(3-4):33.

Untergasser A, Cutcutache I, Koressaar T, et al. Primer3–new capabilities and interfaces. Nucleic Acids Research, 2012, 40(15):e115-e115.

von Ahsen N, Wittwer CT, Schutz E, et al. Oligonucleotide melting temperatures under PCR conditions: deoxynucleotide Triphosphate and Dimethyl sulfoxide concentrations with comparison to alternative empirical formulas. Clin Chem 2001, 47:1956-1961.

Examples

```
# Example with multiple sequences
input_seq <- c("ATCGTGCGTAGCAGTACGATCAGTAG", "ATCGTGCGTAGCAGTACGATCAGTAG")
gr_seq <- to_genomic_ranges(input_seq)
out <- tm_gc(gr_seq, ambiguous = TRUE, variant = "Primer3Plus", Na = 50, mismatch = TRUE)
```

```
out
out$Options
```

---

tm_nn *Calculate melting temperature using nearest neighbor thermodynamics*

Description

Calculate melting temperature using nearest neighbor thermodynamics. The function checks if all sequence combinations in the input sequence are present in the thermodynamic parameter tables before performing calculations.

Usage

```
tm_nn(
  gr_seq,
  ambiguous = FALSE,
  shift = 0,
 nn_table = c("DNA_NN_SantaLucia_2004", "DNA_NN_Breslauer_1986", "DNA_NN_Sugimoto_1996",
   "DNA_NN_Allawi_1998", "RNA_NN_Freier_1986", "RNA_NN_Xia_1998", "RNA_NN_Chen_2012",
     "RNA_DNA_NN_Sugimoto_1995"),
  tmm_table = "DNA_TMM_Bommarito_2000",
  imm_table = "DNA_IMM_Peyret_1999",
  de_table = c("DNA_DE_Bommarito_2000", "RNA_DE_Turner_2010"),
  dnac_high = 25,
  dnac_low = 25,
  self_comp = FALSE,
  Na = 50,
  K = 0,
  Tris = 0,
  Mg = 0,
  dNTPs = 0,
  salt_corr_method = c("Schildkraut2010", "Wetmur1991", "SantaLucia1996",
    "SantaLucia1998-1", "Owczarzy2004", "Owczarzy2008"),
  DMSO = 0,
  formamide_value_unit = list(value = 0, unit = "percent"),
  dmso_factor = 0.75,
  formamide_factor = 0.65
)
```

Arguments

gr_seq	Pre-processed sequence(s) in 5' to 3' direction. This should be the output from to_genomic_ranges() function.

`ambiguous` Logical value controlling how ambiguous bases are handled: - TRUE: Ambiguous bases (e.g., N, R, Y) are included in calculations - FALSE (default): Ambiguous bases are excluded from calculations

`shift` Integer value controlling the alignment offset between primer and template sequences. Visual representation of different shift values:

shift = 0 (default): Primer: 5' ATGCG 3' Template: 3' TACGC 5'

shift = -1: Primer: 5' ATGCG 3' Template: 3' TACGC 5' ^

shift = 1: Primer: 5' ATGCG 3' Template: 3' TACGC 5' ^

The shift parameter is necessary when: - Sequences have different lengths - Dangling ends are required - Specific alignment positions are needed

`nn_table` Thermodynamic nearest-neighbor parameters for different nucleic acid hybridizations. Eight parameter sets are available, organized by hybridization type:

DNA/DNA hybridizations: - "DNA_NN_Breslauer_1986": Original DNA/DNA parameters - "DNA_NN_Sugimoto_1996": Improved DNA/DNA parameters - "DNA_NN_Allawi_1998": DNA/DNA parameters with internal mismatch corrections - "DNA_NN_SantaLucia_2004": Updated DNA/DNA parameters

RNA/RNA hybridizations: - "RNA_NN_Freier_1986": Original RNA/RNA parameters - "RNA_NN_Xia_1998": Improved RNA/RNA parameters - "RNA_NN_Chen_2012": Updated RNA/RNA parameters with GU pair corrections

RNA/DNA hybridizations: - "RNA_DNA_NN_Sugimoto_1995": RNA/DNA hybridization parameters

`tmm_table` Thermodynamic parameters for terminal mismatches. Default: "DNA_TMM_Bommarito_2000" These parameters account for mismatches at the ends of the duplex.

