Package ‘rSFFreader’

April 15, 2017

Type Package

Title rSFFreader reads in sff files generated by Roche 454 and Life Sciences Ion Torrent sequencers

Version 0.22.0

Date 2011-10-02

Author Matt Settles <mattsettles@gmail.com>, Sam Hunter, Brice Sarver, Ilia Zhbannikov, Kyu-Chul Cho

Maintainer Matt Settles <mattsettles@gmail.com>

Description rSFFreader reads sequence, qualities and clip point values from sff files generated by Roche 454 and Life Sciences Ion Torrent sequencers into similar classes as are present for fastq files.

License Artistic-2.0

Depends ShortRead (>= 1.23.17)

Imports methods, Biostrings, IRanges

Suggests xtable

LinkingTo S4Vectors, IRanges, XVector, Biostrings

Collate allClasses.R allGenerics.R methods-Misc.R methods-SffHeader.R methods-SffReads.R methods-SffReadsQ.R readSFF.R

biocViews DataImport, Sequencing

NeedsCompilation yes

R topics documented:

rSFFreader-package *Base classes and methods for high-throughput sequence data stored in SFF files.*

Description

Base classes, functions, and methods for representation of high-throughput sequencing data stored in SFF files (such as Roche 454 Data and Life Sciences Ion Torrent). Implementation defined according to http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=show&f=formats&m=doc&s=formats#sff

Details

See `packageDescription('rSFFreader')`

Author(s)

Matt Settles <msettles@uidaho.edu> and Kyu-chul Cho Maintainer: Matt Settles <msettles@uidaho.edu>

References

Coming Soon

availableClipModes *availableClipModes*

Description

Clip modes are used to store "views" on the sequence object. They are often used to identify adapter sequences and low-quality "ends" which will be trimmed before further analysis. Storing clipping information instead of clipped sequences is useful for avoiding loss of data while maintaining information about the appropriate nucleotides for down-stream analysis. Each clip mode defines a set of left and right clip points, one set for each read. Clip points are typically included in the SFF file, are generated by the sequence provider and are loaded into the appropriate `IRanges` object when the SFF file is loaded via readSff. The vendor-generated clip points are not always desireable however, so accomidations for custom clip points are also provided.

Usage

```
availableClipModes(object)
```

Arguments

`object`	An object of class `SffReads` or `SffReadsQ`.

Details

`availableClipModes` produces a list of clip modes supported by the rSFFreader package and the object being passed to it. These can include:

adapter: defined in the SFF file, and meant to indicate positions of adapter sequence

quality: defined in the SFF file, and meant to indicate positions of low-quality regions of the sequence

full: uses the "interior" of quality and adapter and is the most conservative, equivalant to Roche clip points

raw: no clipping is applied and full length reads are returned

custom: clip points set by the user as an `IRanges` object. (see examples below)

Functions are provided for setting clip mode as well as extracting and setting clip points of each type from a `SffReads` or `SffReadsQ` object. The functions for getting/setting clip points all work in the same way and an example is provided in the examples section below. The functions include:

clipMode gets/sets the adapter clip mode

adapterClip get/set the adapter clip points as an `IRanges` object

customClip get/set the custom clip points as an `IRanges` object

fullClip get/set the full clip points as an `IRanges` object

qualityClip get/set the quality clip points as an `IRanges` object

rawClip get/set the raw clip points as an `IRanges` object

Currently available clipModes returned by `availableClipModes` is dependant on the which clipping slots (qualityIR,adapterIR, and customIR) are set (length != 0).

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
## Load in an example dataset:
sff <- loadIonSampleData()

## Get a list of available clip modes:
availableClipModes(sff)

## Check the current clipMode. It should default to "full":
clipMode(sff)

## full clipping is the most conservative, resulting in shorter reads
hist(width(sff))
summary(width(sff))

## These reads should also be free of adapters although the first base looks suspect:
alphabetByCycle(DNAStringSet(substr(sread(sff), 1,15)), alphabet=c("A","C","T","G"))

cols <- c("green","blue","black","red","darkgrey")
leg <-  c("A","C","T","G","N")
matplot(t(alphabetByCycle(DNAStringSet(substr(sread(sff), 1,15)),
    alphabet=c("A","C","T","G"))), type="l", lty=1, col=cols)
legend("topright", col=cols, legend=leg, pch=18, cex=.8)
```

