Package 'sangerseqR'

November 12, 2026

Type Package

Title Tools for Sanger Sequencing Data in R

Version 1.48.0

Date 2024-4-23

Author Jonathon T. Hill, Bradley Demarest

Maintainer Jonathon Hill <jhill@byu.edu>

VignetteBuilder knitr

Imports methods, shiny

Depends R (>= 3.5.0), Biostrings, pwalign, stringr

Suggests BiocStyle, knitr, RUnit, BiocGenerics

Description This package contains several tools for analyzing Sanger Sequencing data files in R, including reading .scf and .ab1 files, making basecalls and plotting chromatograms.

License GPL-2

biocViews Sequencing, SNP, Visualization

RoxygenNote 7.3.1

Encoding UTF-8

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

git_branch RELEASE_3_23

git_last_commit 67e49b6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

sangerseqR-package 2
abif-class 2
chromatogram 3

makeBaseCalls 5
PolyPeakParser 6
primarySeqID 7
read.abif 9
read.scf 10
readsangerseq 11
sangerseq-class 11
scf-class 13
setAllelePhase 14

Index **16**

`sangerseqR-package` *Tools for Sanger Sequencing Data in R*

Description

This package contains several tools for analyzing Sanger Sequencing data files in R, including reading .scf and .ab1 files, making basecalls and plotting chromatograms.

Author(s)

Jonathon T. Hill, Bradley Demarest

Maintainer: Jonathon Hill <jhill@genetics.utah.edu>

References

Jonathon T Hill, Bradley L Demarest, Brent W Bisgrove, Yi-chu Su, Megan Smith and H. Joseph Yost (2014). Poly peak parser: Method and software for identification of unknown indels using sanger sequencing of polymerase chain reaction products. Developmental Dynamics 243:1632-1636

See Also

Biostrings

`abif-class` *ABIF Class Objects*

Description

S4 object returned by `read.abif` containing all fields in the ABIF file format (see `http://home.appliedbiosystems.com/support/software_community/ABIF_File_Format.pdf`). Data fields vary by machine and basecaller versions. Must be converted to `sangerseq` to be used in other functions from this package.

Slots

`header` Header information from the file.

`directory` Directory information from file containing field names and information for reading binary data.

`data` List object containing all data fields and values in file. Included fields vary by machine and basecaller versions.

See Also

`read.abif`, `scf`, `sangerseq`

Examples

```
hetab1 <- read.abif(system.file("extdata",
                                "heterozygous.ab1",
                                package = "sangerseqR"))
str(hetab1)
```

`chromatogram` *Generate Chromatogram*

Description

Generates a chromatogram from a `sangerseq` class object.

Usage

```
chromatogram(
  obj,
  trim5 = 0,
  trim3 = 0,
  showcalls = c("primary", "secondary", "both", "none"),
  width = 500,
  height = NA,
  cex.mtext = 1,
  cex.base = 1,
  ylim = 2,
  filename = NULL,
  showtrim = FALSE,
  showhets = TRUE
)

## S4 method for signature 'sangerseq'
chromatogram(
  obj,
  trim5 = 0,
```

```
  trim3 = 0,
  showcalls = c("primary", "secondary", "both", "none"),
  width = 100,
  height = 2,
  cex.mtext = 1,
  cex.base = 1,
  ylim = 3,
  filename = NULL,
  showtrim = FALSE,
  showhets = TRUE
)
```

Arguments

obj	sangerseq class object
trim5	Number of bases to trim from the beginning of the sequence.
trim3	Number of bases to trim from the end of the sequence.
showcalls	Which basecall sequence to show. Any value other than "primary", "secondary" or "both" will result in basecalls not being shown.
width	Approximate number of bases per line.
height	Height of each plot row. Ignored by some devices.
cex.mtext	Size factor for the text in the margins.
cex.base	Size factor for the basecall text.
ylim	Peaks larger than this many times the IQR larger than the median will be cutoff.
filename	Name of PDF file to save to. A "NULL" value outputs the chromatogram to the current device.
showtrim	If true, highlights trimmed region instead of hiding it.
showhets	Whether or not to highlight heterozygous positions.

