Package 'rhinotypeR'

November 22, 2024

Type Package

Title Rhinovirus genotyping

Version 1.1.0

Date 2024-10-03

Description

``rhinotypeR" is designed to automate the comparison of sequence data against prototype strains, streamlining the genotype assignment process. By implementing predefined pairwise distance thresholds,
this package makes genotype assignment accessible to researchers and public health professionals. This tool enhances our epidemiological toolkit by enabling more efficient surveillance and analysis of
rhinoviruses (RVs) and other viral pathogens with complex genomic landscapes.
Additionally, ``rhinotypeR" supports comprehensive visualization and analysis of
single nucleotide polymorphisms (SNPs) and amino acid substitutions,
facilitating in-depth genetic and evolutionary studies.

License MIT + file LICENSE

Encoding UTF-8

Imports Biostrings

LazyData false

biocViews Sequencing, Genetics, Phylogenetics

Depends R (>= 4.4.0)

Suggests knitr, rmarkdown, BiocManager, BiocStyle, testthat (>= 3.0.0)

VignetteBuilder knitr

Vignettes `Introduction to rhinotypeR` = ``vignettes/rhinotypeR.Rmd"

URL https://github.com/omicscodeathon/rhinotypeR

BugReports https://github.com/omicscodeathon/rhinotypeR/issues

RoxygenNote 7.3.2

Config/testthat/edition 3

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

git_branch devel

git_last_commit 4fd8c88

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-21

Author Martha Luka [aut, cre] (ORCID: <https://orcid.org/0000-0001-6217-4426>),
Ruth Nanjala [aut],
Winfred Gatua [aut],
Wafaa M. Rashed [aut],
Olaitan Awe [aut]

Maintainer Martha Luka <marthaluka20@gmail.com>

Contents

`assignTypes` *Assigns genotypes to input sequences*

Description

Rapidly assigns genotypes to input sequences. The input fasta file should include the prototype strains, which can be downloaded using getPrototypeSeqs().

Usage

```
assignTypes(fastaData, model = "p-distance", gapDeletion = TRUE, threshold = 0.105)
```

Arguments

`fastaData`	The fasta data to be processed.
`model`	The evolutionary model to be used. Default is set to "p-distance".
`gapDeletion`	Logical. If TRUE, gaps are deleted. Default is TRUE.
`threshold`	The distance threshold for genotype assignment. Default is 0.105.

Details

This function compares input sequences against prototype strains using a specified evolutionary model and assigns genotypes based on predefined distance thresholds.

Value

A list with the assigned genotypes and their distances to the reference sequences.

Note

Ensure the input fasta file includes the necessary prototype strains.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

`getPrototypeSeqs`, `pairwiseDistances`

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Run command
fastaD <- Biostrings::readDNAStringSet(test)
assignTypes(fastaD, model = "p-distance", gapDeletion = TRUE, threshold = 0.105)
```

---

countSNPs *Counts single nucleotide polymorphisms*

Description

Counts single nucleotide polymorphisms across input sequences and produces an output matrix.

Usage

```
countSNPs(fastaData, gapDeletion = TRUE)
```

Arguments

`fastaData`	The fasta data to be processed.
`gapDeletion`	Whether or not to delete positions with gaps. Default is set to TRUE.

Details

This function counts the number of single nucleotide polymorphisms (SNPs) across the provided sequences.

Value

A matrix with the SNP counts for each sequence.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

pairwiseDistances

Examples

```
# Load the dataset
test <- system.file("extdata", "test.fasta", package = "rhinotypeR")

# Run the function
fastaData <- Biostrings::readDNAStringSet(test)
countSNPs(fastaData)
```

---

getPrototypeSeqs *Download rhinovirus prototype strains*

Description

Download rhinovirus prototype strains into a local directory. These sequences should be combined with and aligned alongside newly generated sequences before being imported into R for genotype assignment.

Usage

```
getPrototypeSeqs(destinationFolder, overwrite = TRUE)
```

Arguments

destinationFolder	Provide a path that will act as an output folder for the prototype sequences. The default is set to "output", but you will need to create it prior to running the command in your current directory.
overwrite	Logical value indicating whether to overwrite existing files in the destination folder. The default is TRUE.

Details

This function downloads rhinovirus prototype strains and saves them in the specified folder. If overwrite = TRUE, existing files with the same name in the destination folder will be replaced.

Value

A character vector of the downloaded file paths.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

assignTypes

Examples