```
## Compare this to unclipped reads using "raw" mode:
clipMode(sff) <- "raw"
hist(width(sff),breaks=500,col="grey",xlab="Read Length",main="Raw Read Length")

alphabetByCycle(DNAStringSet(substr(sread(sff), 1,15)), alphabet=c("A","C","T","G"))

cols <- c("green","blue","black","red","darkgrey")
leg <-  c("A","C","T","G","N")
matplot(t(alphabetByCycle(DNAStringSet(substr(sread(sff), 1,15)),
    alphabet=c("A","C","T","G"))), type="l", lty=1, col=cols)
legend("topright", col=cols, legend=leg, pch=18, cex=.8)

## Extract clip points for further analysis:
full.clippoints <- fullClip(sff)

raw.clippoints <- rawClip(sff)

table(start(full.clippoints))
table(start(raw.clippoints))

par(mfrow=c(1,2))
hist(end(full.clippoints))
hist(end(raw.clippoints))

par(mfrow=c(1,1))
## determine how much was trimmed from each read by clipping
barplot(table(end(raw.clippoints) - end(full.clippoints)))

## Custom clip points can also be set using an IRanges object:
customClip(sff) <- IRanges(start = 1, end = 4)
clipMode(sff) <- "custom"
table(counts=as.character(sread(sff)))
```

load454SampleData *Load the example 454 dataset*

Description

Load a small, 1000 record example sff file.

Usage

```
load454SampleData()
```

Arguments

None

Value

Loads a small example Roche 454 dataset (sff file) into a `SffReadsQ` object containing 1000 reads.

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
##Read in the included sample 454 data set.
sff <- load454SampleData()
##Generate summary statistics for read lengths.
summary(width(sff))

##Generate some QA plots:
##Read length histograms:
par(mfrow=c(2,2))
clipMode(sff) <- "raw"
hist(width(sff),breaks=500,col="grey",xlab="Read Length",main="Raw Read Length")

clipMode(sff) <- "full"
hist(width(sff),breaks=500,col="grey",xlab="Read Length",main="Clipped Read Length")

## Base by position plots:
clipMode(sff) <- "raw"
ac <- alphabetByCycle(sread(sff),alphabet=c("A","C","T","G","N"))
ac.reads <- apply(ac,2,sum)
acf <- sweep(ac,MARGIN=2,FUN="/",STATS=apply(ac,2,sum))

matplot(cbind(t(acf),ac.reads/ac.reads[1]),col=c("green","blue","black","red","darkgrey","purple"),
          type="l",lty=1,xlab="Base Position",ylab="Base Frequency",
          main="Base by position")
cols <- c("green","blue","black","red","darkgrey","purple")
leg <-  c("A","C","T","G","N","%reads")
legend("topright", col=cols, legend=leg, pch=18, cex=.8)

clipMode(sff) <- "full"
ac <- alphabetByCycle(sread(sff),alphabet=c("A","C","T","G","N"))
ac.reads <- apply(ac,2,sum)
acf <- sweep(ac,MARGIN=2,FUN="/",STATS=apply(ac,2,sum))

matplot(cbind(t(acf),ac.reads/ac.reads[1]),col=c("green","blue","black","red","darkgrey","purple"),
          type="l",lty=1,xlab="Base Position",ylab="Base Frequency",
          main="Base by position")
legend("topright", col=cols, legend=leg, pch=18, cex=.8)
```

`loadIonSampleData` *Load the example Ion Torrent dataset*

Description

Load a small, 1000 record example sff file.

Usage

```
load454SampleData()
```

Arguments

None

Value

Loads a small example Life Sciences Ion Torrent dataset (sff file) into a SffReadsQ object containing 1000 reads.

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
##Read in the included Ion Torrent sample data set.
sff <- loadIonSampleData()
##Generate summary statistics for read lengths.
summary(width(sff))

##Generate some QA plots:
##Read length histograms:
par(mfrow=c(2,2))
clipMode(sff) <- "raw"
hist(width(sff),breaks=500,col="grey",xlab="Read Length",main="Raw Read Length")

clipMode(sff) <- "full"
hist(width(sff),breaks=500,col="grey",xlab="Read Length",main="Clipped Read Length")