`imm_table` Thermodynamic parameters for internal mismatches. Default: "DNA_IMM_Peyret_1999" These parameters account for mismatches within the duplex, including inosine mismatches.

`de_table` Thermodynamic parameters for dangling ends. Default: "DNA_DE_Bommarito_2000" Available options: - "DNA_DE_Bommarito_2000": Parameters for DNA dangling ends - "RNA_DE_Turner_2010": Parameters for RNA dangling ends

`dnac_high` Concentration of the higher concentrated strand in nM. Default: 25 Typically this is the primer (for PCR) or the probe concentration.

`dnac_low` Concentration of the lower concentrated strand in nM. Default: 25 This is typically the template concentration.

`self_comp` Logical value indicating if the sequence is self-complementary: - TRUE: Sequence can bind to itself, dnac_low is ignored - FALSE (default): Sequence binds to a different complementary sequence

`Na` Millimolar concentration of sodium ions. Default: 50

`K` Millimolar concentration of potassium ions. Default: 0

`Tris` Millimolar concentration of Tris buffer. Default: 0

`Mg` Millimolar concentration of magnesium ions. Default: 0

`dNTPs` Millimolar concentration of deoxynucleotide triphosphates. Default: 0

`salt_corr_method` Method for calculating salt concentration corrections to the melting temperature. Available options: - "Schildkraut2010": Updated salt correction method

- "Wetmur1991": Classic salt correction method - "SantaLucia1996": DNA-specific salt correction - "SantaLucia1998-1": Improved DNA salt correction - "SantaLucia1998-2": Alternative DNA salt correction - "Owczarzy2004": Comprehensive salt correction - "Owczarzy2008": Updated comprehensive salt correction Note: Setting to NA disables salt correction

`DMSO` Percent DMSO concentration in the reaction mixture. Default: 0 DMSO can lower the melting temperature of nucleic acid duplexes.

`formamide_value_unit`
A list containing formamide concentration information: - value: numeric value of formamide concentration - unit: character string specifying the unit ("percent" or "molar") Default: list(value=0, unit="percent")

`dmso_factor` Coefficient of melting temperature (Tm) decrease per percent DMSO. Default: 0.75 (von Ahsen N, 2001, PMID:11673362) Other published values: 0.5, 0.6, 0.675

`formamide_factor`
Coefficient of melting temperature (Tm) decrease per percent formamide. Default: 0.65 Literature reports values ranging from 0.6 to 0.72

Details

DNA_NN_Breslauer_1986: Breslauer K J (1986) <doi:10.1073/pnas.83.11.3746>

DNA_NN_Sugimoto_1996: Sugimoto N (1996) <doi:10.1093/nar/24.22.4501>

DNA_NN_Allawi_1998: Allawi H (1998) <doi:10.1093/nar/26.11.2694>

DNA_NN_SantaLucia_2004: SantaLucia J (2004) <doi:10.1146/annurev.biophys.32.110601.141800>

RNA_NN_Freier_1986: Freier S (1986) <doi:10.1073/pnas.83.24.9373>

RNA_NN_Xia_1998: Xia T (1998) <doi:10.1021/bi9809425>

RNA_NN_Chen_2012: Chen JL (2012) <doi:10.1021/bi3002709>

RNA_DNA_NN_Sugimoto_1995: Sugimoto N (1995)<doi:10.1016/S0048-9697(98)00088-6>

DNA_TMM_Bommarito_2000: Bommarito S (2000) <doi:10.1093/nar/28.9.1929>

DNA_IMM_Peyret_1999: Peyret N (1999) <doi:10.1021/bi9825091> & Allawi H T (1997) <doi:10.1021/bi962590c> & Santalucia N (2005) <doi:10.1093/nar/gki918>

DNA_DE_Bommarito_2000: Bommarito S (2000) <doi:10.1093/nar/28.9.1929>

RNA_DE_Turner_2010: Turner D H (2010) <doi:10.1093/nar/gkp892>

Author(s)

Junhui Li

References

Breslauer K J , Frank R , Blocker H , et al. Predicting DNA duplex stability from the base sequence.[J]. Proceedings of the National Academy of Sciences, 1986, 83(11):3746-3750.

