Package 'qrqc'

April 23, 2016

Version 1.24.0

Date 2012-04-17

Title Quick Read Quality Control

Author Vince Buffalo

Maintainer Vince Buffalo <vsbuffalo@ucdavis.edu>

Imports reshape, ggplot2, Biostrings, biovizBase, graphics, methods, plyr, stats

Depends reshape, ggplot2, Biostrings, biovizBase, brew, xtable, Rsamtools (>= 1.19.38), testthat

LinkingTo Rsamtools

Description Quickly scans reads and gathers statistics on base and quality frequencies, read length, k-mers by position, and frequent sequences. Produces graphical output of statistics for use in quality control pipelines, and an optional HTML quality report. S4 SequenceSummary objects allow specific tests and functionality to be written around the data collected.

License GPL (>=2)

URL http://github.com/vsbuffalo/qrqc

biocViews Sequencing, QualityControl, DataImport, Preprocessing, Visualization

NeedsCompilation yes

R topics documented:

`basePlot-methods` *Plot Base Frequency or Proportion by Position*

Description

`basePlot` plots the frequency or proportion of bases by position in the read.

Specific bases (such as "N") can be plot alone with this function too.

Usage

```
basePlot(x, geom=c("line", "bar", "dodge"),
  type=c("frequency", "proportion"), bases=DNA_BASES_N,
  colorvalues=getBioColor("DNA_BASES_N"))
```

Arguments

`x`	an S4 object that inherits from `SequenceSummary` from `readSeqFile`.
`geom`	Either "line", "bar", or "dodge" indicating the geom to use when plotting the bases. "line" will plot base proportion of frequency with lines. "bar" and "dodge" will use bars; "bar" defaults to filling the bars with different colors to distinguish bases, "dodge" lays the bars side by side.
`type`	Either "frequency" or "proportion" indicating whether to use count data or the proportion per base.

bases — a character vector indicating which bases to include. By default, all bases in DNA_BASES_N. Another good option would be IUPAC_CODE_MAP, which is included in the Biostrings package.

colorvalues — a character vectors of colors to use; the names of the elements must map to the bases.

Methods

signature(x = "FASTQSummary") basePlot will plot the base frequencies or proportions for a single object that inherits from SequenceSummary.

signature(x = "list") basePlot will plot the base frequencies or proportions for each of the SequenceSummary items in the list and display them in a series of panels.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getBase, getBaseProp

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot bases
basePlot(s.fastq)

## Plot bases with filled bars
basePlot(s.fastq, geom="bar")

## Plot bases with dodged bars
basePlot(s.fastq, geom="dodge")

## Plot bases with dodged bars
basePlot(s.fastq, geom="bar", bases=c("G", "T"))

## Plot multiple base plots
s.trimmed.fastq <- readSeqFile(system.file('extdata',
  'test-trimmed.fastq', package='qrqc'))
basePlot(list("not trimmed"=s.fastq, "trimmed"=s.trimmed.fastq))


## Graphical features can be added
basePlot(s.trimmed.fastq, type="proportion") +
  geom_hline(yintercept=0.25, color="purple")
```

`calcKL-methods` *Calculate the Kullback-Leibler Divergence Between the k-mer Distribution by Position and the k-mer Distribution Across All Positions.*

Description

`calcKL` takes in an object that inherits from `SequenceSummary` that has a kmers slot, and returns the terms of the K-L divergence sum (which correspond to items in the sample space, in this case, k-mers).

Usage

```
calcKL(x)
```

Arguments

`x`	an S4 object a class that inherits from `SequenceSummary`.

Value

`calcKL` returns a `data.frame` with columns:

`kmer`	the k-mer sequence.
`position`	the position in the read.
`kl`	the K-L term for this k-mer in the K-L sum, calculated as p(i)*log2(p(i)/q(i)).
`p`	the probability for this k-mer, at this position.
`q`	the probability for this k-mer across all positions.

Note

The K-L divergence calculation in `calcKL` uses base 2 in the log; the units are in bits.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`kmerKLPlot`, `getKmer`

Examples

```
## Load a somewhat contaminated FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'), hash.prop=1)

