Package ‘sangeranalyseR’

May 30, 2024

Type Package

Title sangeranalyseR: a suite of functions for the analysis of Sanger sequence data in R

Version 1.14.0

Date 2024-04-24

Author Rob Lanfear <rob.lanfear@gmail.com>, Kuan-Hao Chao <ntueeb05howard@gmail.com>

Maintainer Kuan-Hao Chao <ntueeb05howard@gmail.com>

biocViews Genetics, Alignment, Sequencing, SangerSeq, Preprocessing, QualityControl, Visualization, GUI

Description This package builds on sangerseqR to allow users to create contigs from collections of Sanger sequencing reads. It provides a wide range of options for a number of commonly-performed actions including read trimming, detecting secondary peaks, and detecting indels using a reference sequence. All parameters can be adjusted interactively either in R or in the associated Shiny applications. There is extensive online documentation, and the package can outputs detailed HTML reports, including chromatograms.

License GPL-2

Encoding UTF-8

Depends R (>= 4.0.0), stringr, ape, Biostrings, pwalign, DECIPHER, parallel, reshape2, sangerseqR, gridExtra, shiny, shinydashboard, shinyjs, data.table, plotly, DT, zealot, excelR, shinycssloaders, ggdendro, shinyWidgets, openxlsx, tools, rmarkdown (>= 2.9), knitr (>= 1.33), seqinr, BiocStyle, logger

RoxygenNote 7.2.1

VignetteBuilder knitr

Suggests testthat (>= 2.1.0)

Collate '.AllGenerics.R' '.ClassChromatogramParam.R'
'.ClassObjectResults.R' '.ClassQualityReport.R'
'.ClassSangerRead.R' '.ClassSangerAlignment.R'
'.ClassSangerContig.R' '.Constructors.R' '.LoadMessage.R'
Contents

ChromatogramParam-class .................................................. 3
generateReport ................................................................. 4
generateReportSA ............................................................. 5
generateReportSC ............................................................. 6
generateReportSR ............................................................ 7
launchApp .......................................................... 8
launchAppSA ............................................................. 9
launchAppSC ............................................................. 9
MakeBaseCalls ......................................................... 10
ObjectResults-class ....................................................... 11
qualityBasePlot ..................................................... 11
QualityReport-class ..................................................... 12
QualityReport-class-qualityBasePlot ................................... 13
QualityReport-class-updateQualityParam ................................ 14
qualityReportData ...................................................... 15
readTable .............................................................. 15
SangerAlignment ......................................................... 16
SangerAlignment-class .................................................... 19
SangerAlignment-class-generateReportSA ................................ 22
SangerAlignment-class-launchAppSA .................................... 23
SangerAlignment-class-updateQualityParam ................................ 24
SangerAlignment-class-writeFastaSA .................................... 25
sangerAlignmentData ..................................................... 26
sangeranalyseR .......................................................... 26
SangerContig ............................................................ 26
SangerContig-class .......................................................... 30
SangerContig-class-generateReportSC .................................... 33
SangerContig-class-launchAppSC .......................................... 34
SangerContig-class-readTable ............................................. 35
SangerContig-class-updateQualityParam .................................. 36

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

Repository Bioconductor 3.19

Date/Publication 2024-05-29
**ChromatogramParam-class**

An S4 class storing chromatogram related inputs in a SangerRead S4 object.

**Slots**

- `baseNumPerRow` It defines maximum base pairs in each row. The default value is 100.
- `heightPerRow` It defines the height of each row in chromatogram. The default value is 200.
- `signalRatioCutoff` The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is 0.33.
- `showTrimmed` The logical value storing whether to show trimmed base pairs in chromatogram. The default value is TRUE.

**Author(s)**

Kuan-Hao Chao

**Examples**

```r
Chromatogram <- new("ChromatogramParam",
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE)
```
generateReport  

Method generateReport

Description

A method which generates final reports of the SangerRead, SangerContig, and SangerAlignment instance.

Usage

generateReport(
  object,
  outputDir = NULL,
  includeSangerContig = TRUE,
  includeSangerRead = TRUE,
  colors = "default",
  ...
)

Arguments

object        A SangerRead, SangerContig, or SangerAlignment S4 instance.
outputDir     The output directory of the generated HTML report.
includeSangerContig The parameter that decides whether to include SangerContig level report. The
                     value is TRUE or FALSE and the default is TRUE.
includeSangerRead The parameter that decides whether to include SangerRead level report. The
                   value is TRUE or FALSE and the default is TRUE.
colors        A vector for users to set the colors of (A, T, C, G, else). There are three options
              for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0),
              (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
...

Value

A SangerRead, SangerContig, or SangerAlignment object.

Author(s)

Kuan-Hao Chao
Examples

data(sangerReadFData)
data(sangerContigData)
data(sangerAlignmentData)
## Not run:
generateReport(sangerReadFData)
generateReport(sangerReadFData, colors="cb_friendly")
generateReport(sangerContigData)
generateReport(sangerContigData, colors="cb_friendly")
generateReport(sangerAlignmentData)
generateReport(sangerAlignmentData, colors="cb_friendly")
## End(Not run)

generateReportSA Method generateReportSA

Description
Method generateReportSA

Usage

generateReportSA(
  object,
  outputDir = NULL,
  includeSangerContig = TRUE,
  includeSangerRead = TRUE,
  colors = "default",
  ...
)

Arguments

object A SangerAlignment S4 instance.
outputDir The output directory of the generated HTML report.
includeSangerContig The parameter that decides whether to include SangerContig level report. The value is TRUE or FALSE and the default is TRUE.
includeSangerRead The parameter that decides whether to include SangerRead level report. The value is TRUE or FALSE and the default is TRUE.
colors A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
... Further generateReportSA-related parameters.
generateReportSC

Value

The output absolute path to the SangerAlignment’s HTML file.

Examples

data(sangerAlignmentData)
## Not run:
generateReportSA(sangerAlignmentData)
## End(Not run)

generateReportSC  Method generateReportSC

Description

Method generateReportSC

Usage

generateReportSC(
  object,
  outputDir = NULL,
  includeSangerRead = TRUE,
  colors = "default",
  ...
)

Arguments

object  A SangerContig S4 instance.
outputDir  The output directory of the generated HTML report.
includeSangerRead
  The parameter that decides whether to include SangerRead level report. The value is TRUE or FALSE and the default is TRUE.
colors  A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
...
  Further generateReportSC-related parameters.

Value

The output absolute path to the SangerContig’s HTML file.
Method `generateReportSR`

**Usage**

```r
generateReportSR(object, outputDir = NULL, colors = "default", ...)
```

**Arguments**

- `object` A SangerRead S4 instance.
- `outputDir` The output directory of the generated HTML report.
- `colors` A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from: 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
- `...` Further `generateReportSR`-related parameters.

**Value**

The output absolute path to the SangerRead's HTML file.

**Examples**

```r
data(sangerContigData)
## Not run:
generateReportSC(sangerContigData)
## End(Not run)
```
launchApp

Method launchApp

Description

A method which launches Shiny application of the SangerContig and SangerAlignment instance.

Usage

launchApp(object, outputDir = NULL, colors = "default")

Arguments

- **object**: A SangerContig or SangerAlignment S4 instance.
- **outputDir**: The output directory of the saved new SangerContig or SangerAlignment S4 instance.
- **colors**: A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.

Value

A SangerContig or SangerAlignment object.

Author(s)

Kuan-Hao Chao

Examples

data(sangerContigData)
data(sangerAlignmentData)
## Not run:
launchApp(sangerContigData)
launchApp(sangerContigData, colors="cb_friendly")
launchApp(sangerAlignmentData)
launchApp(sangerAlignmentData, colors="cb_friendly")
## End(Not run)
launchAppSA

Method launchAppSA

Description

Method launchAppSA

Usage

launchAppSA(object, outputDir = NULL, colors = "default")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A SangerAlignment S4 instance.</td>
</tr>
<tr>
<td>outputDir</td>
<td>The output directory of the saved new SangerAlignment S4 instance.</td>
</tr>
<tr>
<td>colors</td>
<td>A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. &quot;default&quot;: (green, blue, black, red, purple). 2. &quot;cb_friendly&quot;: ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.</td>
</tr>
</tbody>
</table>

Value

A shiny.appobj object.

Examples

data(sangerAlignmentData)
## Not run:
launchAppSA(sangerAlignmentData)
## End(Not run)

launchAppSC

Method launchAppSC

Description

Method launchAppSC

Usage

launchAppSC(object, outputDir = NULL, colors = "default")
Arguments

- **object**: A SangerContig S4 instance.
- **outputDir**: The output directory of the saved new SangerContig S4 instance.
- **colors**: A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.