## Base by position plots:
clipMode(sff) <- "raw"
ac <- alphabetByCycle(sread(sff),alphabet=c("A","C","T","G","N"))
ac.reads <- apply(ac,2,sum)
acf <- sweep(ac,MARGIN=2,FUN="/",STATS=apply(ac,2,sum))

matplot(cbind(t(acf),ac.reads/ac.reads[1]),col=c("green","blue","black","red","darkgrey","purple"),
          type="l",lty=1,xlab="Base Position",ylab="Base Frequency",
          main="Base by position")
cols <- c("green","blue","black","red","darkgrey","purple")
leg <-  c("A","C","T","G","N","%reads")
legend("topright", col=cols, legend=leg, pch=18, cex=.8)

clipMode(sff) <- "full"
ac <- alphabetByCycle(sread(sff),alphabet=c("A","C","T","G","N"))
ac.reads <- apply(ac,2,sum)
acf <- sweep(ac,MARGIN=2,FUN="/",STATS=apply(ac,2,sum))

matplot(cbind(t(acf),ac.reads/ac.reads[1]),col=c("green","blue","black","red","darkgrey","purple"),
          type="l",lty=1,xlab="Base Position",ylab="Base Frequency",
          main="Base by position")
legend("topright", col=cols, legend=leg, pch=18, cex=.8)
```

readSff *Read SFF-formatted files into SffReads or SffReadsQ objects*

Description

readSff reads all sff files specified in filenames, returning an R object of type SffReads or SffReadsQ which acts and behaves similarly to the ShortRead and ShortReadQ classes from package ShortRead

Usage

```
readSff(filenames, use.qualities=TRUE, use.names=TRUE,
    clipMode = c("full", "adapter", "quality", "raw"), verbose=TRUE)
```

Arguments

filenames	File or files to be read in.
use.qualities	logical(1) indicating whether to include quality values in the returned object.
use.names	logical(1) indicating whether to return read names, not yet implemented
clipMode	character(1) indicating the clipMode of the returned object.
verbose	logical(1) indicating whether to return verbose output.

Value

A SffReads or SffReadsQ object.

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
sff <- readSff(system.file("extdata","Small454Test.sff",package="rSFFreader"))
```

readSffGeometry *Read in basic statistics about an sff file*

Description

readSffGeometry processes a sff file or set of sff files, counting the number of reads and the length of each read.

Usage

```
readSffGeometry(filenames)
```

Arguments

filenames	File or files to be read in.

Value

A list with two elements, nReads and Read_Widths

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
sffgeom <- readSffGeometry(system.file("extdata","Small454Test.sff",package="rSFFreader"))
sffgeom$nReads
sffgeom$Read_Widths
```

readSffHeader *Read in sff header*

Description

readSffHeader Processes an sff file or set of sff files, and store the header information in a SffHeader object.

Usage

```
readSffHeader(filenames, verbose=TRUE)
```

Arguments

filenames File or files to be read in.

verbose logical(1) indicating whether to return verbose output.

Value

A SffHeader object.

Author(s)

Matt Settles <msettles@uidaho.edu>

Examples

```
sffhead <- readSffHeader(system.file("extdata","Small454Test.sff",package="rSFFreader"))
sffhead@header
```

SffHeader-class SffHeader

Description

Class `SffHeader` contains meta-data stored in the header of the SFF files read in.

Objects from this class are the result of `readSffHeader`, or from the result of `readSff`. The resulting object will contain a `header` slot which is a list. If multiple sff files were processed by either of the above functions, this list will contain meta-data about each of these files in corresponding positions in the list.

Meta-data included is defined in the SFF file specifications and include:

filename: The name of the file that was read in.

magic_number: 779314790, which encodes the string ".sff"

version: Version number

index_offset: An optional field which indicates the position of a read index within the file.

index_length: An optional field which indicates the length of a read index within the file.

number_of_reads: Stores the number of reads in the file.

header_length: The number of bytes required by header fields.

key_length: The length of the key sequence used for these reads.

number_of_flows_per_read: The number of flows carried out during sequencing of the reads.

flowgram_format_code: Indicates the format of the flowgram encoding. Currently "1" is the only valid value.

flow_chars: Indicates the nucleotide bases used for each sequencing flow.

key_sequence: Nucleotide sequence use for these reads.

Slots

`header`: Object of class `"list"`, containing data frames or lists of data frames summarizing a description of the SFF files.

Methods

header `signature(object = "SffHeader")`: access the header slot of `object`, returning a list object.

Author(s)

Matt Settles <msettles@uidaho.edu>

See Also

`SffReads`,`SffReadsQ`.

Examples

```
showClass("SffHeader")