## As with getQual, this function is provided so custom graphics can
## be made easily. For example K-L divergence by position:
kld <- with(calcKL(s.fastq), aggregate(kl, list(position),
  sum))
colnames(kld) <- c("position", "KL")
p <- ggplot(kld) + geom_line(aes(x=position, y=KL), color="blue")
p + scale_y_continuous("K-L divergence")
```

FASTASummary-class FASTASummary *class representing the summaries of a FASTA file*

Description

This class contains the same slots as the SequenceSummary, but it is used to indicate the data originated from a FASTA file.

Note that many accessor functions transform data in the slots into data frames. The data in the slots is mostly untransformed and less easy to work with directly, so using the accessor functions is recommended.

Slots

FASTASummary has the slots inherited from SequenceSummary.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

FASTQSummary is the counterpart of this class for FASTQ data.

readSeqFile is the function that takes a FASTA file and returns a FASTASummary object.

basePlot is a function that plots the distribution of bases over sequence length for a particular FASTASummary object. gcPlot combines and plots the GC proportion.

seqlenPlot is a function that plots a histogram of sequence lengths for a particular FASTASummary object.

kmerKLPlot is a function that uses Kullback-Leibler divergence to make a plot that can aid in finding possible contamination (if readSeqFile had kmer=TRUE).

kmerEntropyPlot is a function that plots the Shannon entropy of k-mers per position.

There are acccessor functions getQual, getBase, getBaseProp, getSeqlen, getKmer, getGC for transforming the raw data in the object's slot (direct from the C call) to more usable data frames.

Examples

```
showClass("FASTASummary")
```

FASTQSummary-class FASTQSummary *class representing the summaries of a FASTQ file*

Description

This class contains the same slots as the SequenceSummary, as well as additional slots for quality information.

Note that many accessor functions transform data in the slots into data frames. The data in the slots is mostly untransformed and less easy to work with directly, so using the accessor functions is recommended.

Slots

In addition to the slots inherited from SequenceSummary, FASTQSummary contains:

`quality` a string indicating the type of quality (used to convert ASCII characters to quality integers). Either "phred", "solexa", or "illumina".

`qual.freqs` a data frame of quality frequencies by position, if the file was a FASTQ file.

`mean.qual` a numeric that is the mean quality across all positions, weighted by the number of reads that extended to that position.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

FASTASummary is the counterpart of this class for FASTA data.

readSeqFile is the function that takes a FASTQ file and returns a FASTQSummary object.

basePlot is a function that plots the distribution of bases over sequence length for a particular FASTQSummary object. gcPlot combines and plots the GC proportion.

qualPlot is a function that plots the distribution of qualities over sequence length for a particular FASTQSummary object.

seqlenPlot is a function that plots a histogram of sequence lengths for a particular FASTQSummary object.

kmerKLPlot is a function that uses Kullback-Leibler divergence to make a plot that can aid in finding possible contamination (if readSeqFile had kmer=TRUE).

kmerEntropyPlot is a function that plots the Shannon entropy of k-mers per position.

There are acccessor functions getQual, getBase, getBaseProp, getSeqlen, getKmer, getGC for transforming the raw data in the object's slot (direct from the C call) to more usable data frames.

Examples

```
showClass("FASTQSummary")
```

---

gcPlot-methods *Plot GC Content by Position*

Description

gcPlot plots the GC content by position in the read.

Usage

```
gcPlot(x, color="red")
```

Arguments

x	an S4 object that inherits from SequenceSummary from readSeqFile.
color	the color to use for the GC content line.

Methods

signature(x = "FASTQSummary") gcPlot will plot the GC content for a single object that inherits from SequenceSummary.

signature(x = "list") gcPlot will plot the GC content for each of the SequenceSummary items in the list and display them in a series of panels.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getBase, getBaseProp

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot GC content
gcPlot(s.fastq)

## Plot multiple GC content plots
s.trimmed.fastq <- readSeqFile(system.file('extdata',
  'test-trimmed.fastq', package='qrqc'))
gcPlot(list("not trimmed"=s.fastq, "trimmed"=s.trimmed.fastq))

## Graphical features can be added
gcPlot(s.trimmed.fastq) + geom_hline(yintercept=0.5, color="purple")
```

geom_qlinerange *Use Line Segments and Points to Plot Quality Statistics by Position in the Read*

Description

geom_qlinerange uses multiple line segments and points to plot quality ranges. By default the 10% and 90% range in plotted in grey, the quartile range in orange, and the mean as a point in blue. It is used in qualPlot.

Usage

```
geom_qlinerange(extreme.color="grey", quartile.color="orange", mean.color="blue", median.color=NULL)
```

Arguments

extreme.color	a character value indicating the color to use for the extreme values (the 10% and 90% quantiles). If NULL, these line seqments will not be added.
quartile.color	a character value indicating the color to use for the quartiles. If NULL, these line seqments will not be added.
mean.color	a character value indicating the color to use for the mean. If NULL, these line seqments will not be added.
median.color	a character value indicating the color to use for the median. If NULL, these line seqments will not be added.

Value

A list of geoms from ggplot2 that this function put together, to be added to a call to ggplot which contains a data frame of quality data, i.e. from getQual.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getQual, qualPlot

getBase-methods *Get a Data Frame of Base Frequency Data from a* SequenceSummary *Object*

Description

An object that inherits from class SequenceSummary contains base frequency data by position gathered by readSeqFile. getBase is an accessor function that reshapes the base frequency data by position into a data frame.