Sugimoto N , Nakano S , Yoneyama M , et al. Improved Thermodynamic Parameters and Helix Initiation Factor to Predict Stability of DNA Duplexes[J]. Nucleic Acids Research, 1996, 24(22):4501-5.

Allawi, H. Thermodynamics of internal C.T mismatches in DNA[J]. Nucleic Acids Research, 1998, 26(11):2694-2701.

Hicks L D , Santalucia J . The thermodynamics of DNA structural motifs.[J]. Annual Review of Biophysics & Biomolecular Structure, 2004, 33(1):415-440.

Freier S M , Kierzek R , Jaeger J A , et al. Improved free-energy parameters for predictions of RNA duplex stability.[J]. Proceedings of the National Academy of Sciences, 1986, 83(24):9373-9377.

Xia T , Santalucia , J , Burkard M E , et al. Thermodynamic Parameters for an Expanded Nearest-Neighbor Model for Formation of RNA Duplexes with Watson-Crick Base Pairs,[J]. Biochemistry, 1998, 37(42):14719-14735.

Chen J L , Dishler A L , Kennedy S D , et al. Testing the Nearest Neighbor Model for Canonical RNA Base Pairs: Revision of GU Parameters[J]. Biochemistry, 2012, 51(16):3508-3522.

Bommarito S, Peyret N, Jr S L. Thermodynamic parameters for DNA sequences with dangling ends[J]. Nucleic Acids Research, 2000, 28(9):1929-1934.

Turner D H , Mathews D H . NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure[J]. Nucleic Acids Research, 2010, 38(Database issue):D280-D282.

Sugimoto N , Nakano S I , Katoh M , et al. Thermodynamic Parameters To Predict Stability of RNA/DNA Hybrid Duplexes[J]. Biochemistry, 1995, 34(35):11211-11216.

Allawi H, SantaLucia J: Thermodynamics and NMR of internal G-T mismatches in DNA. Biochemistry 1997, 36:10581-10594.

Santalucia N E W J . Nearest-neighbor thermodynamics of deoxyinosine pairs in DNA duplexes[J]. Nucleic Acids Research, 2005, 33(19):6258-67.

Peyret N , Seneviratne P A , Allawi H T , et al. Nearest-Neighbor Thermodynamics and NMR of DNA Sequences with Internal A-A, C-C, G-G, and T-T Mismatches, [J]. Biochemistry, 1999, 38(12):3468-3477.

Examples

```
input_seq <- c("AAAATTTTTTTCCCCCCCCCCCCCCCGGGGGGGGGGGGGTGTGCGCTGC",
"AAAATTTTTTTCCCCCCCCCCCCCCCGGGGGGGGGGGGGTGTGCGCTGC")
gr_seq <- to_genomic_ranges(input_seq)
out <- tm_nn(gr_seq, Na=50)
out
out$tm_nn$Options
```

`tm_wallace` *Calculate the melting temperature using the 'Wallace rule'*

Description

The Wallace rule is often used as rule of thumb for approximate melting temperature calculations for primers with 14 to 20 nt length.

Usage

```
tm_wallace(gr_seq, ambiguous = FALSE)
```

Arguments

gr_seq	Pre-processed sequence(s) in 5' to 3' direction. This should be the output from to_genomic_ranges() function.
ambiguous	Ambiguous bases are taken into account to compute the G and C content when ambiguous is TRUE.

Value

Returns a list of sequences with updated Tm attributes

Author(s)

Junhui Li

References

Thein S L , Lynch J R , Weatherall D J , et al. DIRECT DETECTION OF HAEMOGLOBIN E WITH SYNTHETIC OLIGONUCLEOTIDES[J]. The Lancet, 1986, 327(8472):93.