Details

This function outputs a chromatogram to the current device or to a PDF file (filename is not NULL). Primary, Secondary or both basecalls can be shown if they are contained in the sangerseq object provided. What is shown will depend on how they were generated. If generated and provided by the ABIF or SCF file, then it will show the primary calls on the first line and the secondary calls on the second line. If generated by makeBaseCalls, then they will show the maximum and alternate basecalls only for heterozygous peaks. Finally, if the setAllelePhase has been run on the object, then the first line is the reference sequence and the second line is the alternate allele.

The range of the trace data shown is trimmed to the called sequence by default. Setting trim5 and trim3 to NULL will show the complete trace 5' and 3' of the called bases, respectively. This will generally create a very long trace. Conversely, setting trim5 and trim3 to an integer > 0 will hide the data corresponding to that number of bases at each end. For example, trim5=10 and trim3=20 will remove 10 bases from the 5' end and 20 bases from the 3' end.

Several output parameters can also be set to control how the figure appears. However, it should be noted that if the current device is too small, R will return an error and not show the chromatogram.

This is common with long sequence reads. To bypass this error, we recommend that the user set `filename`. This will cause the chromatogram to be saved to a PDF file in the current working directory.

Value

A plot showing the chromatogram tracedata and, optionally, basecalls.

See Also

`makeBaseCalls`, `setAllelePhase`, `sangerseq`

Examples

```
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
hetcalls <- makeBaseCalls(hetsangerseq, ratio = 0.33)
chromatogram(hetcalls, width = 100, height = 2, trim5 = 50, trim3 = 100,
             showcalls = "both", filename = "chromatogram.pdf")
```

---

makeBaseCalls *Make Basecalls*

Description

Updates a `sangerseq` class object to contain primary and secondary peak calls.

Usage

```
makeBaseCalls(obj, ratio = 0.33)

## S4 method for signature 'sangerseq'
makeBaseCalls(obj, ratio = 0.33)
```

Arguments

`obj`	`sangerseq` class object
`ratio`	cutoff ratio for separating signal and noise. Ratio is relative to maximum peak in basecall window.

Details

Scf files do not contain secondary basecalls and ABIF files sometimes (but not always) contain secondary peak calls that show the secondary peak even if clearly a negative peak. This is problematic in sequence reads where heterozygous sequence data is contained in the chromatogram data. `makeBaseCalls` identifies basecall windows containing more than one peak using the provided cutoff ratio and then makes heterozygous calls for those windows. The primarySeq will always contain the base corresponding to the maximum peak amplitude within the window. The secondaryPeak will have the same base if the peak was classified as a homozygous peak, the base corresponding to the second tallest peak if two peaks were above the cutoff, or an ambiguous base if more than two peaks were identified in the window that are higher than the cutoff ratio.

Value

`sangerseq` s4 object with new basecalls in the primarySeq and secondarySeq fields. Matrix values are also updated to reflect newly called base positions and amplitudes of all peaks within window.

See Also

`chromatogram`, `setAllelePhase`, `sangerseq`

Examples

```
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
hetcalls <- makeBaseCalls(hetsangerseq, ratio = 0.33)
```

`PolyPeakParser` *Run Poly Peak Parser*

Description

Runs the Poly Peak Parser shiny (shiny.rstudio.com) app in the system's default browser. Poly Peak Parser is a web front end that reads, plots and parses double peaks from chromatogram files. Instructions can be found on the webpage once it launches.

Usage

```
PolyPeakParser()
```

Value

`scf` s4 object

See Also

`read.abif`, `readsangerseq`, `scf`

Examples

```
## Not run:
PolyPeakParser()

## End(Not run)
```

---

primarySeqID *Sangerseq Accessor Functions*

Description

Accessor Functions allow the user to retrieve results from and assign values to sangerseq-class objects.