This accessor function is useful if you want to map variables to custom ggplot2 aesthetics. Base proportions can be accessed with getBaseProp.

Usage

```
getBase(x, drop=TRUE)
```

Arguments

x	an S4 object that inherits from SequenceSummary from readSeqFile.
drop	a logical value indicating whether to drop bases that don't have any counts.

Value

getBase returns a data.frame with columns:

position	the position in the read.
base	the base.
frequency	the number of a base found per position in the read.

Methods

signature(x = "SequenceSummary") getBase is an accessor function that works on any object read in with readSeqFile; that is, objects that inherit from SequenceSummary.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getGC, getSeqlen, getBaseProp, getQual, getMCQual, basePlot

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'))

# A custom base plot
ggplot(getBase(s.fastq)) + geom_line(aes(x=position, y=frequency,
  color=base)) + facet_grid(. ~ base) + scale_color_dna()
```

`getBaseProp-methods` *Get a Data Frame of Base Proportion Data from a* `SequenceSummary` *object*

Description

An object that inherits from class `SequenceSummary` contains base frequency data by position gathered by `readSeqFile`. `getBaseProp` is an accessor function that reshapes the base frequency data by position into a data frame and calculates the proportions of each base per position.

This accessor function is useful if you want to map variables to custom `ggplot2` aesthetics. Base frequency be accessed with `getBase`.

Usage

```
getBaseProp(x, drop=TRUE)
```

Arguments

`x`	an S4 object that inherits from `SequenceSummary` from `readSeqFile`.
`drop`	a logical value indicating whether to drop bases that don't have any counts.

Value

`getBaseProp` returns a `data.frame` with columns:

`position`	the position in the read.
`base`	the base.
`proportion`	the proportion of a base found per position in the read.

Methods

`signature(x = "SequenceSummary")` `getBaseProp` is an accessor function that works on any object read in with `readSeqFile`; that is, objects that inherit from `SequenceSummary`.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getGC, getSeqlen, getBase, getQual, getMCQual, basePlot

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'))

# A custom base plot
ggplot(getBaseProp(s.fastq)) + geom_line(aes(x=position, y=proportion,
  color=base)) + facet_grid(. ~ base) + scale_color_dna()
```

---

getGC-methods *Get a Data Frame of GC Content from a* SequenceSummary *object*

Description

An object that inherits from class SequenceSummary contains base frequency data by position gathered by readSeqFile. getGC is an accessor function that reshapes the base frequency data into a data frame and returns the GC content by position.

This accessor function is useful if you want to map variables to custom ggplot2 aesthetics. Frequencies or proportions of all bases (not just GC) can be accessed with getBase and getBaseProp respectively.

Usage

```
getGC(x)
```

Arguments

x	an S4 object that inherits from SequenceSummary from readSeqFile.

Value

getGC returns a data.frame with columns:

position	the position in the read.
gc	GC content per position in the read.

Methods

signature(x = "SequenceSummary") getGC is an accessor function that works on any object read in with readSeqFile; that is, objects that inherit from SequenceSummary.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getSeqlen, getBase, getBaseProp, getQual, getMCQual, getKmer, gcPlot

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'))

# A custom GC plot
d <- merge(getQual(s.fastq), getGC(s.fastq), by.x="position", by.y="position")
p <- ggplot(d) + geom_linerange(aes(x=position, ymin=lower,
  ymax=upper, color=gc)) + scale_color_gradient(low="red",
  high="blue") + scale_y_continuous("GC content")
p
```

---

getKmer-methods — *Get a Data Frame of k-mer Frequency by Position from a* SequenceSummary *Object*

Description

An object that inherits from class SequenceSummary contains k-mer frequency data by position gathered by readSeqFile when kmer=TRUE. getKmer is an accessor function that is useful for custom ggplot2 aesthetics.

Usage

```
getKmer(x)
```

Arguments

x	an S4 object that inherits from class SequenceSummary from, as returned from readSeqFile.

Value

getKmer returns a data.frame with columns:

kmer	the k-mer sequence.
position	the position in the read.
count	the frequency of the k-mer at this position.

Methods

signature(x="SequenceSummary") getKmer is an accessor function that only works if there is k-mer data, thus it only works if readSeqFile was called with kmer=TRUE (and hash.prop is greater than 0).

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getGC, getSeqlen, getBase, getBaseProp, getQual, getMCQual, kmerKLPlot, kmerEntropyPlot

Examples