Value

A shiny.appobj object.

Examples

```r
data(sangerContigData)
## Not run:
launchAppSC(sangerContigData)
## End(Not run)
```

MakeBaseCalls

Method **MakeBaseCalls**

Description

Method MakeBaseCalls

Usage

```
MakeBaseCalls(object, signalRatioCutoff = 0.33)
```

Arguments

- **object**: A SangerRead S4 instance.
- **signalRatioCutoff**: The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is **0.33**.

Value

A SangerRead instance.

Examples

```r
data(sangerReadFData)
MakeBaseCalls(sangerReadFData, signalRatioCutoff = 0.22)
```
ObjectResults-class  

ObjectResults

Description
An S4 class storing results related inputs in a SangerRead, SangerContig, and SangerAlignment S4 object.

Slots
printLevel

Author(s)
Kuan-Hao Chao

Examples

objectResults <- new("ObjectResults",
  creationResult = TRUE,
  errorMessages = character(0),
  errorTypes = character(0),
  warningMessages = character(0),
  warningTypes = character(0),
  readResultTable = data.frame(),
  printLevel = "SangerRead")

qualityBasePlot

Method qualityBasePlot

Description
Method qualityBasePlot

Usage
qualityBasePlot(object)

Arguments

object  
A QualityReport or SangerRead S4 instance

Value
A quality plot.
Examples

data(qualityReportData)
data(sangerReadFData)
qualityBasePlot(qualityReportData)
qualityBasePlot(sangerReadFData)

Description

An S4 class storing quality related inputs and results in a SangerRead S4 object.

Slots

TrimmingMethod The read trimming method for this SangerRead. The value must be "M1" (the
default) or 'M2'.
M1TrimmingCutoff The trimming cutoff for the Method 1. IfTrimmingMethod is "M1", then the
default value is 0.0001. Otherwise, the value must be NULL.
M2CutoffQualityScore The trimming cutoff quality score for the Method 2. IfTrimmingMethod
is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with
M2SlidingWindowSize.
M2SlidingWindowSize The trimming sliding window size for the Method 2. IfTrimmingMethod
is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with
M2CutoffQualityScore.
qualityPhredScores The Phred quality scores of each base pairs after base calling.
qualityBaseScores The probability of incorrect base call of each base pairs. They are calculated
from qualityPhredScores.
rawSeqLength The number of nucleotides of raw primary DNA sequence.
trimmedSeqLength The number of nucleotides of trimmed primary DNA sequence.
trimmedStartPos The base pair index of trimming start point from 5' end of the sequence.
trimmedFinishPos The base pair index of trimming finish point from 3' end of the sequence.
rawMeanQualityScore The mean quality score of the primary sequence after base calling. In other
words, it is the mean of qualityPhredScores.
trimmedMeanQualityScore The mean quality score of the trimmed primary sequence after base
calling.
rawMinQualityScore The minimum quality score of the primary sequence after base calling.
trimmedMinQualityScore The minimum quality score of the trimmed primary sequence after
base calling.
remainingRatio The remaining sequence length ratio after trimming.

Author(s)

Kuan-Hao Chao
Examples

```r
inputFilesPath <- system.file("extdata/", package = "sangeranalyseR")
A_chloroticaFFN <- file.path(inputFilesPath,
    "Allolobophora_chlorotica",
    "ACHLO",
    "Achl_ACHLO006-09_1_F.ab1")
sangerReadF <- new("SangerRead",
    inputSource = "ABIF",
    readFeature = "Forward Read",
    readFileName = A_chloroticaFFN,
    geneticCode = GENETIC_CODE,
    TrimmingMethod = "M1",
    M1TrimmingCutoff = 0.0001,
    M2CutoffQualityScore = NULL,
    M2SlidingWindowSize = NULL,
    baseNumPerRow = 100,
    heightPerRow = 200,
    signalRatioCutoff = 0.33,
    showTrimmed = TRUE)
"@(sangerReadF, QualityReport)
```

Description

A QualityReport method which creates quality base interactive plot.

Usage

```r
## S4 method for signature 'QualityReport'
qualityBasePlot(object)
```

Arguments

- **object**: A QualityReport S4 instance.

Value

A quality plot.

Examples

```r
data("qualityReportData")
## Not run:
qualityBasePlot(qualityReportData)
## End(Not run)
```
Description

A QualityReport method which updates quality base interactive plot.

Usage

```r
## S4 method for signature 'QualityReport'
updateQualityParam(
  object,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL
)
```

Arguments

- `object` : A QualityReport S4 instance.
- `TrimmingMethod` : The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.
- `M1TrimmingCutoff` : The trimming cutoff for the Method 1. If `TrimmingMethod` is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.
- `M2CutoffQualityScore` : The trimming cutoff quality score for the Method 2. If `TrimmingMethod` is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with `M2SlidingWindowSize`.
- `M2SlidingWindowSize` : The trimming sliding window size for the Method 2. If `TrimmingMethod` is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with `M2CutoffQualityScore`.

Value

A QualityReport instance.

Examples

```r
data("qualityReportData")
updateQualityParam(qualityReportData,
  TrimmingMethod = "M2",
  M1TrimmingCutoff = NULL,
  M2CutoffQualityScore = 30,
  M2SlidingWindowSize = 15)
```
qualityReportData

Description
QualityReport instance

Usage
data(qualityReportData)

Author(s)
Kuan-Hao Chao

readTable

Description
Method readTable

Usage
readTable(object, indentation = 0, ...)

Arguments
object A SangerRead, SangerContig, or SangerAlignment S4 instance.
indentation The indentation for different level printing
... Further generateReportSR-related parameters.

Value
None.

Examples
data(sangerReadFData)
data(sangerContigData)
data(sangerAlignmentData)
## Not run:
readTable(sangerReadFData)
readTable(sangerContigData)
readTable(sangerAlignmentData)
## End(Not run)
Description

the wrapper function for SangerAlignment

Usage

SangerAlignment(
  printLevel = "SangerAlignment",
  inputSource = "ABIF",
  processMethod = "REGEX",
  ABIF_Directory = NULL,
  FASTA_File = NULL,
  REGEX_SuffixForward = NULL,
  REGEX_SuffixReverse = NULL,
  CSV_NamesConversion = NULL,
  geneticCode = GENETIC_CODE,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE,
  refAminoAcidSeq = "",
  minReadsNum = 2,
  minReadLength = 20,
  minFractionCall = 0.5,
  maxFractionLost = 0.5,
  acceptStopCodons = TRUE,
  readingFrame = 1,
  processorsNum = 1
)

Arguments

inputSource The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".

ABIF_Directory The parent directory of all of the reads contained in ABIF format you wish to analyse. In SangerAlignment, all reads in subdirectories will be scanned recursively.

FASTA_File If inputSource is "FASTA", then this value has to be the name of the FASTA file; if inputSource is "ABIF", then this value is "" by default.
REGEX_SuffixForward
The suffix of the filenames for forward reads in regular expression, i.e. reads that do not need to be reverse-complemented. For forward reads, it should be "_F.ab1".

REGEX_SuffixReverse
The suffix of the filenames for reverse reads in regular expression, i.e. reads that need to be reverse-complemented. For reverse reads, it should be "_R.ab1".

CSV_NamesConversion
The file path to the CSV file that provides read names that follow the naming regulation. If inputSource is "FASTA", then users need to prepare the csv file or make sure the original names inside FASTA file are valid; if inputSource is "ABIF", then this value is NULL by default.

geneticCode
Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.

TrimmingMethod
TrimmingMethod The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.

M1TrimmingCutoff
The trimming cutoff for the Method 1. If TrimmingMethod is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.

M2CutoffQualityScore
The trimming cutoff quality score for the Method 2. If TrimmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.

