Package 'baseq'

May 3, 2023

Title Basic Sequence Processing Tool for Biological Data

Version 0.1.4

Description Primarily created as an easy and understanding way to do basic sequences surrounding the central dogma of molecular biology.

License GPL-3

URL https://github.com/ambuvjyn/baseq

BugReports https://github.com/ambuvjyn/baseq/issues

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Ambu Vijayan [aut, cre] (<https://orcid.org/0000-0001-8924-5685>), J. Sreekumar [aut] (<https://orcid.org/0000-0002-4253-6378>, Principal Scientist, ICAR - Central Tuber Crops Research Institute)

Maintainer Ambu Vijayan <ambuvjyn@gmail.com>

Repository CRAN

Date/Publication 2023-05-03 11:40:02 UTC

R topics documented:

`clean_DNA_file` *Clean DNA file*

Description

This function reads a multi FASTA file containing DNA sequences, removes any characters other than A, T, G, and C, and writes the cleaned sequences to a new multi FASTA file. The output file name is generated from the input file name with the suffix '_clean.fasta'.

Usage

```
clean_DNA_file(input_file, output_dir = "")
```

Arguments

`input_file`	The name of the input multi FASTA file.
`output_dir`	The directory where the output file will be saved. If not given, the output file will be saved in the same directory as the input file.

Value

A character string specifying the path to the output FASTA file.

Examples

```
#sample_file_path_three <- system.file("extdata", "sample2_fa.fasta", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path_three))
#file.copy(sample_file_path_three, temp_file_path, overwrite = TRUE)
#clean_DNA_file(temp_file_path, output_dir = tempdir)

# Write to working directory
```

```
# clean_DNA_file(file_path)

# Write to custom directory
# clean_DNA_file(file_path, output_dir = "/path/to/directory/")
```

`clean_DNA_sequence` *Clean DNA sequence*

Description

This function takes a DNA sequence as input and removes any characters other than A, C, G, and T.

Usage

```
clean_DNA_sequence(sequence)
```

Arguments

`sequence`	DNA sequence to be cleaned

Value

Cleaned DNA sequence

Examples

```
clean_DNA_sequence("ATGTCGTAGCTAGCTN")
# Output: "ATGTCGTAGCTAGCT"
```

`clean_RNA_file` *Clean RNA file*

Description

This function reads a multi FASTA file containing RNA sequences, removes any characters other than A, T, G, and C, and writes the cleaned sequences to a new multi FASTA file. The output file name is generated from the input file name with the suffix '_clean.fasta'.

Usage

```
clean_RNA_file(input_file, output_dir = "")
```

Arguments

input_file	The name of the input multi FASTA file.
output_dir	The directory where the output file will be saved. If not given, the output file will be saved in the same directory as the input file.

Value

A character string specifying the path to the output FASTA file.

Examples

```
#sample_file_path_three <- system.file("extdata", "sample2_fa.fasta", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path_three))
#file.copy(sample_file_path_three, temp_file_path, overwrite = TRUE)
#clean_RNA_file(temp_file_path, output_dir = tempdir)

# Write to working directory
# clean_RNA_file(file_path)

# Write to custom directory
# clean_RNA_file(file_path, output_dir = "/path/to/directory/")
```

---

clean_RNA_sequence *Clean RNA sequence*

Description

This function takes a RNA sequence as input and removes any characters other than A, C, G, and T.

Usage

```
clean_RNA_sequence(sequence)
```

Arguments

sequence	RNA sequence to be cleaned

Value

Cleaned RNA sequence

Examples

```
clean_RNA_sequence("AUGUCGTAGCTAGCTN")
# Output: "AUGUCGAGCAGC"
```

clean_sequence *Clean DNA or RNA sequence*

Description

This function takes a DNA or RNA sequence as input and removes any characters that are not A, C, G, T (for DNA) or A, C, G, U (for RNA).