```
## Load a FASTQ file, with sequence and k-mer hashing on by default.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## plot counts of a subset of k-mers by position
s.kmers <- getKmer(s.fastq)
top.kmers <- s.kmers$kmer[order(s.kmers$count, decreasing=TRUE)[1:40]]
p <- ggplot(subset(s.kmers, kmer %in% top.kmers)) + geom_bar(aes(x=position, y=count,
  fill=kmer), stat="identity")
p
```

---

getMCQual-methods *Get a Data Frame of Simulated Qualitied from a* FASTQSummary *object*

Description

An object that inherits from class FASTQSummary contains base quality data by position gathered by readSeqFile. getMCQual generates simulated quality data for each base from this binned quality data that can be used for adding smoothed lines via lowess.

This accessor function is useful if you want to map variables to custom ggplot2 aesthetics.

Usage

```
getMCQual(x, n=100)
```

Arguments

x	an S4 object that inherits from FASTQSummary from readSeqFile.
n	a numeric value indicating the number of quality values to draw per base.

Value

getMCQual returns a data.frame with columns:

position	the position in the read.
quality	simulated quality scores.

Methods

`signature(x = "FASTQSummary")` `getMCQual` is a function that works on any object with class `FASTQSummary` read in with `readSeqFile`.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`getGC`, `getSeqlen`, `getBase`, `getBaseProp`, `getQual`, `qualPlot`

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'))

# A custom quality plot
ggplot(getQual(s.fastq)) + geom_linerange(aes(x=position, ymin=lower,
  ymax=upper), color="grey") + geom_smooth(aes(x=position, y=quality),
  data=getMCQual(s.fastq), color="blue", se=FALSE)
```

`getQual-methods` *Get a Data Frame of Quality Data from a* `FASTQSummary` *object*

Description

An object of class `FASTQSummary` contains quality data (binned by `readSeqFile`). `getQual` is an accessor function that reshapes the data into a data frame.

This accessor function is useful if you want to map variables to custom `ggplot2` aesthetics.

Usage

```
getQual(x)
```

Arguments

`x`	an S4 object of class `FASTQSummary` from `readSeqFile`.

Value

`getQual` returns a `data.frame` with columns:

`position`	the position in the read.
`ymin`	the minimum quality found per a position in the read.
`alt.lower`	the 10% quantile found per a position in the read.

lower	the 25% quartile found per a position in the read.
middle	the median found per a position in the read.
upper	the 75% quartile found per a position in the read.
alt.upper	the 90% quantile found per a position in the read.
ymax	the maximum quality found per a position in the read.
mean	the mean quality (calculated from the binned data by using a weighted mean function) per the position in the read.

Methods

`signature(x="FASTQSummary")` `getQual` is an accessor function that only works if there is quality data, thus it only works with objects of class `FASTQSummary`.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`getGC`, `getSeqlen`, `getBase`, `getBaseProp`, `getMCQual`, `qualPlot`

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Mean quality by position
p <- ggplot(getQual(s.fastq)) + geom_line(aes(x=position, y=mean), color="blue")
p <- p + scale_y_continuous(limits=c(0, 42))
p

## A different type of quality plot
p <- ggplot(getQual(s.fastq)) + geom_linerange(aes(x=position,
  ymin=lower, ymax=upper, color=mean))
p <- p +  scale_color_gradient("mean quality", low="red", high="green")
p + scale_y_continuous("quality")
```

`getSeqlen-methods` *Get a Data Frame of Sequence Lengths from a* `SequenceSummary` *object*

Description

An object that inherits from class `SequenceSummary` contains sequence length data by position gathered by `readSeqFile`. `getSeqlen` is an accessor function that returns the sequence length data.

This accessor function is useful if you want to map variables to custom `ggplot2` aesthetics.

Usage

```
getSeqlen(x)
```

Arguments

x	an S4 object that inherits from SequenceSummary from readSeqFile.

Value

getSeqlen returns a data.frame with columns:

length	the sequence length.
count	the number of reads with this sequence length.

Methods

signature(x = "SequenceSummary") getSeqlen is an accessor function that works on any object read in with readSeqFile; that is, objects that inherit from SequenceSummary.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getGC, getBase, getBaseProp, getQual, getMCQual, seqlenPlot

Examples

```
library(ggplot2)

## Load a FASTQ file, with sequence hashing.
s.trimmed.fastq <- readSeqFile(system.file('extdata', 'test-trimmed.fastq',
  package='qrqc'))