M2SlidingWindowSize
The trimming sliding window size for the Method 2. If TrimmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.

baseNumPerRow
It defines maximum base pairs in each row. The default value is 100.

heightPerRow
It defines the height of each row in chromatogram. The default value is 200.

signalRatioCutoff
The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is 0.33.

showTrimmed
The logical value storing whether to show trimmed base pairs in chromatogram. The default value is TRUE.

refAminoAcidSeq
An amino acid reference sequence supplied as a string or an AAString object. If your sequences are protein-coding DNA sequences, and you want to have frameshifts automatically detected and corrected, supply a reference amino acid sequence via this argument. If this argument is supplied, the sequences are then kept in frame for the alignment step. Fwd sequences are assumed to come from the sense (i.e. coding, or "+") strand. The default value is "".
minReadsNum  The minimum number of reads required to make a consensus sequence, must be 2 or more. The default value is 2.

minReadLength  Reads shorter than this will not be included in the readset. The default 20 means that all reads with length of 20 or more will be included. Note that this is the length of a read after it has been trimmed.

minFractionCall  Minimum fraction of the sequences required to call a consensus sequence for SangerContig at any given position (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.75 implying that 3/4 of all reads must be present in order to call a consensus.

maxFractionLost  Numeric giving the maximum fraction of sequence information that can be lost in the consensus sequence for SangerContig (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.5, implying that each consensus base can ignore at most 50 percent of the information at a given position.

acceptStopCodons  The logical value TRUE or FALSE. TRUE (the default): keep all reads, regardless of whether they have stop codons; FALSE: reject reads with stop codons. If FALSE is selected, then the number of stop codons is calculated after attempting to correct frameshift mutations (if applicable).

readingFrame  1, 2, or 3. Only used if accept.stop.codons == FALSE. This specifies the reading frame that is used to determine stop codons. If you use a refAminoAcidSeq, then the frame should always be 1, since all reads will be shifted to frame 1 during frameshift correction. Otherwise, you should select the appropriate reading frame.

processorsNum  The number of processors to use, or NULL (the default) for all available processors.

minFractionCallSA  Minimum fraction of the sequences required to call a consensus sequence for SangerAlignment at any given position (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.75 implying that 3/4 of all reads must be present in order to call a consensus.

maxFractionLostSA  Numeric giving the maximum fraction of sequence information that can be lost in the consensus sequence for SangerAlignment (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.5, implying that each consensus base can ignore at most 50 percent of the information at a given position.

Value  A SangerAlignment instance.

Author(s)  Kuan-Hao Chao
Examples

```r
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, "Allolobophora_chlorotica", "RBNII")
REGEX_SuffixForward <- "\[0-9\]*_F.ab1$"
REGEX_SuffixReverse <- "\[0-9\]*_R.ab1$"
sangerAlignment <- SangerAlignment(
  inputSource = "ABIF",
  ABIF_Directory = parentDir,
  REGEX_SuffixForward = REGEX_SuffixForward,
  REGEX_SuffixReverse = REGEX_SuffixReverse,
  refAminoAcidSeq = "IHWYPLFTGLTLNNKWLKSHFIIMFIGVNLTFFPQHFLGLAGMPRRYSDYPDAYTTWNIVSTIGSTISLLGILFFFFIIWESLVSQRQVIYPIQLNSSIEWYQNTPPAEHSYSELPLLTN",
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 0.0001,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE,
  processorsNum = 2)
```

Description

An S4 class containing SangerContigs lists and contigs alignment results which corresponds to a final alignment in Sanger sequencing.

Slots

- **objectResults** This is the object that stores all information of the creation result.
- **inputSource** The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".
- **processMethod** The method to create a contig from reads. The value is "REGEX" or "CSV". The default value is "REGEX".
- **ABIF_Directory** If inputSource is "ABIF", then this value is the path of a parent directory storing all reads in ABIF format you want to analyse. If inputSource is "FASTA", then this value has to be NULL by default.
- **FASTA_File** If inputSource is "FASTA", then this value has to be the path to a valid FASTA file; if inputSource is "ABIF", then this value has to be NULL by default.
- **REGEX_SuffixForward** The suffix of the filenames for forward reads in regular expression, i.e. reads that do not need to be reverse-complemented. For forward reads, it should be ".F.ab1".
- **REGEX_SuffixReverse** The suffix of the filenames for reverse reads in regular expression, i.e. reads that need to be reverse-complemented. For reverse reads, it should be ".R.ab1".
CSV_NamesConversion  The file path to the CSV file that provides read names, directions, and their contig groups. If processMethod is "CSV", then this value has to be the path to a valid CSV file; if processMethod is "REGEX", then this value has to be NULL by default.

geneticCode  Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.

refAminoAcidSeq  An amino acid reference sequence supplied as a string or an AAString object. If your sequences are protein-coding DNA sequences, and you want to have frameshifts automatically detected and corrected, supply a reference amino acid sequence via this argument. If this argument is supplied, the sequences are then kept in frame for the alignment step. Fwd sequences are assumed to come from the sense (i.e. coding, or "+") strand. The default value is "".

contigList  A list storing all SangerContigs S4 instances.

contigsConsensus  The consensus read of all SangerContig S4 instances in DNAString object.

contigsAlignment  The alignment of all SangerContig S4 instances with the called consensus sequence in DNAStringSet object. Users can use BrowseSeqs() to view the alignment.

contigsTree  A phylo instance returned by bionj function in ape package. It can be used to draw the tree.

Author(s)

Kuan-Hao Chao

Examples

```r
## Simple example
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, 'Allolobophora_chlorotica', 'ACHLO')
my_aligned_contigs <- new("SangerAlignment",
  ABIF_Directory = parentDir,
  REGEX_SuffixForward = "\([0-9]*\)_F.ab1$",
  REGEX_SuffixReverse = "\([0-9]*\)_R.ab1$"
)

rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, 'Allolobophora_chlorotica', 'ACHLO')
CSV_NamesConversion <- file.path(rawDataDir, "ab1", "SangerAlignment", "names_conversion.csv")
sangerAlignment <- new("SangerAlignment",
  processMethod = "CSV",
  ABIF_Directory = parentDir,
  CSV_NamesConversion = CSV_NamesConversion)

## Input From ABIF file format (Regex)
REGEX_SuffixForward <- "\([0-9]*\)_F.ab1$"
REGEX_SuffixReverse <- "\([0-9]*\)_R.ab1$"
sangerAlignment <- new("SangerAlignment",
  printLevel = "SangerAlignment",
  inputSource = "ABIF",
  processMethod = "REGEX",
"
```
SangerAlignment-class

```r
FASTA_File = NULL,
CSV_NamesConversion = NULL,
ABIF_Directory = parentDir,
REGEX_SuffixForward = REGEX_SuffixForward,
REGEX_SuffixReverse = REGEX_SuffixReverse,
TrimmingMethod = "M1",
M1TrimmingCutoff = 0.0001,
M2CutoffQualityScore = NULL,
M2SlidingWindowSize = NULL,
baseNumPerRow = 100,
heightPerRow = 200,
signalRatioCutoff = 0.33,
showTrimmed = TRUE,

refAminoAcidSeq = "IHWYPLFTGLTLNNKWLKSHFIIMFIGVNLTFFPQHFLGLAGMPRRYSDYPDAYTTWNIVSTIGSTISLLGILFFFFIIWESLVSQRQVIYPIQLNSSIEWYQNTPPAEHSYSELPLLTN",

minReadsNum = 2,
minReadLength = 20,
minFractionCall = 0.5,
maxFractionLost = 0.5,
geneticCode = GENETIC_CODE,
acceptStopCodons = TRUE,
readingFrame = 1,
processorsNum = 2)

## Input From ABIF file format (Csv three column)
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, 'Allolobophora_chlorotica', 'ACHLO')
CSV_NamesConversion <- file.path(rawDataDir, "ab1", "SangerAlignment", "names_conversion_all.csv")
sangerAlignment <- new("SangerAlignment",
inputSource = "ABIF",
processMethod = "CSV",
ABIF_Directory = parentDir,
CSV_NamesConversion = CSV_NamesConversion,
refAminoAcidSeq = "IHWYPLFTGLTLNNKWLKSHFIIMFIGVNLTFFPQHFLGLAGMPRRYSDYPDAYTTWNIVSTIGSTISLLGILFFFFIIWESLVSQRQVIYPIQLNSSIEWYQNTPPAEHSYSELPLLTN",

TrimmingMethod = "M1",
M1TrimmingCutoff = 0.0001,
M2CutoffQualityScore = NULL,
M2SlidingWindowSize = NULL,
baseNumPerRow = 100,
heightPerRow = 200,
signalRatioCutoff = 0.33,
showTrimmed = TRUE,

## Input From FASTA file format (No Csv - Regex)
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
fastaFN <- file.path(rawDataDir, "fasta", "SangerAlignment", "Sanger_all_reads.fa")
REGEX_SuffixForwardFa <- ".[_0-9]*_F$"
REGEX_SuffixReverseFa <- ".[_0-9]*_R$"
sangerAlignmentFa <- new("SangerAlignment",
inputSource = "FASTA",
processMethod = "REGEX",

```
```
## Input From FASTA file format (Csv three column method)