Usage

```
clean_sequence(sequence, type = "DNA")
```

Arguments

sequence	A character string containing the DNA or RNA sequence to be cleaned.
type	A character string indicating the type of sequence. The default is "DNA". If set to "RNA", the function will remove any characters that are not A, C, G, U.

Value

A character string containing the cleaned DNA or RNA sequence.

Examples

```
clean_sequence("atgcNnRYMK") # Returns "ATGC"
clean_sequence("auggcuuNnRYMK", type = "RNA") # Returns "AUGGCUU"
```

count_bases *Count the number of A's, C's, G's, and T's in a DNA sequence*

Description

This function takes a single argument, a DNA sequence as a character string, and counts the number of A's, C's, G's, and T's in the sequence. The counts are returned as a named vector.

Usage

```
count_bases(sequence)
```

Arguments

sequence	a character string containing a DNA sequence

Value

a named integer vector containing the counts of A's, C's, G's, and T's

Examples

```
sequence <- "ATCGAGCTAGCTAGCTAGCTAGCT"
count_bases(sequence)
# A  C  G  T
# 6  6  6  6
```

`count_seq_pattern` *Count frequency of a pattern in a sequence*

Description

This function counts the frequency of a specific character or pattern in a given sequence.

Usage

```
count_seq_pattern(seq, pattern)
```

Arguments

`seq`	A character vector representing the sequence to count the pattern in.
`pattern`	A character string representing the pattern to count in the sequence.

Value

An integer representing the count of the pattern in the sequence.

Examples

```
seq <- "ATGGTGCTCCGTGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCTACGTAG"
count_seq_pattern(seq, "CG")
# [1] 31
```

`dna_to_protein` *Translation of a DNA sequence*

Description

This function takes a DNA sequence as input and translates it in all six reading frames.

Usage

```
dna_to_protein(sequence)
```

Arguments

`sequence`	A character string representing a DNA sequence.

Value

A list of character strings representing the translated protein sequences in all six frames.

Examples

```
sequence <- "ATCGAGCTAGCTAGCTAGCTAGCT"
dna_to_protein(sequence)
# Returns a list containing the translated protein sequences in all six frames:
# $`Frame F1`
# [1] "IELAS"
#
# $`Frame F2`
# [1] "SS"
#
# $`Frame F3`
# [1] "RAS"
#
# $`Frame R1`
# [1] "S"
#
# $`Frame R2`
# [1] "AS"
#
# $`Frame R3`
# [1] "LAS"
```

---

dna_to_rna *Transcription of a DNA sequence*

Description

This function takes a DNA sequence as input and returns its RNA transcript.

Usage

```
dna_to_rna(sequence)
```

Arguments

sequence	A character string representing a DNA sequence.

Value

A character string representing the RNA transcript of the input DNA sequence.

Examples

```
sequence <- "ATCGAGCTAGCTAGCTAGCTAGCT"
dna_to_rna(sequence)
# Returns "AUCGAGCUAGCUAGCUAGCUAGCU"
```

fastq_to_fasta *Convert a FASTQ file to a FASTA file*

Description

This function converts a FASTQ file to a FASTA file. The output file has the same name as the input FASTQ file, but with the extension changed to `.fasta`. This function removes the `@` symbol at the beginning of FASTQ sequence names and replaces it with the `>` symbol for the FASTA format.

Usage

```
fastq_to_fasta(fastq_file)
```

Arguments

`fastq_file`	A character string specifying the path to the input FASTQ file.

Value

A character string specifying the path to the output FASTA file.

Examples

```
#sample_file_path_two <- system.file("extdata", "sample_fq.fastq", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path_two))
#file.copy(sample_file_path_two, temp_file_path, overwrite = TRUE)
#fastq_to_fasta(temp_file_path)

# Output: "path/to/Temp/tempfoldername/sample_fq.fasta"
```

gc_content *Calculate GC content of a DNA sequence*

Description

Calculates the percentage of nucleotides in a DNA sequence that are either guanine (G) or cytosine (C).

Usage

```
gc_content(sequence)
```

Arguments

`sequence`	A character string containing the DNA sequence.