# A custom plot - a bit contrived, but should show power
d <- merge(getSeqlen(s.trimmed.fastq), getQual(s.trimmed.fastq),
  by.x="length", by.y="position")
ggplot(d) + geom_linerange(aes(x=length, ymin=0, ymax=count),
  color="grey") + geom_linerange(aes(x=length, ymin=lower, ymax=upper),
  color="blue") + scale_y_continuous("quality/count") + theme_bw()
```

kmerEntropyPlot-methods

Plot Entropy of k-mers by Position

Description

kmerEntropyPlot plots the Shannon entropy per position of k-mers. Lower Shannon entropy implies that the distribution of k-mers is non-random and could indicate bias.

Usage

```
kmerEntropyPlot(x)
```

Arguments

x	an S4 object a class that inherits from SequenceSummary from readSeqFile or a list of objects that inherit from SequenceSummary with names.

Methods

signature(x = "SequenceSummary") kmerEntropyPlot will plot Shannon entropy per position for an object that inherits from SequenceSummary.

signature(x = "list") kmerEntropyPlot will plot the Shannon entropy per position for each of the objects that inherit from SequenceSummary in the list and display them in a series of panels.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getKmer, calcKL, kmerKLPlot

Examples

```
## Load a somewhat contaminated FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'), hash.prop=1)

## Load a really contaminated FASTQ file
s.contam.fastq <- readSeqFile(system.file('extdata',
  'test-contam.fastq', package='qrqc'), hash.prop=1)

## Load a random (equal base frequency) FASTA file
s.random.fasta <- readSeqFile(system.file('extdata',
  'random.fasta', package='qrqc'), type="fasta", hash.prop=1)
```

```
## Plot the Shannon entropy for a single file
kmerEntropyPlot(s.fastq)

## Plot the Shannon entropy for many files
kmerEntropyPlot(list("highly contaminated"=s.contam.fastq, "less
  contaminated"=s.fastq, "random"=s.random.fasta))
```

`kmerKLPlot-methods` *Plot K-L Divergence Components for a Subset of k-mers to Inspect for Contamination*

Description

`kmerKLPlot` calls `calcKL`, which calculates the Kullback-Leibler divergence between the k-mer distribution at each position compared to the k-mer distribution across all positions. `kmerKLPlot` then plots each k-mer's contribution to the total K-L divergence by stack bars, for a *subset* of the k-mers. Since there are 4^k possible k-mers for some value k-mers, plotting each often dilutes the interpretation; however one can increase `n.kmers` to a number greater than the possible number of k-mers to force `kmerKLPlot` to plot the entire K-L divergence and all terms (which are k-mers) in the sum.

If a `x` is a `list`, the K-L k-mer plots are faceted by sample; this allows comparison to a FASTA file of random reads.

Again, please note that this is *not* the total K-L divergence, but rather the K-L divergence calculated on a subset of the sample space (those of the top `n.kmers` k-mers selected).

Usage

```
kmerKLPlot(x, n.kmers=20)
```

Arguments

`x`	an S4 object a class that inherits from `SequenceSummary` from `readSeqFile` or a list of objects that inherit from `SequenceSummary` with names.
`n.kmers`	a integer value indicating the size of top k-mers to include.

Methods

`signature(x = "SequenceSummary")` `kmerKLPlot` will plot the K-L divergence for a subset of k-mers for a single object that inherits from `SequenceSummary`.

`signature(x = "list")` `kmerKLPlot` will plot the K-L divergence for a susbet of k-mers for each of the objects that inherit from `SequenceSummary` in the list and display them in a series of panels.

Note

The K-L divergence calculation in `calcKL` uses base 2 in the log; the units are in bits.

Also, note that `ggplot2` warns that "Stacking is not well defined when ymin != 0". This occurs when some k-mers are less frequent in the positional distribution than the distribution across all positions, and the term of the K-L sum is negative (producing a bar below zero). This does not appear to affect the plot much. In examples below, warnings are suppressed, but the given this is a valid concern from `ggplot2`, warnings are not suppressed in the function itself.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`getKmer`, `calcKL`, `kmerEntropyPlot`

Examples

```
## Load a somewhat contaminated FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq',
  package='qrqc'), hash.prop=1)

## Load a really contaminated FASTQ file
s.contam.fastq <- readSeqFile(system.file('extdata',
  'test-contam.fastq', package='qrqc'), hash.prop=1)

## Load a random (equal base frequency) FASTA file
s.random.fasta <- readSeqFile(system.file('extdata',
  'random.fasta', package='qrqc'), type="fasta", hash.prop=1)

## Make K-L divergence plot - shows slight 5'-end bias. Note units
## (bits)
suppressWarnings(kmerKLPlot(s.fastq))

## Plot multiple K-L divergence plots
suppressWarnings(kmerKLPlot(list("highly contaminated"=s.contam.fastq, "less
  contaminated"=s.fastq, "random"=s.random.fasta)))
```

---

list2df *Apply a function to items in list and combine into data frame*

Description

`list2df` is a helper function that takes a named list and applies a function to each element, and combines the resulting data frames into a single data frame. The output data frame will have an additional column named `sample` indicating which element the data came from.

Usage

