

# Package ‘sangerseqR’

October 8, 2014

**Type** Package

**Title** Tools for Sanger Sequencing Data in R

**Version** 1.0.0

**Date** 2014-02-13

**Author** Jonathon T. Hill, Bradley Demarest

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

**VignetteBuilder** knitr

**Imports** methods

**Depends** R (>= 3.0.2), Biostrings

**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

**Collate** AllClasses.R AllGenerics.R UnExportedFunctions.R  
ReadFileFunctions.R sangerseqmethods.R

**biocViews** Sequencing, SNP, Visualization

## R topics documented:

sangerseqR-package	2
abif-class	2
chromatogram	3
makeBaseCalls	4
primarySeqID	5
read.abif	8
read.scf	9

readsangerseq . . . . .	9
sangerseq-class . . . . .	10
scf-class . . . . .	11
setAllelePhase . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

sangerseqR-package	<i>Tools for Sanger Sequencing Data in R</i>
--------------------	--

---

### Description

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

### Author(s)

Jonathon T. Hill, Bradley Demarest  
 Maintainer: Jonathon Hill <jhill@genetics.utah.edu>

### References

Hill et al. (submitted)

### See Also

Biostrings

---

abif-class	<i>ABIF Class Objects</i>
------------	---------------------------

---

### 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](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	<i>Generate Chromatogram</i>
--------------	------------------------------

---

**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**

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

---

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
string	TRUE/FALSE. FALSE (default) returns a <a href="#">DNAStrng</a> class object. TRUE returns the DNA sequence as a character string.
value	The value to set the slot to.

### 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.scf	<i>Read Scf Files</i>
----------	-----------------------

---

### 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](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	<i>Read Scf or ABIF Files</i>
---------------	-------------------------------

---

### Description

This is a convenience 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 [DNAStrng](#) object.
- `secondarySeq` The secondary Basecalls formatted as a [DNAStrng](#) 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](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 [DNAStrng](#) 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 DNAStrng 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, 8, 10–12
- abif (abif-class), 2
- abif-class, 2
  
- chromatogram, 3, 5, 10, 13
- chromatogram, sangerseq-method (chromatogram), 3
  
- DNAStrng, 7, 11, 12
  
- makeBaseCalls, 4, 4, 10, 11, 13
- makeBaseCalls, sangerseq-method (makeBaseCalls), 4
  
- peakAmpMatrix, 11
- peakAmpMatrix (primarySeqID), 5
- peakAmpMatrix, sangerseq-method (primarySeqID), 5
- peakAmpMatrix<- (primarySeqID), 5
- peakAmpMatrix<-, sangerseq-method (primarySeqID), 5
- peakPosMatrix, 11
- peakPosMatrix (primarySeqID), 5
- peakPosMatrix, sangerseq-method (primarySeqID), 5
- peakPosMatrix<- (primarySeqID), 5
- peakPosMatrix<-, sangerseq-method (primarySeqID), 5
  
- primarySeq, 11
- primarySeq (primarySeqID), 5
- primarySeq, sangerseq-method (primarySeqID), 5
- primarySeq<- (primarySeqID), 5
- primarySeq<-, sangerseq-method (primarySeqID), 5
  
- primarySeqID, 5, 11
- primarySeqID, sangerseq-method (primarySeqID), 5
- primarySeqID<- (primarySeqID), 5
- primarySeqID<-, sangerseq-method (primarySeqID), 5
  
- read.abif, 2, 3, 8, 9, 10
- read.scf, 8, 9, 9–12
- readsangerseq, 8, 9, 9
  
- sangerseq, 2–5, 9–13
- sangerseq (sangerseq-class), 10
- sangerseq, abif-method (sangerseq-class), 10
- sangerseq, scf-method (sangerseq-class), 10
  
- sangerseq-class, 10
- sangerseqR (sangerseqR-package), 2
- sangerseqR-package, 2
  
- scf, 3, 9–11
- scf (scf-class), 11
- scf-class, 11
  
- secondarySeq, 11
- secondarySeq (primarySeqID), 5
- secondarySeq, sangerseq-method (primarySeqID), 5
- secondarySeq<- (primarySeqID), 5
- secondarySeq<- , sangerseq-method (primarySeqID), 5
  
- secondarySeqID, 11
- secondarySeqID (primarySeqID), 5
- secondarySeqID, sangerseq-method (primarySeqID), 5
- secondarySeqID<- (primarySeqID), 5
- secondarySeqID<- , sangerseq-method (primarySeqID), 5
  
- setAllelePhase, 4, 5, 11, 12
- setAllelePhase, sangerseq-method (setAllelePhase), 12
  
- traceMatrix, 11
- traceMatrix (primarySeqID), 5
- traceMatrix, sangerseq-method (primarySeqID), 5
- traceMatrix<- (primarySeqID), 5

`traceMatrix<-`, `sangerseq-method`  
(`primarySeqID`), [5](#)