```
if (interactive()) {
    # Specify a destination directory
    dest_dir <- tempdir()  # Create a temporary directory for the example

    # Download prototypes with default overwrite = TRUE
    getPrototypeSeqs(destinationFolder = dest_dir)

    # List downloaded files
    list.files(dest_dir)
    }
```

---

overallMeanDistance *Estimates the overall mean distance*

Description

Estimates the overall mean distance of input sequences.

Usage

```
overallMeanDistance(fastaData, model="p-distance", gapDeletion=TRUE)
```

Arguments

fastaData	The fasta file used here is the output from the Biostrings function 'Biostrings::readDNAStringSet'.
model	The evolutionary model used to calculate distances. Default is set to "p-distance".
gapDeletion	Whether or not to delete positions with gaps. Default is set to TRUE.

Details

This function estimates the overall mean genetic distance of input sequences using the specified evolutionary model.

Value

A numeric value representing the overall mean distance.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

pairwiseDistances

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Usage
fastaData <- Biostrings::readDNAStringSet(test)
overallMeanDistance(fastaData, model="p-distance")
```

---

pairwiseDistances *Estimates pairwise distances*

Description

Estimates pairwise distances across input sequences using a specified evolutionary model.

Usage

```
pairwiseDistances(fastaData, model = "p-distance", gapDeletion = TRUE)
```

Arguments

fastaData	The fasta data to be processed.
model	The evolutionary model used to calculate distances. Default is set to "p-distance".
gapDeletion	Whether or not to delete positions with gaps. Default is set to TRUE.

Details

This function calculates the pairwise genetic distances between sequences using the specified evolutionary model.

Value

A matrix of pairwise distances.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

overallMeanDistance

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Example usage
fastaD <- Biostrings::readDNAStringSet(test)
pairwiseDistances(fastaData = fastaD, model = "p-distance", gapDeletion = TRUE)
```

---

plotAA *Visualize amino acid substitutions*

Description

Plots amino acid substitutions with a specified sequence as the reference. The input is an amino acid fasta file (translated DNA sequences). To specify the reference sequence, move it to the bottom of the alignment. Changes are colored by the class of amino acid: Red = Positively charged, Blue = Negatively charged, Green = Polar, Yellow = Non-polar.

Usage

```
plotAA(AAfastaFile, showLegend = FALSE)
```

Arguments

AAfastaFile	The file path to the input amino acid sequences in fasta format.
showLegend	Logical indicating whether to show the legend. Default is FALSE.

Details

This function visualizes amino acid substitutions in a given set of sequences with color-coded classes.

Value

A plot object showing the amino acid substitutions.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

SNPeek

Examples

```
# Load the dataset
test <- system.file("extdata", "test.translated.fasta", package = "rhinotypeR")

# Usage
testData <- Biostrings::readAAStringSet(test)
plotAA(testData)
```

`plotDistances` *Visualizes pairwise genetic distances*

Description

Visualizes pairwise genetic distances in a heatmap. This function uses the output of pairwiseDistances() as input.

Usage

```
plotDistances(distancesMatrix)
```

Arguments

`distancesMatrix` A matrix of pairwise genetic distances from the function `pairwiseDistances`.

Details

This function creates a heatmap to visualize the pairwise genetic distances between sequences.

Value

A heatmap plot object.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

`pairwiseDistances`

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Example usage
fastaD <- Biostrings::readDNAStringSet(test)
distancesMatrix <- pairwiseDistances(fastaD, "p-distance")
plotDistances(distancesMatrix)
```

plotFrequency *Plots the frequency of assigned genotypes*

Description

Plots the frequency of assigned genotypes. This function uses the output of `assignTypes()` as input.

Usage

```
plotFrequency(assignedTypesDF, showLegend = FALSE)
```

Arguments

`assignedTypesDF`	A dataframe from the function `assignTypes`.
`showLegend`	Logical indicating whether to show the legend. Default is FALSE.

Details

This function visualizes the frequency of assigned genotypes based of the newly generated data.