```r
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
fastaFN <- file.path(rawDataDir, "fasta", "SangerAlignment", "Sanger_all_reads.fa")
CSV_NamesConversion <- file.path(rawDataDir, "fasta", "SangerAlignment", "names_conversion.csv")
sangerAlignmentFa <- new("SangerAlignment",
  inputSource = "FASTA",
  processMethod = "CSV",
  FASTA_File = fastaFN,
  CSV_NamesConversion = CSV_NamesConversion,
  refAminoAcidSeq = "SRQWLFSTNHKDIGTLFYIFGAWAGMVGTSLSILIRAEGLHGPALIGDDQIYNVIPHTAHAFIMIFFMVMPPIMIGGIGPGVIIWESLVSQRQVIYPIQLNSSIEWYQNTPPAEHSYSELPLLTN",
  processorsNum = 2)
```

---

### Description

A SangerAlignment method which generates final reports of the SangerContig instance.

### Usage

```r
## S4 method for signature 'SangerAlignment'
genrateReportSA(
  object, outputDir, includeSangerContig = TRUE,
  includeSangerRead = TRUE, colors
)
```

### Arguments

- **object**: A SangerAlignment S4 instance.
- **outputDir**: The output directory of the generated HTML report.
- **includeSangerContig**: The parameter that decides whether to include SangerContig level report. The value is TRUE or FALSE and the default is TRUE.
- **includeSangerRead**: The parameter that decides whether to include SangerRead level report. The value is TRUE or FALSE and the default is TRUE.
colors A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.

Value
The output absolute path to the SangerAlignment's HTML file.

Examples

```r
data("sangerAlignmentData")
## Not run:
generateReportSA(sangerAlignmentData)
generateReportSA(sangerAlignmentData, colors="cb_friendly")
## End(Not run)
```

Description
A SangerAlignment method which launches Shiny app for SangerAlignment instance.

Usage
```r
## S4 method for signature 'SangerAlignment'
launchAppSA(object, outputDir = NULL, colors = "default")
```

Arguments

- `object` A SangerAlignment S4 instance.
- `outputDir` The output directory of the saved new SangerContig S4 instance.
- `colors` A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.

Value
A shiny.appobj object.

Examples

```r
data("sangerAlignmentData")
RShinySA <- launchAppSA(sangerAlignmentData)
RShinySA <- launchAppSA(sangerAlignmentData, colors="cb_friendly")
```
Description

A SangerAlignment method which updates QualityReport parameter for each the SangerRead instance inside SangerAlignment.

Usage

```r
## S4 method for signature 'SangerAlignment'
updateQualityParam(
  object,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  processorsNum = NULL
)
```

Arguments

- `object`: A SangerAlignment S4 instance.
- `TrimmingMethod`: The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.
- `M1TrimmingCutoff`: The trimming cutoff for the Method 1. If TrimmingMethod is "M1", then the default value is 0.001. Otherwise, the value must be NULL.
- `M2CutoffQualityScore`: The trimming cutoff quality score for the Method 2. If TrimmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.
- `M2SlidingWindowSize`: The trimming sliding window size for the Method 2. If TrimmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.
- `processorsNum`: The number of processors to use, or NULL (the default) for all available processors.

Value

A SangerAlignment instance.
Examples

```r
data("sangerAlignmentData")
## Not run:
updateQualityParam(sangerAlignmentData,
   TrimmingMethod = "M2",
   M1TrimmingCutoff = NULL,
   M2CutoffQualityScore = 40,
   M2SlidingWindowSize = 15)
## End(Not run)
```

Description

A SangerAlignment method which writes sequences into Fasta files.

Usage

```r
## S4 method for signature 'SangerAlignment'
writeFastaSA(
   object,
   outputDir = NULL,
   compress = FALSE,
   compression_level = NA,
   selection = "all"
)
```

Arguments

- `object` A SangerAlignment S4 instance.
- `outputDir` The output directory of generated FASTA files.
- `compress` Like for the `save` function in base R, must be `TRUE` or `FALSE` (the default), or a single string specifying whether writing to the file is to use compression. The only type of compression supported at the moment is "gzip". This parameter will be passed to `writeXStringSet` function in Biostrings package.
- `compression_level` This parameter will be passed to `writeXStringSet` function in Biostrings package.
- `selection` This value can be `all`, `contigs_alignment`, `contigs_unalignment` or `all_reads`. It generates reads and contigs FASTA files.

Value

The output directory of FASTA files.
Examples

```r
data("sangerAlignmentData")
writeFastaSA(sangerAlignmentData)
```

---

`sangerAlignmentData` *SangerAlignment instance*

---

**Description**

*SangerAlignment instance*

**Usage**