Value

A numeric value representing the percentage of nucleotides in the sequence that are G or C.

Examples

```
sequence <- "ATCGAGCTAGCTAGCTAGCTAGCT"
gc_content(sequence)
50
```

`gc_content_file` *GC content of sequences in a multi FASTA file*

Description

Function to calculate GC content of sequences in a multi FASTA file and write the results to a new FASTA file

Usage

```
gc_content_file(input_file)
```

Arguments

`input_file`	A string indicating the path and name of the input multi-FASTA file

Examples

```
#sample_file_path <- system.file("extdata", "sample_fa.fasta", package = "baseq")
#clean_DNA_file(sample_file_path)
```

`read.fasta_to_df` *Read a fasta file into a dataframe and assign to the environment*

Description

This function reads a fasta file and creates a dataframe with two columns: Header and Sequence. The dataframe is then assigned to the environment with the name same as the fasta file name but without the .fasta extension.

Usage

```
read.fasta_to_df(fasta_file)
```

Arguments

`fasta_file`	The path to the fasta file to be read.

Value

This function does not return anything. It assigns the resulting dataframe to the environment.

Examples

```
# Read in sequences from a FASTA file

sample_file_path <- system.file("extdata", "sample_fa.fasta", package = "baseq")
read.fasta_to_df(sample_file_path)
```

`read.fasta_to_list` *Read a fasta file into a list and assign to the environment*

Description

This function reads a fasta file and creates a list with two columns: Header and Sequence. The list is then assigned to the environment with the name same as the fasta file name but without the .fasta extension.

Usage

```
read.fasta_to_list(fasta_file)
```

Arguments

`fasta_file`	The path to the fasta file to be read.

Value

This function does not return anything. It assigns the resulting list to the environment.

Examples

```
# Read in sequences from a FASTA file

sample_file_path <- system.file("extdata", "sample_fa.fasta", package = "baseq")
read.fasta_to_list(sample_file_path)

# Access a specific sequence by name
# sample_fa[["sample_seq.1"]]
```

read.fastq_to_df *Read a Fastq file and store it as a dataframe*

Description

This function reads a Fastq file and stores it as a dataframe with three columns: Header, Sequence, and QualityScore.

Usage

```
read.fastq_to_df(fastq_file)
```

Arguments

fastq_file	A character string specifying the path to the Fastq file to be read.

Value

This function returns a dataframe with three columns: Header, Sequence, and QualityScore.

Examples

```
# Read in sequences from a FASTQ file

#sample_file_path_two <- system.file("extdata", "sample_fq.fastq", package = "baseq")
#read.fastq_to_df(sample_file_path_two)
```

read.fastq_to_list *Read a Fastq file and store it as a list*

Description

This function reads a Fastq file and stores it as a list with three columns: Header, Sequence, and QualityScore.

Usage

```
read.fastq_to_list(fastq_file)
```

Arguments

fastq_file	A character string specifying the path to the Fastq file to be read.

Value

This function returns a list with three columns: Header, Sequence, and QualityScore.

Examples

```
# Read in sequences from a FASTQ file

sample_file_path_two <- system.file("extdata", "sample_fq.fastq", package = "baseq")
read.fastq_to_list(sample_file_path_two)
```

---

reverse_complement *Generate Reverse Complement of DNA sequence*

Description

Given a DNA sequence, the function generates the reverse complement of the sequence and returns it.

Usage

```
reverse_complement(sequence)
```

Arguments

sequence	A character string containing the DNA sequence to be reversed and complemented

Value

A character string containing the reverse complement of the input DNA sequence

Examples

```
sequence <- "ATCGAGCTAGCTAGCTAGCTAGCT"
reverse_complement(sequence)
# [1] "AGCTAGCTAGCTAGCTAGCTCGAT"
```

rna_reverse_complement *Generate Reverse Complement of DNA sequence*

Description

Given a DNA sequence, the function generates the reverse complement of the sequence and returns it.