## The readSffHeader can be used to extract header information from one or more sff files:
sffFiles = c(system.file("extdata", "SmallTorrentTest.sff", package = "rSFFreader"),
             system.file("extdata","Small454Test.sff",package="rSFFreader"))
header <- readSffHeader(sffFiles)
header
header(header)[[1]]$number_of_reads
header(header)[[2]]$number_of_reads

## Header information is also retrieved when using readSff:
sff <- readSff(sffFiles)

## Number of flows for dataset 1 and 2:
header(sff)[[1]]$number_of_flows_per_read
header(sff)[[2]]$number_of_flows_per_read
```

SffReads-class *Class "SffReads". A class for next generation reads stored in SFF files (Roche 454 and Life Sciences Ion Torrent)*

Description

This class provides a way to store and manipulate, in a coordinated fashion, next generation reads stored in SFF files and their identifiers.

Objects from this class are created by readSff, or by calls to the constructor SffReads, as outlined below.

Usage

```
## Constructor:

SffReads(sread, qualityIR, adapterIR, customIR, clipMode="raw", header)
```

Arguments

sread an object of type "DNAStringSet" object representing the sequence data.

qualityIR an object of type "IRanges" object, specifying the quality clip points.

adapterIR an object of type "IRanges" object, specifying the adapter clip points.

customIR an object of type "IRanges" object, specifying the user custom clip points.

## qualityIR, adapterIR, and customIR are allowed to by empty

clipMode a character string specifying the clipping mode to use for the object, see availableClipModes for more information.

header a list object with sff header information, see readSffHeader for more information.

Slots

Slots header are inherited from SffHeader. Additional slots defined in this class are:

sread**:** Object of class "DNAStringSet" containing IUPAC-standard, variable-length DNA strings representing sequence reads.

qualityIR**:** Object of class "IRanges" contains the clip points associated with quality clip locations specified in the sff file read header. Clip locations are auto filled to those specified in the sff file.

adapterIR**:** Object of class "IRanges" contains the clip points associated with adapter clip locations specified in the sff file read header. Clip locations are auto filled to those specified in the sff file.

customIR**:** Object of class "IRanges" contains the clip points associated with custom clip locations as specified by the user. Initial object is set to empty.

clipMode**:** Object of class "character" contains the current active clipMode, one of "raw", "adapter", "quality", "custom", "full". See explaination below for an expanded desription of clip modes.

Extends

Class "SffHeader", directly.

Methods

[signature(x = "SffReads", i = "ANY", j = "missing"): This method creates a new SffReads object containing only those reads indexed by i. Additional methods on '[,SffReads' do not provide additional functionality, but are present to limit appropriate use.

adapterClip signature(object = "SffReads"): Returns an IRanges object of clip points when clipMode is set to "adapter"

adapterClip<- signature(object = "SffReads"): Provide new adapter clip points with an IRanges object.

clipMode signature(object = "SffReads"): returns the current active clipMode being used.

clipMode<- signature(object = "SffReads"): set the current clip mode to one of "raw", "adapter", "quality", "full", or "custom"

customClip signature(object = "SffReads"): Returns an IRanges object of clip points when clipMode is set to "custom"

customClip<- signature(object = "SffReads"):Provide new custom clip points with an IRanges object.

fullClip signature(object = "SffReads"): Returns an IRanges object of clip points when clipMode is set to "full"

length signature(x = "SffReads"): returns a integer(1) vector describing the number of reads in this object.

width signature(x = "SffReads"): returns an integer() vector of the widths of each read in this object.

names signature(x = "SffReads"): access the id slot of the object returning a character vector of read names.

names<- signature(x = "SffReads", value = "ANY"): Provide new read ids with a character() vector of same length as the object.

qualityClip signature(object = "SffReads"): Returns an IRanges object of clip points when clipMode is set to "quality".

qualityClip<- `signature(object = "SffReads")`: Provide new quality clip points with an IRanges object.

rawClip `signature(object = "SffReads")`: Returns an IRanges object of clip points when clipMode is set to "full".

sread `signature(object = "ANY")`: access the sread slot of `object`, returning a DNAStringSet object of reads.

id `signature(object = "SffReads")`: this function is included to maintain compatibilty with the `ShortRead` class, returning a BStringSet object of read names for the `object`.

append `signature(x = "SffReads", values = "SffReads")`: append the `sread` and `id` slots of `values` after the corresponding fields of `x`.

writeFasta `signature(object="SffReads", file, ...)`: write `object` to `file` in fasta format. See `writeFasta` for `...` argument values.

Author(s)

Matt Settles with lots of code 'borrowed' from Martin Morgan's ShortRead Package

See Also

SffReadsQ and availableClipModes

Examples