```r
data(sangerAlignmentData)
```

**Author(s)**

Kuan-Hao Chao

---

`sangeranalyseR` *sangeranalyseR-package*

---

**Description**

*sangeranalyseR-package*

---

`SangerContig` *SangerContig*

---

**Description**

the wrapper function for `SangerContig`
Usage

SangerContig(
    printLevel = "SangerContig",
    inputSource = "ABIF",
    processMethod = "REGEX",
    ABIF_Directory = NULL,
    FASTA_File = NULL,
    REGEX_SuffixForward = NULL,
    REGEX_SuffixReverse = NULL,
    CSV_NamesConversion = NULL,
    contigName = NULL,
    geneticCode = GENETIC_CODE,
    TrimmingMethod = "M1",
    M1TrimmingCutoff = 1e-04,
    M2CutoffQualityScore = NULL,
    M2SlidingWindowSize = NULL,
    baseNumPerRow = 100,
    heightPerRow = 200,
    signalRatioCutoff = 0.33,
    showTrimmed = TRUE,
    refAminoAcidSeq = "",
    minReadsNum = 2,
    minReadLength = 20,
    minFractionCall = 0.5,
    maxFractionLost = 0.5,
    acceptStopCodons = TRUE,
    readingFrame = 1,
    processorsNum = 1
)

Arguments

inputSource  The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".
ABIF_Directory The parent directory of all of the reads contained in ABIF format you wish to analyse. In SangerContig, all reads must be in the first layer in this directory.
FASTA_File If inputSource is "FASTA", then this value has to be the name of the FASTA file; if inputSource is "ABIF", then this value is "" by default.
REGEX_SuffixForward The suffix of the filenames for forward reads in regular expression, i.e. reads that do not need to be reverse-complemented. For forward reads, it should be "_F.ab1".
REGEX_SuffixReverse The suffix of the filenames for reverse reads in regular expression, i.e. reads that need to be reverse-complemented. For reverse reads, it should be "_R.ab1".
CSV_NamesConversion The file path to the CSV file that provides read names that follow the naming regulation. If inputSource is "FASTA", then users need to prepare the csv file
or make sure the original names inside FASTA file are valid; if inputSource is "ABIF", then this value is NULL by default.

contigName: The contig name of all the reads in ABIF_Directory.

geneticCode: Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.

TrimmingMethod: The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.

M1TrimmingCutoff: The trimming cutoff for the Method 1. IfTrimmingMethod is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.

M2CutoffQualityScore: The trimming cutoff quality score for the Method 2. IfTrimmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.

M2SlidingWindowSize: The trimming sliding window size for the Method 2. IfTrimmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.

baseNumPerRow: It defines maximum base pairs in each row. The default value is 100.

heightPerRow: It defines the height of each row in chromatogram. The default value is 200.

signalRatioCutoff: The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is 0.33.

showTrimmed: The logical value storing whether to show trimmed base pairs in chromatogram. The default value is TRUE.

refAminoAcidSeq: An amino acid reference sequence supplied as a string or an AAString object. If your sequences are protein-coding DNA sequences, and you want to have frameshifts automatically detected and corrected, supply a reference amino acid sequence via this argument. If this argument is supplied, the sequences are then kept in frame for the alignment step. Fwd sequences are assumed to come from the sense (i.e. coding, or "+"") strand. The default value is "."

minReadsNum: The minimum number of reads required to make a consensus sequence, must be 2 or more. The default value is 2.

minReadLength: Reads shorter than this will not be included in the readset. The default 20 means that all reads with length of 20 or more will be included. Note that this is the length of a read after it has been trimmed.

minFractionCall: Minimum fraction of the sequences required to call a consensus sequence for SangerContig at any given position (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.75 implying that 3/4 of all reads must be present in order to call a consensus.
maxFractionLost
Numeric giving the maximum fraction of sequence information that can be
lost in the consensus sequence for SangerContig (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.5, implying that
each consensus base can ignore at most 50 percent of the information at a given
position.

acceptStopCodons
The logical value TRUE or FALSE. TRUE (the defualt): keep all reads, regardless of
whether they have stop codons; FALSE: reject reads with stop codons. If FALSE is
selected, then the number of stop codons is calculated after attempting to correct
frameshift mutations (if applicable).

readingFrame
1, 2, or 3. Only used if accept.stop.codons == FALSE. This specifies the reading
frame that is used to determine stop codons. If you use a refAminoAcidSeq,
then the frame should always be 1, since all reads will be shifted to frame 1 dur-
ing frameshift correction. Otherwise, you should select the appropriate reading
frame.

processorsNum
The number of processors to use, or NULL (the default) for all available proces-
sors.

Value
A SangerContig instance.

Author(s)
Kuan-Hao Chao

Examples
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, "Allolobophora_chlorotica", "ACHLO")
contigName <- "Achl_ACHLO006-09"
REGEX_SuffixForward <- ".F.ab1"
REGEX_SuffixReverse <- ".R.ab1"
sangerContig <- SangerContig(
  inputSource = "ABIF",
  ABIF_Directory = parentDir,
  contigName = contigName,
  REGEX_SuffixForward = REGEX_SuffixForward,
  REGEX_SuffixReverse = REGEX_SuffixReverse,
  refAminoAcidSeq = "SRQWLFSTNHKDIGTLYFIFFGAMVGTSLIRAEGLHPAGLGDQIYVINVTAHAFIMIFFMVMPIMGFGN"
  TrimmingMethod = "M2",
  M1TrimmingCutoff = NULL,
  M2CutoffQualityScore = 20,
  M2SlidingWindowSize = 10,
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE,
  processorsNum = 2)
SangerContig-class

Description

An S4 class containing forward and reverse SangerRead lists and alignment, consensus read results which corresponds to a contig in Sanger sequencing.

Slots

objectResults This is the object that stores all information of the creation result.

inputSource The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".

processMethod The method to create a contig from reads. The value is "REGEX" or "CSV". The default value is "REGEX".

ABIF_Directory If inputSource is "ABIF", then this value is the path of a parent directory storing all reads in ABIF format you want to analyse. If inputSource is "FASTA", then this value has to be NULL by default.

FASTA_File If inputSource is "FASTA", then this value has to be the path to a valid FASTA file; if inputSource is "ABIF", then this value has to be NULL by default.

REGEX_SuffixForward The suffix of the filenames for forward reads in regular expression, i.e. reads that do not need to be reverse-complemented.

REGEX_SuffixReverse The suffix of the filenames for reverse reads in regular expression, i.e. reads that need to be reverse-complemented.

CSV_NamesConversion The file path to the CSV file that provides read names, directions, and their contig groups. If processMethod is "CSV", then this value has to be the path to a valid CSV file; if processMethod is "REGEX", then this value has to be NULL by default.

contigName The contig name of all the reads in ABIF_Directory.

geneticCode Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.

forwardReadList The list of SangerRead S4 instances which are all forward reads.

reverseReadList The list of SangerRead S4 instances which are all reverse reads.

minReadsNum The minimum number of reads required to make a consensus sequence, must be 2 or more. The default value is 2.

minReadLength Reads shorter than this will not be included in the readset. The default 20 means that all reads with length of 20 or more will be included. Note that this is the length of a read after it has been trimmed.

refAminoAcidSeq An amino acid reference sequence supplied as a string or an AAString object. If your sequences are protein-coding DNA sequences, and you want to have frameshifts automatically detected and corrected, supply a reference amino acid sequence via this argument.
If this argument is supplied, the sequences are then kept in frame for the alignment step. Fwd sequences are assumed to come from the sense (i.e. coding, or "+") strand. The default value is "".

\textbf{minFractionCall}  Minimum fraction of the sequences required to call a consensus sequence for SangerContig at any given position (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.75 implying that 3/4 of all reads must be present in order to call a consensus.

\textbf{maxFractionLost}  Numeric giving the maximum fraction of sequence information that can be lost in the consensus sequence for SangerContig (see the ConsensusSequence() function from DECIPHER for more information). Defaults to 0.5, implying that each consensus base can ignore at most 50 percent of the information at a given position.

\textbf{acceptStopCodons}  The logical value \texttt{TRUE} or \texttt{FALSE}. \texttt{TRUE} (the default): keep all reads, regardless of whether they have stop codons; \texttt{FALSE}: reject reads with stop codons. If \texttt{FALSE} is selected, then the number of stop codons is calculated after attempting to correct frameshift mutations (if applicable).

\textbf{readingFrame} 1, 2, or 3. Only used if \texttt{accept.stop.codons} == \texttt{FALSE}. This specifies the reading frame that is used to determine stop codons. If you use a \texttt{refAminoAcidSeq}, then the frame should always be 1, since all reads will be shifted to frame 1 during frameshift correction. Otherwise, you should select the appropriate reading frame.

\textbf{contigSeq}  The consensus read of all SangerRead S4 instances in DNAString object.

\textbf{alignment}  The alignment of all SangerRead S4 instances with the called consensus sequence in DNAStringSet object. Users can use BrowseSeqs() to view the alignment.

\textbf{differencesDF}  A data frame of the number of pairwise differences between each read and the consensus sequence, as well as the number of bases in each input read that did not contribute to the consensus sequence. It can assist in detecting incorrect reads, or reads with a lot of errors.

\textbf{distanceMatrix}  A distance matrix of genetic distances (corrected with the JC model) between all of the input reads.

\textbf{dendrogram}  A list storing cluster groups in a data frame and a dendrogram object depicting the distance.matrix. Users can use plot() to see the dendrogram.

\textbf{indelsDF}  If users specified a reference sequence via \texttt{refAminoAcidSeq}, then this will be a data frame describing the number of indels and deletions that were made to each of the input reads in order to correct frameshift mutations.

\textbf{stopCodonsDF}  If users specified a reference sequence via \texttt{refAminoAcidSeq}, then this will be a data frame describing the number of stop codons in each read.

\textbf{secondaryPeakDF}  A data frame with one row for each column in the alignment that contained more than one secondary peak. The data frame has three columns: the column number of the alignment; the number of secondary peaks in that column; and the bases (with IUPAC ambiguity codes representing secondary peak call) in that column represented as a string.

\textbf{Author(s)}

Kuan-Hao Chao
Examples

```r
## Simple example
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, "Allolobophora_chlorotica", "RBNII")
contigName <- "Achl_RBNII384-13"
REGEX_SuffixForward <- "\_[0-9]*_F.ab1$"
REGEX_SuffixReverse <- "\_[0-9]*_R.ab1$"
sangerContig <- new("SangerContig",
  ABIF_Directory = parentDir,
  contigName = contigName,
  REGEX_SuffixForward = REGEX_SuffixForward,
  REGEX_SuffixReverse = REGEX_SuffixReverse)

## forward / reverse reads match error
## Input From ABIF file format (Regex)
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, "Allolobophora_chlorotica", "ACHLO")
contigName <- "Achl_ACHLO006-09"
REGEX_SuffixForward <- "\_[0-9]*_F.ab1$"
REGEX_SuffixReverse <- "\_[0-9]*_R.ab1$"
sangerContig <- new("SangerContig",
  inputSource = "ABIF",
  processMethod = "REGEX",
  ABIF_Directory = parentDir,
  contigName = contigName,
  REGEX_SuffixForward = REGEX_SuffixForward,
  REGEX_SuffixReverse = REGEX_SuffixReverse,
  refAminoAcidSeq = "IHWYPLFTGLTLNNKWLKSHFIIMFIGVNLTFFPQHFLGLAGMPRRYSDYPDAYTTWNIVSTIGSTISLLGILFFFFIIWESLVSQRQVIYPIQLNSSIEWYQNTPPAEHSYSELPLLTN",
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 0.0001,
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE,
  minReadsNum = 2,
  processorsNum = 2)

## Input From ABIF file format (Csv three column method)
rawDataDir <- system.file("extdata", package = "sangeranalyseR")
parentDir <- file.path(rawDataDir, "Allolobophora_chlorotica", "RBNII")
CSV_NamesConversion <- file.path(rawDataDir, "ab1", "SangerContig", "names_conversion_2.csv")
sangerContig <- new("SangerContig",
  inputSource = "ABIF",
  processMethod = "CSV",
  ABIF_Directory = parentDir,
  contigName = "Achl_RBNII384-13",
  CSV_NamesConversion = CSV_NamesConversion,
  refAminoAcidSeq = "SRQWLFSTNKDIGTFYFGAWGVMGTSILIRAELGHPGALIGDDQIYNYVITAHAFIMIFFMVMPPIMGGFNG",
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 0.00001,
  baseNumPerRow = 100,
  heightPerRow = 200,
```
## Description

A SangerContig method which generates final reports of the SangerContig instance.

## Usage