Usage

```
rna_reverse_complement(sequence)
```

Arguments

sequence	A character string containing the DNA sequence to be reversed and complemented

Value

A character string containing the reverse complement of the input DNA sequence

Examples

```
sequence <- "AUCGAGCUAGCUAGCUAGCUAGCU"
rna_reverse_complement(sequence)
# [1] "AGCUAGCUAGCUAGCUAGCUCGAU"
```

rna_to_dna *Reverse Transcription of a RNA sequence*

Description

This function takes a RNA sequence as input and returns its DNA transcript.

Usage

```
rna_to_dna(sequence)
```

Arguments

sequence	A character string representing a RNA sequence.

Value

A character string representing the RNA transcript of the input RNA sequence.

Examples

```
sequence <- "AUCGAGCUAGCUAGCUAGCUAGCU"
rna_to_dna(sequence)
# Returns "ATCGAGCTAGCTAGCTAGCTAGCT"
```

`rna_to_protein` *Translation of a RNA sequence*

Description

This function takes a RNA sequence as input and translates it in all six reading frames.

Usage

```
rna_to_protein(sequence)
```

Arguments

`sequence`	A character string representing a RNA sequence.

Value

A list of character strings representing the translated protein sequences in all six frames.

Examples

```
sequence <- "AUCGAGCUAGCUAGCUAGCUAGCU"
rna_to_protein(sequence)
# Returns a list containing the translated protein sequences in all six frames:
# $`Frame F1`
# [1] "IELAS"
#
# $`Frame F2`
# [1] "SS"
#
# $`Frame F3`
# [1] "RAS"
#
# $`Frame R1`
# [1] "S"
#
# $`Frame R2`
# [1] "AS"
#
# $`Frame R3`
# [1] "LAS"
```

`write.df_to_fasta` *Write a data frame to a fasta file*

Description

This function writes a data frame to a fasta file with the same name as the data frame. The data frame is assumed to have two columns, "Header" and "Sequence", which represent the header and sequence lines of each fasta record, respectively.

Usage

```
write.df_to_fasta(df, output_dir = getwd())
```

Arguments

`df`	A data frame containing fasta records with "Header" and "Sequence" columns.
`output_dir`	The directory path where the output file should be written. If not provided, the working directory will be used.

Value

This function does not return a value, but writes a fasta file to the specified output directory or the working directory.

Examples

```
#sample_file_path <- system.file("extdata", "sample_fa.fasta", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path))
#file.copy(sample_file_path, temp_file_path, overwrite = TRUE)
#read.fasta_to_df(sample_file_path)
#write.df_to_fasta(sample_fa, output_dir = tempdir)

# Write to working directory
# write.df_to_fasta(sample_fa)

# Write to custom directory
# write.df_to_fasta(sample_fa, output_dir = "/path/to/directory/")
```

write.df_to_fastq *Write a FASTQ file from a dataframe of reads*

Description

Write a FASTQ file from a dataframe of reads

Usage

```
write.df_to_fastq(df, output_dir = getwd())
```

Arguments

df	A dataframe containing reads in the format "Header", "Sequence", and "QualityScore".
output_dir	An optional argument specifying the directory where the FASTQ file should be saved. If not specified, the file will be saved in the working directory.

Value

A FASTQ file with the same name as the input dataframe.

Examples

```
#sample_file_path_two <- system.file("extdata", "sample_fq.fastq", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path_two))
#file.copy(sample_file_path_two, temp_file_path, overwrite = TRUE)
#read.fastq_to_df(sample_file_path_two)
#write.df_to_fastq(sample_fq, output_dir = tempdir)

# Write to working directory
# write.df_to_fastq(sample_fq)

# Write to custom directory
# write.df_to_fastq(sample_fq, output_dir = "/path/to/directory/")
```

write.dna_to_rna *Convert DNA file to RNA file*

Description

This function reads a multi FASTA file containing DNA sequences, converts each DNA sequence to RNA sequence, and writes the RNA sequences to a new multi FASTA file. The output file name is generated from the input file name with the suffix '_rna.fasta'.