Usage

```
primarySeqID(obj)

primarySeqID(obj) <- value

primarySeq(obj, string = FALSE)

primarySeq(obj) <- value

secondarySeqID(obj)

secondarySeqID(obj) <- value

secondarySeq(obj, string = FALSE)

secondarySeq(obj) <- value

traceMatrix(obj)

traceMatrix(obj) <- value

peakPosMatrix(obj)

peakPosMatrix(obj) <- value

peakAmpMatrix(obj)

peakAmpMatrix(obj) <- value

## S4 method for signature 'sangerseq'
primarySeq(obj, string = FALSE)
```

```
## S4 method for signature 'sangerseq'
secondarySeq(obj, string = FALSE)

## S4 method for signature 'sangerseq'
traceMatrix(obj)

## S4 method for signature 'sangerseq'
peakPosMatrix(obj)

## S4 method for signature 'sangerseq'
peakAmpMatrix(obj)

## S4 method for signature 'sangerseq'
primarySeqID(obj)

## S4 method for signature 'sangerseq'
secondarySeqID(obj)

## S4 replacement method for signature 'sangerseq'
primarySeq(obj) <- value

## S4 replacement method for signature 'sangerseq'
secondarySeq(obj) <- value

## S4 replacement method for signature 'sangerseq'
traceMatrix(obj) <- value

## S4 replacement method for signature 'sangerseq'
peakPosMatrix(obj) <- value

## S4 replacement method for signature 'sangerseq'
peakAmpMatrix(obj) <- value

## S4 replacement method for signature 'sangerseq'
primarySeqID(obj) <- value

## S4 replacement method for signature 'sangerseq'
secondarySeqID(obj) <- value
```

Arguments

obj	sangerseq object to be manipulated
value	The value to set the slot to.
string	TRUE/FALSE. FALSE (default) returns a DNAString class object. TRUE returns the DNA sequence as a character string.

See Also

sangerseq-class

Examples

```
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
primarySeq(hetsangerseq)
secondarySeq(hetsangerseq, string=TRUE)
primarySeqID(hetsangerseq)
primarySeqID(hetsangerseq) <- "Some string"
primarySeqID(hetsangerseq)
```

---

read.abif *Read ABIF Files*

Description

Reads ABIF sanger sequencing data files. ABIF files are a proprietary binary sanger sequencing chromatogram data file created by Applied Biosystems (see http://home.appliedbiosystems.com/support/software_community/ABIF_File_Format.pdf). The file is read and parsed into an abif class object. This method is based on the read.abif function in the seqinr package available on CRAN.

Usage

```
read.abif(filename)
```

Arguments

filename	Location of the file.

Value

abif s4 object

References

Charif, D. and Lobry, J.R. (2007) SeqinR 1.0-2: a contributed package to teh R project for statistical computing devoted to biological sequences retrieval and analysis. Structural approches to sequenc eevolution: Molecules, networks, populations. pp. 207-232.

See Also

read.scf, readsangerseq, abif

Examples

```
hetab1 <- read.abif(system.file("extdata",
                                "heterozygous.ab1",
                                package = "sangerseqR"))
str(hetab1)
```

`read.scf` *Read Scf Files*

Description

Reads Scf sanger sequencing data files. Scf files are an open source binary sanger sequencing chromatogram data file (see http://staden.sourceforge.net/manual/formats_unix_2.html). The file is read and parsed into an `scf` class object.

Usage

```
read.scf(filename)
```

Arguments

`filename`	Location of the file.

Value

`scf` s4 object

See Also

`read.abif`, `readsangerseq`, `scf`

Examples

```
homoscf <- read.scf(system.file("extdata",
                                "homozygous.scf",
                                package = "sangerseqR"))
str(homoscf)
```

readsangerseq *Read Scf or ABIF Files*

Description

This is a convienience function for reading Scf or ABIF files into a sangerseq object, which can be used by the other sangerseq package functions. It is equivalent to calling `read.scf` or `read.abif` as appropriate and then calling `sangerseq`.

Usage

```
readsangerseq(filename)
```

Arguments

`filename`	Location of the file.