```
list2df(x, fun)
```

Arguments

x	a named list of objects.
fun	a function that takes in the elements of x and outputs a data frame.

Value

A data frame made by applying fun to each element of the list x. An additional column named sample will indicate which element the data came from.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

Examples

```
## Get some sequence files
sq.files = list.files(system.file('extdata', package='qrqc'),
  pattern="test.*fastq", full.names=TRUE)
names(sq.files) <- gsub("(.*)\\.fastq", "\1", basename(sq.files))
sq <- lapply(sq.files, readSeqFile)


## Take the FASTQSummary objects, extract quality data from each of
## the, and combine.
d <- list2df(sq, getQual)

## Look at difference in average quality
aggregate(d$mean, list(sample=d$sample), mean)

## Look at difference in variance - this is where we really see a
## change.
aggregate(d$mean, list(sample=d$sample), var)
```

---

makeReport-methods *Make an HTML report from a FASTASummary of FASTQSummary object*

Description

makeReport takes a FASTQSummary or FASTASummary object, creates an HTML report, and writes it to a file within a directory. The directory naming is incremental so past reports will not be overwritten.

Usage

```
makeReport(x, outputDir=".")
```

Arguments

x	an object that is either FASTQSummary or FASTASummary.
outputDir	an optional character argument to to indicate the report output directory. By default, the current directory.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

Examples

```
## Load a FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Make and save a report
makeReport(s.fastq)
```

---

plotBases-methods *Plot Bases by Position*

Description

plotBases plots the frequency or proportion of bases by position in the read.

plotBases uses the Sanger base color scheme: blue is Cytosine, green is Adenine, black is Guanine, red is Thymine, and purple in N (any base). Other IUPAC nucleotides are colored using **RColorBrewer**.

Usage

```
plotBases(obj, type="freq", bases=NULL, legend=TRUE)
```

Arguments

obj	an S4 object of class that inherits from SequenceSummary (either FASTASummary or FASTQSummary) from readSeqFile.
type	a character string that is either "freq" or "prop" indicating whether to plot frequencies or proportions on the y-axis.
bases	a vector of characters indicating which bases to include. The default value NULL indicates to plot _all_ bases.
legend	a logical value indicating whether to include a legend on the top right.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

basePlot

Examples

```
## Not run:
  ## Load a FASTQ file, with sequence hashing.
  s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

  ## Plot base frequencies
  plotBases(s.fastq, type="freq")

  ## Plot base proportions
  plotBases(s.fastq, type="prop")

## End(Not run)
```

---

plotGC-methods *Plot per Base GC Content by Position*

Description

plotGC plots the GC proportion by position.

Usage

```
plotGC(obj)
```

Arguments

obj	an S4 object of class that inherits from SequenceSummary (either FASTASummary or FASTQSummary) from readSeqFile.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

gcPlot

Examples

```
## Not run:
  ## Load a FASTQ file, with sequence hashing.
  s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

  ## Plot Qualities
  plotGC(s.fastq)

## End(Not run)
```

---

plotQuals-methods *Plot a Base Quality Boxplot by Position*

Description

plotQuals plots quality statistics by position. Optionally, it adds a lowess curve through the qualities, which is fit with data randomly drawn from the distribution of qualities at each position. A histogram of the sequence length distribution is plotted above the quality plot when histogram is TRUE.

A legend is plotted on the bottom left if legend is TRUE (this location is used because this where the bases are likely to be of highest quality, and thus not overlap the legend). The grey lines indicate the range of the 10% and 90% quantiles, the orange lines indicate the range of the 25% and 75% quartiles, the blue point is the median, the green dash is the mean, and the purple line is the lowess curve if lowess is TRUE.

Usage

```
plotQuals(obj, ylim='relative', lowess=TRUE, histogram=TRUE, legend=TRUE)
```

Arguments

obj	an S4 object of class FASTQSummary from readSeqFile.
ylim	either 'relative' or 'fixed', which will scale the y axis to either the relative range (from the data) or absolute range of qualities.
lowess	a logical value indicating whether to fit a lowess curve through the quality plot.
histogram	a logical value indicating whether to add a histogram of the sequence length distribution above the quality plot.
legend	a logical value indicating whether a legend is to be included.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

qualPlot

Examples

```
## Not run:
  ## Load a FASTQ file, with sequence hashing.
  s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

  ## Plot Qualities
  plotQuals(s.fastq)

## End(Not run)
```

---

plotSeqLengths-methods

Plot Histogram of Sequence Lengths

Description

plotSeqLengths plots histogram of sequence lengths.

Usage