```r
# S4 method for signature 'SangerContig'
generateReportSC(
  object,
  outputDir,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE,
  processorsNum = 2)
```
Arguments

- **object**: A SangerContig S4 instance.
- **outputDir**: The output directory of the generated HTML report.
- **includeSangerRead**: The parameter that decides whether to include SangerRead level report. The value is TRUE or FALSE and the default is TRUE.
- **colors**: A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
- **navigationAlignmentFN**: The internal parameter passed to HTML report. Users should not modify this parameter on their own.

Value

The output absolute path to the SangerContig’s HTML file.

Examples

```r
data("sangerContigData")
## Not run:
generateReportSC(sangerContigData)
generateReportSC(sangerContigData, colors="cb_friendly")
## End(Not run)
```

Description

A SangerContig method which launches Shiny app for SangerContig instance.

Usage

```r
## S4 method for signature 'SangerContig'
launchAppSC(object, outputDir = NULL, colors = "default")
```
**Arguments**

- **object**: A SangerContig S4 instance.
- **outputDir**: The output directory of the saved new SangerContig S4 instance.
- **colors**: A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.

**Value**

A shiny.appobj object.

**Examples**

data("sangerContigData")
RShinySC <- launchAppSC(sangerContigData)
RShinySC <- launchAppSC(sangerContigData, colors="cb_friendly")

---

**Description**

A SangerContig method which generates summary table for SangerContig instance

**Usage**

```r
## S4 method for signature 'SangerContig'
readTable(object, indentation = 0)
```

**Arguments**

- **object**: A SangerContig S4 instance.
- **indentation**: The indentation for different level printing.

**Value**

None
### Description

A SangerContig method which updates QualityReport parameter for each the SangerRead instance inside SangerContig.

### Usage

```r
## S4 method for signature 'SangerContig'
updateQualityParam(
  object,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  processorsNum = NULL
)
```

### Arguments

- **object**: A SangerContig S4 instance.
- **TrimmingMethod**: The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.
- **M1TrimmingCutoff**: The trimming cutoff for the Method 1. If TrimmingMethod is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.
- **M2CutoffQualityScore**: The trimming cutoff quality score for the Method 2. If TrimmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.
M2SlidingWindowSize
The trimming sliding window size for the Method 2. If TrimmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.

processorsNum
The number of processors to use, or NULL (the default) for all available processors.

Value
A SangerContig instance.

Examples

data("sangerContigData")
## Not run:
updateQualityParam(sangerContigData,
TrimmingMethod = "M2",
M1TrimmingCutoff = NULL,
M2CutoffQualityScore = 40,
M2SlidingWindowSize = 15)
## End(Not run)

Description
A SangerContig method which writes sequences into Fasta files.

Usage

## S4 method for signature 'SangerContig'
writeFastaSC(
    object, outputDir = NULL, compress = FALSE, compression_level = NA, selection = "all"
)

Arguments

- **object**: A SangerContig S4 instance.
- **outputDir**: The output directory of generated FASTA files.
- **compress**: Like for the save function in base R, must be TRUE or FALSE (the default), or a single string specifying whether writing to the file is to use compression. The only type of compression supported at the moment is "gzip". This parameter will be passed to writeXStringSet function in Biostrings package.
compression_level

This parameter will be passed to `writeXStringSet` function in Biostrings package.

selection

This value can be `all`, `reads_alignment`, `reads_unalignment` or `contig`. It generates reads and the contig FASTA files.

Value

The output directory of FASTA files.

Examples

```r
data("sangerContigData")
writeFastaSC(sangerContigData)
```

---

tables

<table>
<thead>
<tr>
<th>sangerContigData</th>
<th>SangerContig instance</th>
</tr>
</thead>
</table>

Description

SangerContig instance

Usage

```r
data(sangerContigData)
```

Author(s)

Kuan-Hao Chao

---

<table>
<thead>
<tr>
<th>SangerRead</th>
<th>SangerRead</th>
</tr>
</thead>
</table>

Description

the wrapper function for SangerRead
Usage

```r
SangerRead(
  printLevel = "SangerRead",
  inputSource = "ABIF",
  readFeature = "",
  readFileName = "",
  fastaReadName = NULL,
  geneticCode = GENETIC_CODE,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  baseNumPerRow = 100,
  heightPerRow = 200,
  signalRatioCutoff = 0.33,
  showTrimmed = TRUE
)
```

Arguments

- **inputSource**: The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".
- **readFeature**: The direction of the Sanger read. The value must be "Forward Read" or "Reverse Read".
- **readFileName**: The filename of the target ABIF file.
- **fastaReadName**: If `inputSource` is "FASTA", then this value has to be the name of the read inside the FASTA file; if `inputSource` is "ABIF", then this value is "" by default.
- **geneticCode**: Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.
- **TrimmingMethod**: The read trimming method for this SangerRead. The value must be "M1" (the default) or "M2". M1 is the modified Mott's trimming algorithm that can also be found in Phred/Phrap and Biopython. M2 is like trimomatic's sliding window method.
- **M1TrimmingCutoff**: The trimming cutoff for the Method 1. If `TrimmingMethod` is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.
- **M2CutoffQualityScore**: The trimming cutoff quality score for the Method 2. If `TrimmingMethod` is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with `M2SlidingWindowSize`.
- **M2SlidingWindowSize**: The trimming sliding window size for the Method 2. If `TrimmingMethod` is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with `M2CutoffQualityScore`.
baseNumPerRow  It defines maximum base pairs in each row.  The default value is 100.
heightPerRow   It defines the height of each row in chromatogram.  The default value is 200.
signalRatioCutoff  The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is 0.33.
showTrimmed    The logical value storing whether to show trimmed base pairs in chromatogram. The default value is TRUE.

Value
A SangerRead instance.

Author(s)
Kuan-Hao Chao

Examples
inputFilesPath <- system.file("extdata/", package = "sangeranalyseR")
A_chloroticaFdFN <- file.path(inputFilesPath,
                             "Allolobophora_chlorotica",
                             "ACHLO",
                             "Ach1_ACHLO0006-09_1_F.ab1")
sangerRead <- SangerRead(
    printLevel = "SangerRead",
    inputSource = "ABIF",
    readFeature = "Forward Read",
    readFileName = A_chloroticaFdFN,
    geneticCode = GENETIC_CODE,
    TrimmingMethod = "M1",
    M1TrimmingCutoff = 0.0001,
    M2CutoffQualityScore = NULL,
    M2SlidingWindowSize = NULL,
    baseNumPerRow = 100,
    heightPerRow = 200,
    signalRatioCutoff = 0.33,
    showTrimmed = TRUE)

SangerRead-class  SangerRead

Description
An S4 class extending sangerseq S4 class which corresponds to a single ABIF file in Sanger sequencing.
Slots

- **objectResults**: This is the object that stores all information of the creation result.
- **inputSource**: The input source of the raw file. It must be "ABIF" or "FASTA". The default value is "ABIF".
- **readFeature**: The direction of the Sanger read. The value must be "Forward Read" or "Reverse Read".
- **readFileName**: The filename of the target input file.
- **fastaReadName**: If **inputSource** is "FASTA", then this value has to be the name of the read inside the FASTA file; if **inputSource** is "ABIF", then this value is NULL by default.
- **geneticCode**: Named character vector in the same format as GENETIC_CODE (the default), which represents the standard genetic code. This is the code with which the function will attempt to translate your DNA sequences. You can get an appropriate vector with the getGeneticCode() function. The default is the standard code.
- **abifRawData**: An S4 class containing all fields in the ABIF file. It is the abif class defined in sangerseqR package.
- **QualityReport**: A S4 class containing quality trimming related inputs and trimming results.
- **ChromatogramParam**: A S4 class containing chromatogram inputs.
- **primaryAASeqS1**: A polypeptide translated from primary DNA sequence starting from the first nucleic acid.
- **primaryAASeqS2**: A polypeptide translated from primary DNA sequence starting from the second nucleic acid.
- **primaryAASeqS3**: A polypeptide translated from primary DNA sequence starting from the third nucleic acid.
- **primarySeqRaw**: The raw primary sequence from sangerseq class in sangerseqR package before base calling.
- **secondarySeqRaw**: The raw secondary sequence from sangerseq class in sangerseqR package before base calling.
- **peakPosMatrixRaw**: The raw peak position matrix from sangerseq class in sangerseqR package before base calling.
- **peakAmpMatrixRaw**: The raw peak amplitude matrix from sangerseq class in sangerseqR package before base calling.

Author(s)

Kuan-Hao Chao

Examples