```
showClass("SffReads")
showMethods(class="SffReads", where=getNamespace("rSFFreader"),
            inherit=FALSE)

sff <- readSff(system.file("extdata","Small454Test.sff",package="rSFFreader"), use.qualities=FALSE)
detail(sff)
```

SffReadsQ-class *Class "SffReadsQ" for next generation reads stored in SFF files (Roche 454 and Life Sciences Ion Torrent) and their quality scores*

Description

SffReadsQ class is a container for storing, next-generation sequencing read data, read id, and sequencing quality information imported directly from SFF files generated by technologies such as Roche 454 and Life Sciences Ion Torrent. This class provides a way to store and manipulate, in a coordinated fashion, next generation reads stored in SFF files, their identifiers and quality scores.

Objects from this class are the result of `readSff`, or can be constructed from `DNAStringSet`, `QualityScore`, and `BStringSet` objects, using a call to the constructor `SffReadsQ`, as described below.

Usage

```
## Constructor:

SffReadsQ(sread, quality, qualityIR, adapterIR, customIR, clipMode="raw", header)
```

Arguments

`sread`	an object of class "`DNAStringSet`" representing the sequence data.
`quality`	an object of class "`BStringSet`", or "`FastqQuality`" representing the quality values associated with the sequence in sread.
`qualityIR`	an object of type "`IRanges`" object, specifying the quality clip points.
`adapterIR`	an object of type "`IRanges`" object, specifying the adapter clip points.
`customIR`	an object of type "`IRanges`" object, specifying the user custom clip points. ## qualityIR, adapterIR, and customIR are allowed to by empty
`clipMode`	a character string specifying the clipping mode to use for the object, see availableClipModes for more information.
`header`	a list object with sff header information, see `readSffHeader` for more information.

Slots

Slot `header` is inherited from `SffHeader`. Slots `sread`, `qualityIR`, `adapterIR`, `customIR` and `clipMode` are inherited from `SffReads`. An additional slot defined in this class is:

`quality`**:** Object of class `"BStringSet"` representing a quality score (see `readFastq` for some discussion of quality score).

Extends

Class "`SffReads`", directly. Class "`SffHeader`", by class "SffReads", distance 2.

Methods

quality inherited from `signature(object = "ANY")`: access the quality slot of `object`.

[`signature(x = "SffReads", i = "ANY", j = "missing")`: This method creates a new `SffReads` object containing only those reads indexed by `i`. Additional methods on '[,SffReads' do not provide additional functionality, but are present to limit appropriate use.

writeFastaQual `signature(object = "SffReadsQ")`: ...

writeFastq `signature(object = "SffReadsQ", file = "character", mode="character", ...)`: Write `object` to `file` in fastq format. `mode` defaults to 'w'. This creates a new file, or fails if `file` already exists. Use `mode="a"` to append to an existing file. `file` is expanded using `path.expand`.

writePhredQual `signature(object = "FastqQuality", filepath, mode="w")`

writePhredQual `signature(object = "SffReadsQ", filepath, mode="w")`: Write `object`'s quality values to `filepath` in phred qual format (numeric). append defaults to 'w'. This creates a new file, or fails if `filepath` already exists. Use `mode="a"` to append to an existing file. `filepath` is expanded using `path.expand`.

writeFastaQual `signature(object = "SffReadsQ", basefilename = "character", append="logical", ...)`: Write `object` to `basefilename` in fasta and phred qual format. append defaults to 'FALSE'. This creates a new file, or fails if `basefilename` already exists. Use `append=TRUE` to append to an existing file. `basefilename` is expanded using `path.expand` and the suffixes [basefilename].fasta (sequence) and [basefilename].fasta.qual (qualities) are added.

append `signature(x = "SffReadsQ", values = "SffReads")`: append the `sread`, `quality` and `id` slots of `values` after the corresponding fields of `x`.

Author(s)

Matt Settles with lots of code ’borrowed’ from Martin Morgan’s ShortRead Package

See Also

readSff for creation of objects of this class from SFF files.

Examples

```
showClass("SffReadsQ")
showMethods(class="SffReadsQ", where=getNamespace("rSFFreader"),
            inherit=FALSE)
showMethods(class="SffReads", where=getNamespace("rSFFreader"),
            inherit=FALSE)

sff <- readSff(system.file("extdata","Small454Test.sff",package="rSFFreader"))
quality(sff)
```

Index