```
plotSeqLengths(obj)
```

Arguments

obj	an S4 object of class that inherits from SequenceSummary (either FASTASummary or FASTQSummary) from readSeqFile.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

seqlenPlot

Examples

```
## Not run:
  ## Load a FASTQ file, with sequence hashing.
  s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

  ## Plot Qualities
  plotSeqLengths(s.fastq)

## End(Not run)
```

qualPlot-methods *Plot a Base Quality Boxplot by Position*

Description

basePlot plots quality statistics by position. Optionally, it adds a smoothed curve through the qualities, which is fit with data randomly drawn from the distribution of qualities at each position.

The grey lines indicate the range of the 10% and 90% quantiles, the orange lines indicate the range of the 25% and 75% quartiles, the blue point is the mean. Optionally, one can plot the median as well.

Usage

```
qualPlot(x, smooth=TRUE, extreme.color="grey", quartile.color="orange", mean.color="blue", median.c
```

Arguments

x	an S4 object of class FASTQSummary from readSeqFile or a list of FASTQSummary objects with names.
smooth	a logical value indicating whether to add a smooth curve.
extreme.color	a character value indicating the color to use for the extreme values (the 10% and 90% quantiles). If NULL, these line seqments will not be added.
quartile.color	a character value indicating the color to use for the quartiles. If NULL, these line seqments will not be added.
mean.color	a character value indicating the color to use for the mean. If NULL, these line seqments will not be added.
median.color	a character value indicating the color to use for the median. If NULL, these line seqments will not be added.

Methods

signature(x = "FASTQSummary") qualPlot will plot the qualities for a single object of class FASTQSummary.

signature(x = "list") qualPlot will plot the qualities for each of the FASTQSummary items in the list and display them in a series of panels.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

getQual

Examples

```
## Load a FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot qualities
qualPlot(s.fastq)

## Combine with ggplot2 to produce custom graphics
p <- qualPlot(s.fastq, smooth=FALSE)
p <- p + geom_smooth(aes(x=position, y=quality),
  data=getMCQual(s.fastq), method="lm", color="green", se=FALSE)
p <- p + theme_bw()
p

## Plot multiple quality plots
s.trimmed.fastq <- readSeqFile(system.file('extdata',
  'test-trimmed.fastq', package='qrqc'))
qualPlot(list("not trimmed"=s.fastq, "trimmed"=s.trimmed.fastq))
```

`readSeqFile` *Read and Summarize a Sequence (FASTA or FASTQ) File*

Description

`readSeqFile` reads a FASTQ or FASTA file, summarizing the nucleotide distribution across position (cycles) and the sequence length distributions. If `type` is 'fastq', the distribution of qualities across position will also be recorded. If `hash` is `TRUE`, the unique sequences will be hashed with counts of their frequency. By default, only 10% of the reads will be hashed; this proportion can be controlled with `hash.prop`. If `kmer=TRUE`, k-mers of length `k` will be hashed by position, also with the sampling proportion controlled by `hash.prop`.

Usage

```
readSeqFile(filename, type=c("fastq", "fasta"), max.length=1000,
            quality=c("sanger", "solexa", "illumina"), hash=TRUE,
            hash.prop=0.1, kmer=TRUE, k=6L, verbose=FALSE)
```

Arguments

`filename`	the name of the file which the sequences are to be read from.
`type`	either 'fastq' or 'fasta', representing the type of the file. FASTQ files will have the quality distribution by position summarized.
`max.length`	the largest sequence length likely to be encountered. For efficiency, a matrix larger than the largest sequence is allocated to *this* size in C, populated, and then trimmed in R. Specifying a value too small will lead to an error and the function will need to be re-run.

quality	either 'illumina', 'sanger', or 'solexa', this determines the quality offsets and range. See the values of QUALITY.CONSTANTS for more information.
hash	a logical value indicating whether to hash sequences
hash.prop	a numeric value in (0, 1] that functions as the proportion of reads to hash.
kmer	a logical value indicating whether to hash k-mers by position.
k	an integer value indicating the k-mer size.
verbose	a logical value indicating whether be verbose (in the C backend).

Value

An S4 object of FASTQSummary or FASTASummary containing the summary statistics.

Note

Identifying the correct quality can be difficult. readSeqFile will error out if it a base quality outside of the range of a known quality type, but it is possible one could have reads with a different quality type that won't fall outside of the another type.

Here is a bit more about quality:

phred PHRED quality scores (e.g. from Roche 454). ASCII with no offset, range: [4, 60]. This has been removed as an option since sequence reads with this type are very, very uncommon.

sanger Sanger are PHRED ASCII qualities with an offset of 33, range: [0, 93]. From NCBI SRA, or Illumina pipeline 1.8+.

solexa Solexa (also very early Illumina - pipeline < 1.3). ASCII offset of 64, range: [-5, 62]. Uses a different quality-to-probabilities conversion than other schemes.

illumina Illumina output from pipeline versions between 1.3 and 1.7. ASCII offset of 64, range: [0, 62].

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

FASTQSummary and FASTASummary are the classes of the objects returned by readSeqFile.

basePlot is a function that plots the distribution of bases over sequence length for a particular FASTASummary or FASTQSummary object. gcPlot combines and plots the GC proportion.

qualPlot is a function that plots the distribution of qualities over sequence length for a particular FASTASummary or FASTQSummary object.

seqlenPlot is a function that plots a histogram of sequence lengths for a particular FASTASummary or FASTQSummary object.

kmerKLPlot is a function that plots K-L divergence of k-mers to look for possible biase in reads.

Examples