Value

A plot showing the frequency of assigned genotypes.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

`assignTypes`

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Run
queryFastaData <- Biostrings::readDNAStringSet(test)
df <- assignTypes(queryFastaData, "p-distance")

plotFrequency(df)
```

plotTree *Plots a simple phylogenetic tree*

Description

Plots a simple phylogenetic tree using the genetic distances estimated by `pairwiseDistances()` and allows for customization using additional arguments.

Usage

```
plotTree(distance_matrix, ...)
```

Arguments

`distance_matrix`	Distance matrix from the function `pairwiseDistances`.
`...`	Additional parameters passed to the `plot()` function for further customization of the tree plot.

Details

This function visualizes a phylogenetic tree based on the calculated pairwise genetic distances. Users can customize the appearance of the plot by providing additional parameters through the `...` argument, which are passed directly to the `plot()` function.

Value

A plot object representing the phylogenetic tree.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

`pairwiseDistances`

Examples

```
# Load the dataset
test <- system.file("extdata", "input_aln.fasta", package = "rhinotypeR")

# Example usage
fastaD <- Biostrings::readDNAStringSet(test)
pdistances <- pairwiseDistances(fastaD, "p-distance")
plotTree(pdistances, col = "blue", lwd = 2)
```

rhinotypeR *rhinotypeR: A Package for the genotyping of rhinoviruses using the VP4/2 region*

Description

The **rhinotypeR** package provides tools and functions that streamline the genotyping of rhinoviruses using the VP4/2 region.

Details

This package includes functions to calculate pairwise distances and visualization, specifically tailored for rhinovirus data. It also provides additional utilities for working with sequence data in FASTA format such as simple phylogenetic trees.

Main functions

- `getPrototypeSeqs`: Download prototype sequences into local machine.
- `pairwiseDistances`: Calculates pairwise distances using a specified evolutionary model.
- `assignTypes`: Assigns sequences to their respective rhinovirus genotypes.
- `plotTree`: Plots a simple phylogenetic tree.

Getting started

To get started with **rhinotypeR**, download the prototype sequences and combine these with your newly generated VP4/2 sequences and align using a suitable tool. Then import the curated alignment into R:

```
\# Download prototype sequences
getPrototypeSeqs("path/to/destination")

\# Read sequences from a FASTA file
sequences <- Biostrings::readDNAStringSet("../inst/extdata/input_aln.fasta")

\# Perform sequence analysis
distance_matrix <- pairwiseDistances(sequences)

\# Assign to genotypes
assigned_types <- assignTypes(distance_matrix)

\# Simple phylogenetic tree
plotTree(distance_matrix)
```

For more detailed examples and usage, refer to the package vignettes:

```
vignette(package = "rhinotypeR")
vignette("rhinotypeR", package = "rhinotypeR")
```

Author(s)

Maintainer: Martha Luka <marthaluka20@gmail.com> (ORCID)

Authors:

- Ruth Nanjala
- Winfred Gatua
- Wafaa M. Rashed
- Olaitan Awe

See Also

https://github.com/omicscodeathon/rhinotypeR

SNPeek *Visualize single nucleotide polymorphisms*

Description

Visualizes single nucleotide polymorphisms (SNPs) relative to a specified reference sequence. To specify the reference, manually move it to the bottom of the alignment. Substitutions are color-coded by nucleotide: A = green, T = red, C = blue, G = yellow.

Usage

```
SNPeek(fastaData, showLegend = FALSE)
```

Arguments

fastaData	The fasta file used here is the output from the function 'Biostrings::readDNAStringSet'.
showLegend	Logical indicating whether to show the legend. Default is FALSE.

Details

This function visualizes SNPs in the provided sequence data, using a color-coding scheme for different nucleotides.

Value

A plot showing the SNPs relative to a user specified reference sequence.

Author(s)

Martha Luka, Ruth Nanjala, Wafaa Rashed, Winfred Gatua, Olaitan Awe

See Also

plotAA

Examples

```
# Load the dataset
test <- system.file("extdata", "test.fasta", package = "rhinotypeR")

fastaData <- Biostrings::readDNAStringSet(test)
SNPeek(fastaData, showLegend = FALSE)
```

Index