```r
## Simple example
inputFilesPath <- system.file("extdata/", package = "sangeranalyseR")
A_chloroticaFFN <- file.path(inputFilesPath,
    "Allolobophora_chlorotica",
    "ACHLO",
    "Ach1_ACHLO006-09_1_F.ab1")
```
sangerReadF <- new("SangerRead",
readFeature = "Forward Read",
readFileName = A_chloroticaFFN)

## Input From ABIF file format
# Forward Read
A_chloroticaFFN <- file.path(inputFilesPath,
  "Allolobophora_chlorotica",
  "ACHLO",
  "Achl_ACHLO006-09_1_F.ab1")

sangerReadF <- new("SangerRead",
  printLevel = "SangerRead",
inSource = "ABIF",
readFeature = "Forward Read",
readFileName = A_chloroticaFFN,
fastaReadName = NULL,
geneticCode = GENETIC_CODE,
TrimmingMethod = "M1",
M1TrimmingCutoff = 0.0001,
M2CutoffQualityScore = NULL,
M2SlidingWindowSize = NULL,
baseNumPerRow = 100,
heightPerRow = 200,
signalRatioCutoff = 0.33,
showTrimmed = TRUE)

# Reverse Read
A_chloroticaRFN <- file.path(inputFilesPath,
  "Allolobophora_chlorotica",
  "ACHLO",
  "Achl_ACHLO006-09_2_R.ab1")

sangerReadR <- new("SangerRead",
inSource = "ABIF",
readFeature = "Reverse Read",
readFileName = A_chloroticaRFN,
geneticCode = GENETIC_CODE,
TrimmingMethod = "M1",
M1TrimmingCutoff = 0.0001,
M2CutoffQualityScore = NULL,
M2SlidingWindowSize = NULL,
baseNumPerRow = 100,
heightPerRow = 200,
signalRatioCutoff = 0.33,
showTrimmed = TRUE)

## Input From FASTA file format
# Forward Read
inputFilesPath <- system.file("extdata/", package = "sangeranalyseR")
A_chloroticaFFNfa <- file.path(inputFilesPath,
  "fasta",
  "SangerRead",
  "Achl_ACHLO006-09_1_F.fa")
readNameFfa <- "Achl_ACHLO0006-09_1_F"
sangerReadFfa <- new("SangerRead",
    inputSource = "FASTA",
    readFeature = "Forward Read",
    readFileName = A_chloroticaFFNfa,
    fastaReadName = readNameFfa,
    geneticCode = GENETIC_CODE)

# Reverse Read
A_chloroticaRFNfa <- file.path(inputFilesPath,
    "fasta",
    "SangerRead",
    "Achl_ACHLO0006-09_2_R.fa")
readNameRfa <- "Achl_ACHLO0006-09_2_R"
sangerReadRfa <- new("SangerRead",
    inputSource = "FASTA",
    readFeature = "Reverse Read",
    readFileName = A_chloroticaRFNfa,
   fastaReadName = readNameRfa,
    geneticCode = GENETIC_CODE)

---

Description

A SangerRead method which generates final reports of the SangerRead instance.

Usage

```r
## S4 method for signature 'SangerRead'
generateReportSR(
    object, 
    outputDir, 
    colors, 
    navigationContigFN = NULL, 
    navigationAlignmentFN = NULL
)
```

Arguments

- **object**: A SangerRead S4 instance.
- **outputDir**: The output directory of the generated HTML report.
- **colors**: A vector for users to set the colors of (A, T, C, G, else). There are three options for users to choose from. 1. "default": (green, blue, black, red, purple). 2. "cb_friendly": ((0, 0, 0), (199, 199, 199), (0, 114, 178), (213, 94, 0), (204, 121, 167)). 3. Users can set their own colors with a vector with five elements.
navigationContigFN
The internal parameter passed to HTML report. Users should not modify this parameter on their own.

navigationAlignmentFN
The internal parameter passed to HTML report. Users should not modify this parameter on their own.

Value
The output absolute path to the SangerRead's HTML file.

Examples

```
data("sangerReadFData")
## Not run:
generateReportSR(sangerReadFData, "~/Documents")
generateReportSR(sangerReadFData, colors="cb_friendly")
## End(Not run)
```

---

SangerRead-class-MakeBaseCalls

### MakeBaseCalls

**Description**

A SangerRead method which does base calling on SangerRead instance

**Usage**

```r
## S4 method for signature 'SangerRead'
MakeBaseCalls(object, signalRatioCutoff = 0.33)
```

**Arguments**

- **object**
  - A SangerRead S4 instance.
- **signalRatioCutoff**
  - The ratio of the height of a secondary peak to a primary peak. Secondary peaks higher than this ratio are annotated. Those below the ratio are excluded. The default value is 0.33.

**Value**

A SangerRead instance.

**Examples**

```
data("sangerReadFData")
newSangerReadFData <- MakeBaseCalls(sangerReadFData, signalRatioCutoff = 0.22)
```
Description

A SangerRead method which creates quality base interactive plot.

Usage

```r
## S4 method for signature 'SangerRead'
qualityBasePlot(object)
```

Arguments

- `object`: A SangerRead S4 instance.

Value

A quality plot.

Examples

```r
data("sangerReadFData")
## Not run:
qualityBasePlot(sangerReadFData)
## End(Not run)
```

Description

A SangerRead method which generates summary table for SangerRead instance

Usage

```r
## S4 method for signature 'SangerRead'
readTable(object, indentation = 0)
```

Arguments

- `object`: A SangerRead S4 instance.
- `indentation`: The indentation for different level printing.
Value
None

Examples

```r
data(sangerReadFData)
data(sangerContigData)
data(sangerAlignmentData)
## Not run:
readTable(sangerReadFData)
readTable(sangerContigData)
readTable(sangerAlignmentData)
## End(Not run)
```

Description
A SangerRead method which updates QualityReport parameter inside the SangerRead.

Usage

```r
## S4 method for signature 'SangerRead'
updateQualityParam(
  object,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL
)
```

Arguments

- **object**: A SangerRead S4 instance.
- **TrimmingMethod**: The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.
- **M1TrimmingCutoff**: The trimming cutoff for the Method 1. If TrimmingMethod is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.
- **M2CutoffQualityScore**: The trimming cutoff quality score for the Method 2. If TrimmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.
M2SlidingWindowSize

The trimming sliding window size for the Method 2. If TrimmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.

Value

A SangerRead instance.

Examples

data("sangerReadFData")
updateQualityParam(sangerReadFData,
  TrimmingMethod = "M2",
  M1TrimmingCutoff = NULL,
  M2CutoffQualityScore = 40,
  M2SlidingWindowSize = 15)

Description

A SangerRead method which writes the sequence into Fasta files.

Usage

## S4 method for signature 'SangerRead'
writeFastaSR(
  object,
  outputDir = NULL,
  compress = FALSE,
  compression_level = NA
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A SangerRead S4 instance.</td>
</tr>
<tr>
<td>outputDir</td>
<td>The output directory of the generated FASTA file.</td>
</tr>
<tr>
<td>compress</td>
<td>Like for the save function in base R, must be TRUE or FALSE (the default),</td>
</tr>
<tr>
<td></td>
<td>or a single string specifying whether writing to the file is to use</td>
</tr>
<tr>
<td></td>
<td>compression. The only type of compression supported at the moment is &quot;gzip&quot;.</td>
</tr>
<tr>
<td></td>
<td>This parameter will be passed to writeXStringSet function in Biostrings</td>
</tr>
<tr>
<td>compression_level</td>
<td>This parameter will be passed to writeXStringSet function in Biostrings</td>
</tr>
<tr>
<td></td>
<td>package.</td>
</tr>
</tbody>
</table>


Value

The output absolute path to the FASTA file.

Examples

data("sangerReadFData")
writeFastaSR(sangerReadFData)

Description

SangerRead instance

Usage

data(sangerReadFData)

Author(s)

Kuan-Hao Chao

Description

Method updateQualityParam

Usage

updateQualityParam(
  object,
  TrimmingMethod = "M1",
  M1TrimmingCutoff = 1e-04,
  M2CutoffQualityScore = NULL,
  M2SlidingWindowSize = NULL,
  ...
)
Arguments

- **object**: A QualityReport, SangerRead, SangerContig, or SangerAlignment S4 instance.
- **TrimmingMethod**: The read trimming method for this SangerRead. The value must be "M1" (the default) or 'M2'.
- **M1TrimmingCutoff**: The trimming cutoff for the Method 1. If TrimmmingMethod is "M1", then the default value is 0.0001. Otherwise, the value must be NULL.
- **M2CutoffQualityScore**: The trimming cutoff quality score for the Method 2. If TrimmmingMethod is 'M2', then the default value is 20. Otherwise, the value must be NULL. It works with M2SlidingWindowSize.
- **M2SlidingWindowSize**: The trimming sliding window size for the Method 2. If TrimmmingMethod is 'M2', then the default value is 10. Otherwise, the value must be NULL. It works with M2CutoffQualityScore.

... Further updateQualityParam-related parameters.

Value

A QualityReport, SangerRead, SangerContig, or SangerAlignment instance.

Examples