Value

`sangerseq` s4 object

See Also

`read.abif`, `read.scf`, `abif`, `scf`, `sangerseq`

Examples

```
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
str(hetsangerseq)
#same for scf files
homosangerseq <- readsangerseq(system.file("extdata",
                                           "homozygous.scf",
                                           package = "sangerseqR"))
str(homosangerseq)
```

sangerseq-class *Sangerseq Class Objects*

Description

Sangerseq Class Objects contain data necessary for using sangerseq package functions (e.g. `chromatogram`, `makeBaseCalls`). The exact content will depend on the source of the data (for example, scf files do not have secondary Basecalls).

Usage

```
sangerseq(obj)

## S4 method for signature 'abif'
sangerseq(obj)

## S4 method for signature 'scf'
sangerseq(obj)
```

Arguments

obj	Can be either an abif or scf object.

Slots

`primarySeqID` Source of the primary basecalls. Functions that modify these calls, such as `makeBaseCalls` and `setAllelePhase` will also change this value.

`secondarySeqID` Source of the secondary basecalls. See above.

`primarySeq` The primary Basecalls formatted as a `DNAString` object.

`secondarySeq` The secondary Basecalls formatted as a `DNAString` object.

`traceMatrix` A numerical matrix containing 4 columns corresponding to the normalized signal values for the chromatogram traces. Column order = A,C,G,T.

`peakPosMatrix` A numerical matrix containing the position of the maximum peak values for each base within each Basecall window. If no peak was detected for a given base in a given window, then "NA". Column order = A,C,G,T.

`peakAmpMatrix` A numerical matrix containing the maximum peak amplitudes for each base within each Basecall window. If no peak was detected for a given base in a given window, then 0. Column order = A,C,G,T.

Accessor methods

`primarySeqID`, `primarySeq`, `secondarySeqID`, `secondarySeq`, `traceMatrix`, `peakPosMatrix`, `peakAmpMatrix`

See Also

`abif`, `scf`

Examples

```
#sample sangerseq object created from abif file
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
str(hetsangerseq)
#same for scf files
homosangerseq <- readsangerseq(system.file("extdata",
                                           "homozygous.scf",
```

```
                                        package = "sangerseqR"))
str(homosangerseq)
```

---

scf-class *Scf Class Objects*

Description

S4 object returned by read.scf containing all fields in the SCF file format (see http://staden.sourceforge.net/manual/formats_unix_2.html). Must be converted to sangerseq to be used in other functions from this package.

Slots

header Header information from the file.

sample_points Trace data matrix (Order = A, C, G, T).

sequence_probs Matrix of the relative probabilities for each base at each position (Order = A, C, G, T).

basecall_positions Vector containing trace matrix indices for each basecall.

basecalls DNAString object containing the basecalls.

comments String containing any comments in the file.

private Raw binary data containing any private data in the file. Generally not used.

See Also

read.scf, abif, sangerseq

Examples

```
homoscf <- read.scf(system.file("extdata",
                                "homozygous.scf",
                                package = "sangerseqR"))
str(homoscf)
```

setAllelePhase *Set Reference and Alternate Alleles*

Description

Parses the Primary and Secondary Sequences into Reference and Alternate Alleles

Usage

```
setAllelePhase(obj, refseq, trim5 = 0, trim3 = 0)

## S4 method for signature 'sangerseq'
setAllelePhase(obj, refseq, trim5 = 0, trim3 = 0)
```

Arguments

obj	sangerseq class object
refseq	DNAString for character string of reference allele sequence.
trim5	Number of bases to trim from the beginning of the sequence.
trim3	Number of bases to trim from the end of the sequence.

Details

When multiple heterozygous basecalls are made, it is generally unclear which calls are in phase with each other. This function takes a reference sequence for one of the alleles to match the primary and secondary basecalls as reference or alternate allele.

Value

A sangerseq object with the Reference Allele in the primarySeq slot and the Alternate Allele in the secondarySeq slot.