```
## Load a FASTQ file, with sequence hashing.
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Load a FASTA file, without sequence hashing.
s.fasta <- readSeqFile(system.file('extdata', 'test.fasta', package='qrqc'),
                       type='fasta', hash=FALSE)
```

---

scale_color_dna *Set the color scheme to* biovizBase*'s for DNA*

Description

This wraps ggplot2's scale_color_manual to use biovizBase's scheme for DNA (with N).

Usage

```
scale_color_dna()
```

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

scale_color_iupac, basePlot

Examples

```
## Load a FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot qualities with the DNA color scheme
ggplot(getBase(s.fastq)) + geom_line(aes(x=position, y=frequency,
  color=base)) + scale_color_dna()
```

scale_color_iupac *Set the color scheme to* biovizBase*'s for IUPAC codes*

Description

This wraps ggplot2's scale_color_manual to use biovizBase's scheme IUPAC nucleotides codes.

Usage

```
scale_color_iupac()
```

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

scale_color_dna, basePlot

Examples

```
## Load a FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot qualities with the DNA color scheme
ggplot(getBase(s.fastq)) + geom_line(aes(x=position, y=frequency,
  color=base)) + scale_color_iupac()
```

seqlenPlot-methods *Plot a Histogram of Sequence Lengths*

Description

seqlenPlot plots a histogram of sequence lengths.

Usage

```
seqlenPlot(x)
```

Arguments

x	an S4 object that inherits from SequenceSummary from readSeqFile.

Methods

`signature(x = "FASTQSummary")` `seqlenPlot` will plot a histogram of a single object that inherits from `SequenceSummary`.

`signature(x = "list")` `seqlenPlot` will plot a histogram for each of the `SequenceSummary` items in the list and display them in a series of panels.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`getSeqlen`

Examples

```
## Load a FASTQ file
s.fastq <- readSeqFile(system.file('extdata', 'test.fastq', package='qrqc'))

## Plot sequence lengths
seqlenPlot(s.fastq)

## Plot sequence legnths before and after trimming
s.trimmed.fastq <- readSeqFile(system.file('extdata',
  'test-trimmed.fastq', package='qrqc'))
seqlenPlot(list("not trimmed"=s.fastq, "trimmed"=s.trimmed.fastq))
```

`SequenceSummary-class` `SequenceSummary` *class representing the summaries of a sequence file*

Description

A sequence file read in with `readSeqFile` is summarized by a C call. This is a base class with slots common to both `FASTQSummary` and `FASTASummary`. This is not usually instantiated directly.

Note that many accessor functions transform data in the slots into data frames. The data in the slots is mostly untransformed and less easy to work with directly, so using the accessor functions is recommended.

Slots

`filename` the filename processed by `readSeqFile`.

`base.freqs` a data frame of base frequencies by position. Each column is a nucleotide (there is a column for position too), and each row contains the count frequencies of bases for that position.

`seq.lengths` a numeric vector of the number of sequences of a particular length (the length is the position in the vector).

`hash` a numeric vector of the count frequencies of sequences (the sequences are in the name attribute).

`hash.prop` a numeric value indicating the proportion of sequences that were sampled for hashing.

`kmer` a data frame of k-mer frequency by position.

`k` an integer indicating the length of k-mers hashed.

`hashed` a logical indicating whether the sequences were hashed in `readSeqFile`.

`kmers.hashed` a logical indicating whether the k-mers were hashed in `readSeqFile`.

Author(s)

Vince Buffalo <vsbuffalo@ucdavis.edu>

See Also

`FASTQSummary` and `FASTASummary` are the classes that inherit from `SequenceSummary`.

`readSeqFile` is the function that takes a FASTQ or FASTA file and returns a `FASTQSummary` object or `FASTASummary` object.

Examples

```
showClass("SequenceSummary")
```

Index