Examples

```
input_seq = c('acgtTGCAATGCCGTAWSDBSY','acgtTGCCCCGGCCGCGCCGTAWSDBSY') #for wallace rule
gr_seq <- to_genomic_ranges(input_seq)
out <- tm_wallace(gr_seq, ambiguous = TRUE)
out
out$Options
```

---

to_genomic_ranges *Convert input file into a GenomicRanges Object*

Description

This function processes a vector of sequences string, a FASTA file, or a character vector with genomic coordinates into a GenomicRanges object, optionally including complementary sequences. sequence names are parsed based on their format: - If names have this pattern "chr:start-end:strand:species[:name]" (e.g., "chr1:1-5:+:seq_1"), parse components into seqnames, ranges, strand, and name. - If names have this pattern "chr:start-end:strand" (e.g., "chr1:1-5:+"), parse components into seqnames, ranges, and strand. - If names have this pattern "chr:start-end" (e.g., "chr1:1-5"), parse components into seqnames and ranges. - If no names are provided, use default values: seqnames = "chr1", start = 1, width = sequence length, strand = "*", name = "1", etc. Complementary sequences are either provided or automatically generated.

Usage

```
to_genomic_ranges(input_seq, complement_seq = NULL)
```

Arguments

input_seq	Input sequence(s) in 5' to 3' direction. Can be provided as either: - A character string (e.g., c("ATGCG", "GCTAG")) - A path to a FASTA file containing the sequence(s) - A character vector where each element is a string in the format "chr:start-end:strand:species" #' (e.g., "chr1:100-200:+:BSgenome.Hsapiens.UCSC.hg38"). Strand is "+" for positive or "-" for negative. - chr: Chromosome ID - start: Start position - end: End position - strand: positive or negative strand - species: Species name for reference genome (e.g., "BSgenome.Hsapiens.UCSC.hg38"), see `BSgenome::available.genomes()` for all available genomes. please make sure the genome package is installed, otherwise the function will stop.
complement_seq	Optional complementary sequences. If NULL, complementary sequences will be auto-generated. otherwise, the complementary sequences will be used as metadata. Can be provided as format of input_seq.

Value

A GenomicRanges object with seqnames, ranges, strand, name, sequence, Complement, and Tm as metadata.

Author(s)

Junhui Li

Examples

```
# Using a character vector with auto-generated complementary sequences
seqs <- c("ATGCG", "GCTAG")
names(seqs) <- c("chr1:1-5:+:seq_1", "chr2:1-5:+")
gr <- to_genomic_ranges(seqs)
gr

# Using a character vector with provided complementary sequences
seqs <- c("ATGCG", "GCTAG")
comp_seqs <- c("TACGC", "CGTA")
gr <- to_genomic_ranges(seqs, comp_seqs)
gr

# Using a FASTA file
gr <- to_genomic_ranges(system.file("extdata", "example1.fasta", package = "TmCalculator"))
## Not run:
# Using a character vector with genomic coordinates
seqs <- c(
  "chr1:1898000-1898050:+:BSgenome.Hsapiens.UCSC.hg38",
  "chr2:2563000-2563050:-:BSgenome.Hsapiens.UCSC.hg38"
)
gr <- to_genomic_ranges(seqs)
```

```
gr

## End(Not run)
```

`vec_to_genomic_ranges` *Convert sequence strings to GenomicRanges object*

Description

This function converts sequence strings to a GenomicRanges object, handling both named and unnamed sequences. It can also process complementary sequences if provided. sequence names can be in the format ">chr2:1-10:+:seq2" which will be parsed into chromosome, position, strand, and name components.

Usage

```
vec_to_genomic_ranges(input_seq)
```

Arguments

`input_seq`	A character vector of sequences. If named with format "chr2:1-10:[+|-]:[seq_name]" the name will be parsed into GRanges components.

Value

A GenomicRanges object containing: - GRanges information (seqnames, ranges, strand) - sequence data - Complementary sequences - Names from input or auto-generated

Examples

```
# Example with named sequences in GRanges format
seqs <- c("ATGCG", "GCTAG")
names(seqs) <- c("chr1:1111-1115:+:seq1", "chr2:1221-1225:+")
gr <- vec_to_genomic_ranges(seqs)

# Example with unnamed sequences
seqs <- c("ATGCG", "GCTAG")
gr <- vec_to_genomic_ranges(seqs)
```

Index