See Also

makeBaseCalls, chromatogram, sangerseq

Examples

```
#Load Sequences
hetsangerseq <- readsangerseq(system.file("extdata",
                                          "heterozygous.ab1",
                                          package = "sangerseqR"))
homosangerseq <- readsangerseq(system.file("extdata",
                                           "homozygous.scf",
                                           package = "sangerseqR"))
#Make calls on heterozygous sequence to be parsed
hetcalls <- makeBaseCalls(hetsangerseq, ratio = 0.33)
#Need a reference sequence to set phase. Can get from annotation
```

```
#(e.g. Refseq) or another sanger sequencing file
ref <- subseq(primarySeq(homosangerseq, string = TRUE),
              start = 30,
              width = 500)
#Set the phase
hetseqalleles <- setAllelePhase(hetcalls, ref, trim5 = 50, trim3 = 100)
#Align to compare alleles
pa <- pairwiseAlignment(primarySeq(hetseqalleles),
                                   secondarySeq(hetseqalleles),
                                   type = "global-local")
writePairwiseAlignments(pa)
```

Index

abif, *9*, *11–13*
abif *(*abif-class*)*, 2
abif-class, 2

chromatogram, 3, *6*, *11*, *14*
chromatogram,sangerseq-method *(*chromatogram*)*, 3

DNAString, *8*, *12, 13*

makeBaseCalls, *4, 5*, 5, *11, 12*, *14*
makeBaseCalls,sangerseq-method *(*makeBaseCalls*)*, 5

peakAmpMatrix, *12*
peakAmpMatrix *(*primarySeqID*)*, 7
peakAmpMatrix,sangerseq-method *(*primarySeqID*)*, 7
peakAmpMatrix<- *(*primarySeqID*)*, 7
peakAmpMatrix<-,sangerseq-method *(*primarySeqID*)*, 7
peakPosMatrix, *12*
peakPosMatrix *(*primarySeqID*)*, 7
peakPosMatrix,sangerseq-method *(*primarySeqID*)*, 7
peakPosMatrix<- *(*primarySeqID*)*, 7
peakPosMatrix<-,sangerseq-method *(*primarySeqID*)*, 7
PolyPeakParser, 6
primarySeq, *12*
primarySeq *(*primarySeqID*)*, 7
primarySeq,sangerseq-method *(*primarySeqID*)*, 7
primarySeq<- *(*primarySeqID*)*, 7
primarySeq<-,sangerseq-method *(*primarySeqID*)*, 7
primarySeqID, 7, *12*
primarySeqID,sangerseq-method *(*primarySeqID*)*, 7
primarySeqID<- *(*primarySeqID*)*, 7
primarySeqID<-,sangerseq-method *(*primarySeqID*)*, 7

read.abif, *2, 3*, *6*, 9, *10, 11*
read.scf, *9*, 10, *11*, *13*
readsangerseq, *6*, *9, 10*, 11

sangerseq, *2–6*, *11*, *13, 14*
sangerseq *(*sangerseq-class*)*, 11
sangerseq,abif-method *(*sangerseq-class*)*, 11
sangerseq,scf-method *(*sangerseq-class*)*, 11
sangerseq-class, 11
sangerseqR *(*sangerseqR-package*)*, 2
sangerseqR-package, 2
scf, *3*, *6*, *10–12*
scf *(*scf-class*)*, 13
scf-class, 13
secondarySeq, *12*
secondarySeq *(*primarySeqID*)*, 7
secondarySeq,sangerseq-method *(*primarySeqID*)*, 7
secondarySeq<- *(*primarySeqID*)*, 7
secondarySeq<-,sangerseq-method *(*primarySeqID*)*, 7
secondarySeqID, *12*
secondarySeqID *(*primarySeqID*)*, 7
secondarySeqID,sangerseq-method *(*primarySeqID*)*, 7
secondarySeqID<- *(*primarySeqID*)*, 7
secondarySeqID<-,sangerseq-method *(*primarySeqID*)*, 7
setAllelePhase, *4–6*, *12*, 14
setAllelePhase,sangerseq-method *(*setAllelePhase*)*, 14

traceMatrix, *12*
traceMatrix *(*primarySeqID*)*, 7

traceMatrix,sangerseq-method
(primarySeqID), 7
traceMatrix<- (primarySeqID), 7
traceMatrix<-,sangerseq-method
(primarySeqID), 7