```r
data(qualityReportData)
data(sangerReadFData)
data(sangerContigData)
data(sangerAlignmentData)
## Not run:
updateQualityParam(qualityReportData, 
  TrimmingMethod = "M2", 
  M1TrimmingCutoff = NULL, 
  M2CutoffQualityScore = 40, 
  M2SlidingWindowSize = 15)
updateQualityParam(sangerReadFData, 
  TrimmingMethod = "M2", 
  M1TrimmingCutoff = NULL, 
  M2CutoffQualityScore = 40, 
  M2SlidingWindowSize = 15)
updateQualityParam(sangerContigData, 
  TrimmingMethod = "M2", 
  M1TrimmingCutoff = NULL, 
  M2CutoffQualityScore = 40, 
  M2SlidingWindowSize = 15)
updateQualityParam(sangerAlignmentData, 
  TrimmingMethod = "M2", 
  M1TrimmingCutoff = NULL, 
  M2CutoffQualityScore = 40, 
  M2SlidingWindowSize = 15)
## End(Not run)
```
writeFasta

Method writeFasta

Description

A method which writes FASTA files of the SangerRead, SangerContig, and SangerAlignment instance.

Usage

writeFasta(
  object,
  outputDir = NULL,
  compress = FALSE,
  compression_level = NA,
  selection = "all"
)

Arguments

object A SangerRead, SangerContig, or SangerAlignment S4 instance.
outputDir The output directory of generated FASTA files.
compress Like for the save function in base R, must be TRUE or FALSE (the default), or a single string specifying whether writing to the file is to use compression. The only type of compression supported at the moment is "gzip". This parameter will be passed to writeXStringSet function in Biostrings package.
compression_level This parameter will be passed to writeXStringSet function in Biostrings package.
selection This parameter will be passed to writeFastaSC or writeFastaSA.

Value

A SangerRead, SangerContig, or SangerAlignment object.

Author(s)

Kuan-Hao Chao

Examples

data(sangerReadFData)
data(sangerContigData)
data(sangerAlignmentData)
## Not run:
writeFasta(sangerReadFData)
writeFasta(sangerContigData)
writeFastaSA

writeFasta(sangerAlignmentData)
## End(Not run)

writeFastaSA

Method writeFastaSA

Description

Method writeFastaSA

Usage

writeFastaSA(
  object,
  outputDir = NULL,
  compress = FALSE,
  compression_level = NA,
  selection = "all"
)

Arguments

object A SangerAlignment S4 instance.
outputDir The output directory of generated FASTA files.
compress Like for the save function in base R, must be TRUE or FALSE (the default), or a
  single string specifying whether writing to the file is to use compression. The
  only type of compression supported at the moment is "gzip". This parameter
  will be passed to writeXStringSet function in Biostrings package.
compression_level This parameter will be passed to writeXStringSet function in Biostrings pack-
  age.
selection This value can be all, contigs_alignment, contigs_unalignment or all_reads.
  It generates reads and contigs FASTA files.

Value

The output directory of FASTA files.

Examples

data(sangerAlignmentData)
writeFastaSA(sangerAlignmentData)
Method writeFastaSC

Usage

```r
writeFastaSC(
  object,
  outputDir = NULL,
  compress = FALSE,
  compression_level = NA,
  selection = "all"
)
```

Arguments

- **object**: A SangerContig S4 instance.
- **outputDir**: The output directory of generated FASTA files.
- **compress**: Like for the `save` function in base R, must be `TRUE` or `FALSE` (the default), or a single string specifying whether writing to the file is to use compression. The only type of compression supported at the moment is "gzip". This parameter will be passed to `writeXStringSet` function in Biostrings package.
- **compression_level**: This parameter will be passed to `writeXStringSet` function in Biostrings package.
- **selection**: This value can be `all`, `reads_alignment`, `reads_unalignment` or `contig`. It generates reads and the contig FASTA files.

Value

The output directory of FASTA files.

Examples

```r
data(sangerContigData)
writeFastaSC(sangerContigData)
```
Method writeFastaSR

Usage

```r
writeFastaSR(
  object,
  outputDir = NULL,
  compress = FALSE,
  compression_level = NA
)
```

Arguments

- `object` A SangerRead S4 instance.
- `outputDir` The output directory of the generated FASTA file.
- `compress` Like for the save function in base R, must be TRUE or FALSE (the default), or a single string specifying whether writing to the file is to use compression. The only type of compression supported at the moment is "gzip". This parameter will be passed to `writeXStringSet` function in Biostrings package.
- `compression_level` This parameter will be passed to `writeXStringSet` function in Biostrings package.

Value

The output absolute path to the FASTA file.

Examples

```r
data(sangerReadFData)
writeFastaSR(sangerReadFData)
```
# Index

*datasets*
- qualityReportData, 15
- sangerAlignmentData, 26
- sangerContigData, 38
- sangerReadFData, 48

ChromatogramParam-class, 3

generateReport, 4
generateReportSA, 5
generateReportSA, SangerAlignment-method
  (SangerAlignment-class-generateReportSA), 22

generateReportSC, 6
generateReportSC, SangerContig-method
  (SangerContig-class-generateReportSC), 33

generateReportSR, 7
generateReportSR, SangerRead-method
  (SangerRead-class-generateReportSR), 43

launchApp, 8
launchAppSA, 9
launchAppSA, SangerAlignment-method
  (SangerAlignment-class-launchAppSA), 23
launchAppSC, 9
launchAppSC, SangerContig-method
  (SangerContig-class-launchAppSC), 34

MakeBaseCalls, 10
MakeBaseCalls, SangerRead-method
  (SangerRead-class-MakeBaseCalls), 44

ObjectResults-class, 11
qualityBasePlot, 11

qualityBasePlot, QualityReport-method
  (QualityReport-class-qualityBasePlot), 13
qualityBasePlot, SangerRead-method
  (SangerRead-class-qualityBasePlot), 45

QualityReport-class, 12
QualityReport-class-qualityBasePlot, 13
QualityReport-class-updateQualityParam, 14

readTable, 15
readTable, SangerContig-method
  (SangerContig-class-readTable), 35
readTable, SangerRead-method
  (SangerRead-class-readTable), 45

SangerAlignment, 16
SangerAlignment-class, 19
SangerAlignment-class-generateReportSA, 22
SangerAlignment-class-launchAppSA, 23
SangerAlignment-class-updateQualityParam, 24
SangerAlignment-class-writeFastaSA, 25
sangerAnalyseR, 26
SangerContig, 26
SangerContig-class, 30
SangerContig-class-generateReportSC, 33
SangerContig-class-launchAppSC, 34
SangerContig-class-readTable, 35
SangerContig-class-updateQualityParam, 36
SangerContig-class-writeFastaSC, 37
INDEX

sangerContigData, 38
SangerRead, 38
SangerRead-class, 40
SangerRead-class-generateReportSR, 43
SangerRead-class-MakeBaseCalls, 44
SangerRead-class-qualityBasePlot, 45
SangerRead-class-readTable, 45
SangerRead-class-updateQualityParam, 46
SangerRead-class-writeFastaSR, 47
sangerReadFData, 48
updateQualityParam, 48
updateQualityParam, QualityReport-method
  (QualityReport-class-updateQualityParam), 14
updateQualityParam, SangerAlignment-method
  (SangerAlignment-class-updateQualityParam), 24
updateQualityParam, SangerContig-method
  (SangerContig-class-updateQualityParam), 36
updateQualityParam, SangerRead-method
  (SangerRead-class-updateQualityParam), 46

writeFasta, 50
writeFastaSA, 51
writeFastaSA, SangerAlignment-method
  (SangerAlignment-class-writeFastaSA), 25
writeFastaSC, 52
writeFastaSC, SangerContig-method
  (SangerContig-class-writeFastaSC), 37
writeFastaSR, 53
writeFastaSR, SangerRead-method
  (SangerRead-class-writeFastaSR), 47