Usage

```
write.dna_to_rna(input_file, output_dir = "")
```

Arguments

input_file	The name of the input multi FASTA file.
output_dir	The directory where the output file will be saved. If not given, the output file will be saved in the same directory as the input file.

Value

A character string specifying the path to the output FASTA file.

Examples

```
#sample_file_path <- system.file("extdata", "sample_fa.fasta", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path))
#file.copy(sample_file_path, temp_file_path, overwrite = TRUE)
#write.dna_to_rna(temp_file_path, output_dir = tempdir)

# Write to working directory
# write.dna_to_rna(file_path)

# Write to custom directory
# write.dna_to_rna(file_path, output_dir = "/path/to/directory/")
```

---

write.list_to_fasta *Write a list of sequences to a FASTA file*

Description

This function takes a list of sequences and writes them to a FASTA file. The name of the list is used as the base name for the output file with the .fasta extension. Each sequence in the list is written to the output file in FASTA format with the sequence name as the header.

Usage

```
write.list_to_fasta(sequence_list, output_dir = getwd())
```

Arguments

sequence_list	A list of sequences where each element of the list is a character string representing a single sequence.
output_dir	The directory path where the output file should be written. If not provided, the working directory will be used.

Examples

```
sequences <- list("ACGT", "ATCG")
tempdir <- tempdir()
write.list_to_fasta(sequences, output_dir = tempdir)

# Write to working directory
# write.list_to_fasta(sequences)

# Write to custom directory
# write.list_to_fasta(sequences, output_dir = "/path/to/directory/")
```

---

write.list_to_fastq *Write a list of sequence_bases and quality scores to a FASTQ file*

Description

This function takes a list of sequence_bases and quality scores and writes them to a FASTQ file. The name of the list is used as the base name for the output file with the .fastq extension. Each sequence in the list is written to the output file in FASTQ format with the sequence name as the header and the quality scores on the following line.

Usage

```
write.list_to_fastq(sequence_list, output_dir = getwd())
```

Arguments

sequence_list	A list of sequence_bases where each element of the list is a named list containing "Sequence" and "QualityScore" elements.
output_dir	The directory path where the output file should be written. If not provided, the working directory will be used.

Examples

```
sequence_bases <- list("ACGT", "ATCG")
quality_scores <- list("IIII", "JJJJ")
sequences <- list(seq1=list(Sequence=sequence_bases[[1]], QualityScore=quality_scores[[1]]),
                  seq2=list(Sequence=sequence_bases[[2]], QualityScore=quality_scores[[2]]))
tempdir <- tempdir()
write.list_to_fastq(sequences, output_dir = tempdir)

# Write to working directory
# write.list_to_fastq(sequences)

# Write to custom directory
# write.list_to_fastq(sequences, output_dir = "/path/to/directory/")
```

write.rna_to_dna *Convert RNA file to DNA file*

Description

This function reads a multi FASTA file containing RNA sequences, converts each RNA sequence to DNA sequence, and writes the DNA sequences to a new multi FASTA file. The output file name is generated from the input file name with the suffix '_rna.fasta'.

Usage

```
write.rna_to_dna(input_file, output_dir = "")
```

Arguments

input_file	The name of the input multi FASTA file.
output_dir	The directory where the output file will be saved. If not given, the output file will be saved in the same directory as the input file.

Value

A character string specifying the path to the output FASTA file.

Examples

```
#sample_file_path <- system.file("extdata", "sample3_fa.fasta", package = "baseq")
#tempdir <- tempdir()
#temp_file_path <- file.path(tempdir, basename(sample_file_path))
#file.copy(sample_file_path, temp_file_path, overwrite = TRUE)
#write.rna_to_dna(temp_file_path, output_dir = tempdir)

# Write to working directory
# write.rna_to_dna(file_path)

# Write to custom directory
# write.rna_to_dna(file_path, output_dir = "/path/to/directory/")
```

Index