Package ‘CSESA’

July 21, 2025

Type Package

Title CRISPR-Based Salmonella Enterica Serotype Analyzer

Version 1.2.0

Description Salmonella enterica is a major cause of bacterial food-borne disease worldwide. Serotype identification is the most commonly used typing method to characterize Salmonella isolates. However, experimental serotyping needs great cost on manpower and resources. Recently, we found that the newly incorporated spacer in the clustered regularly interspaced short palindromic repeat (CRISPR) could serve as an effective marker for typing of Salmonella. It was further revealed by Li et. al (2014) <doi:10.1128/JCM.00696-14> that recognized types based on the combination of two newly incorporated spacer in both CRISPR loci showed high accordance with serotypes. Here, we developed an R package 'CSESA' to predict the serotype based on this finding. Considering it’s time saving and of high accuracy, we recommend to predict the serotypes of unknown Salmonella isolates using 'CSESA' before doing the traditional serotyping.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat

Imports Biostrings

NeedsCompilation no

Author Xia Zhang [aut, cre],
Lang Yang [aut]

Maintainer Xia Zhang <zhangxia9403@gmail.com>

Repository CRAN

Date/Publication 2019-03-03 16:40:02 UTC

Contents

CSESA *CSESA (CRISPR-based Salmonella enterica Serotype Analyzer).*

Description

The main function in CSESA package.

Usage

```
CSESA(in.file1 = NULL, in.file2 = NULL, out.file = NULL,
  method = c("PCR", "WGS"))
```

Arguments

in.file1	The first input file, the default value is NULL.
in.file2	The second input file (optional), the default value is NULL.
out.file	Into which results will be saved if this value is set. Otherwise results will be displayed on the screen.
method	The method to handle the input file(s), which can be set as "PCR" or "WGS". Choose "PCR" if the CRISPR sequence(s) from PCR amplification is entered, and choose "WGS" when entering the whole genome assembly of a Salmonella isolate.

Note

If you use the "WGS" method, please make sure you have installed the BLAST software and included it within the working path.

Examples

```
 CSESA(system.file("extdata", "sequence_CRIPSR1.fasta", package = "CSESA"),
 system.file("extdata", "sequence_CRIPSR2.fasta", package = "CSESA"), method = "PCR")
CSESA(system.file("extdata", "sequence_CRIPSR1.fasta", package = "CSESA"), method = "PCR")
 CSESA(system.file("extdata", "Salmonella_whole_genome_assembly.fasta",
 package = "CSESA"), method = "WGS")
```

FindSerotype — *Find the serotype based on the analysis of the new spacers.*

Description

Find the serotype based on the analysis of the new spacers.

Usage

```
FindSerotype(csesa1 = NA, csesa2 = NA)
```

Arguments

`csesa1`	The new spacer of the first sequence.
`csesa2`	The new spacer of the second sequence.

Value

The data frame which represents the serotype.

GetAllNewSpacers — *Get the new spacers from the molecular sequence and its reverse complement.*

Description

Get the new spacers from the molecular sequence and its reverse complement.

Usage

```
GetAllNewSpacers(molecular.seq = NULL)
```

Arguments

`molecular.seq`	The molecular sequence.

Value

The vector of the new spacers, which is extracted from the molecular sequence and its reverse complement.

Note

If there doesn't exist any new spacer, the function would return NA.

`GetNewSpacer` *Get the new spacer from the molecular sequence.*

Description

Get the new spacer from the molecular sequence.

Usage

```
GetNewSpacer(molecular.seq = NULL)
```

Arguments

`molecular.seq`	The molecular sequence.

Value

The new spacer sequence as a string.

Examples

```
GetNewSpacer("AGAGGCGGACCGAAAAACCGTTTTCAGCCAACGTAT")
```

`GetNewSpacerCode` *Get the new spacer from the molecular sequence and map it to the code.*

Description

Get the new spacer from the molecular sequence and map it to the code.

Usage

```
GetNewSpacerCode(molecular.seq = NULL)
```

Arguments

`molecular.seq`	The molecular sequence.

Value

The new spacer code as a string.

`GetReverseComplement` *Return the reverse complement of the sequence.*

Description

Return the reverse complement of the sequence.

Usage

```
GetReverseComplement(x)
```

Arguments

`x`	The input sequence.

Value

The reverse complement sequence as a string.

`GetStr` *Get the information string from the CSESA s3 object.*

Description

Get the information string from the CSESA s3 object.

Usage

```
GetStr(csesa)
```

Arguments

`csesa`	The S3 object CSESA.

Value

The string record the newly spacers and serotype information.

PCR *Get the CSESA obeject through the two sequence.*

Description

Get the CSESA obeject through the two sequence.

Usage

```
PCR(seq1, seq2, out.file)
```

Arguments

`seq1` The first DNA sequence.

`seq2` The second DNA sequence.

`out.file` Into which results will be saved if this value is set. Otherwise results will be displayed on the screen.

ReadInFile *Read the three types of input file.*

Description

Read the three types of input file.

Usage

```
ReadInFile(file.name)
```

Arguments

`file.name` The input file name.

Value

The molecular sequence as a string.

WGS	*Find the serotype based on the analysis of the new spacers.*

Description

Find the serotype based on the analysis of the new spacers.

Usage

```
WGS(file)
```

Arguments

`file`	The input fasta file.

Value

The two DNA molecular sequence.

Index