Package 'seqTools'

October 14, 2021

Type Package

Title Analysis of nucleotide, sequence and quality content on fastq files

Version 1.26.0

Date 2017-04-13

Author Wolfgang Kaisers

Maintainer Wolfgang Kaisers <kaisers@med.uni-duesseldorf.de>

Description Analyze read length, phred scores and alphabet frequency and DNA k-mers on uncompressed and compressed fastq files.

biocViews QualityControl,Sequencing

License Artistic-2.0

Depends methods,utils,zlibbioc

LinkingTo zlibbioc

Suggests RUnit, BiocGenerics

NeedsCompilation yes

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

git_branch RELEASE_3_13

git_last_commit f74d126

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

R topics documented:

seqTools-package 2
ascii2char 3
cbDistMatrix 4
collectDur 5
countDnaKmers 6
countFastaKmers 7
countGenomeKmers 8

countSpliceKmers . . . 10
fastqKmerLocs . . . 11
fastqKmerSubsetLocs . . . 12
fastqq . . . 13
Fastqq-class . . . 14
gcContentMatrix . . . 17
kMerIndex . . . 18
meltDownK . . . 19
mergedPhred . . . 20
mergeFastqq . . . 21
phredDist . . . 22
phredTable . . . 23
plotGCcontent . . . 24
plotKmerCount . . . 25
plotNucCount . . . 26
plotNucFreq . . . 27
plotPhredQuant . . . 29
propPhred . . . 30
revCountDnaKmers . . . 31
simFastqqRunTimes . . . 32
sim_fq . . . 33
trimFastq . . . 34
writeFai . . . 35
writeSimContFastq . . . 36
writeSimFastq . . . 37

Index **39**

`seqTools-package` *SeqTools: Bioconductor package for analysis of FASTQ and fasta files.*

Description

Analyze read length, phred scores and alphabeth frequency and DNA k-mers on uncompressed and compressed files.

Details

Package:	seqTools
Type:	Package
Version:	0.99.31
Date:	2013-10-14
License:	GPL-2
Depends:	methods

Author(s)

Wolfgang Kaisers Maintainer: Wolfgang Kaisers <kaisers@med.uni-duesseldorf.de>

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
# A) Count DNA k-mer
countDnaKmers("ATAAATA", 2)
# B) Quality check on FASTQ file
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq <- fastqq("test_l6.fq")
plotPhredQuant(fq, 1)
```

`ascii2char` *ascii2char: Converting ASCII encoded values to character values.*

Description

`ascii2char` calculates character representations for given phred values. `char2ascii` returns phred values for given ASCII encoded representations (the reverse transformation of `ascii2char`).

Usage

```
ascii2char(x, multiple=FALSE)
char2ascii(c)
```

Arguments

`x`	`numeric`. Vector with ASCII values. All values must be in 1:255. Other values produce an error.
`multiple`	`logical`. For `'FALSE'` (the default), all characters are combined into one single string (i.e. a character vector of length 1). For `'TRUE'`, single characters are combined into a vector.
`c`	`character`. Vector of length 1 (Longer vectors will generate Warnings).

Details

The functions are only wrappers for convenience. `char2ascii` is defined as `strtoi(charToRaw(c),base = 16L)`. `ascii2char` is defined as `rawToChar(as.raw(x),multiple)`.

Value

ascii2char returns character. char2ascii returns integer.

Author(s)

Wolfgang Kaisers

References

Ewing B, Green P Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities Genome Research 1998 8(3): 186-194

See Also

getPhredTable

Examples

```
ascii2char(97:101, multiple=FALSE)
ascii2char(97:101, multiple=TRUE)
char2ascii("abcde")
char2ascii(paste("a", "b", "c", collapse=""))
ascii2char(char2ascii("abcde"))
```

---

cbDistMatrix *cbDistMatrix function: Calculates pairwise distance matrix from DNA k-mer counts based on a modified Canberra distance.*

Description

Calculates pairwise distance matrix from DNA k-mer counts based on a modified Canberra distance. Before calculating canberra distances, read counts are normalized (in order to correct systematic effects on the distance) by scaling up read counts in each DNA k-mer count vector so that normalized read counts in each sample are nearly equal.

Usage

```
cbDistMatrix(object,nReadNorm=max(nReads(object)))
```

Arguments

object	Fastqq: Object from which DNA k-mer counts are used.
nReadNorm	numeric: Number of reads per file to wich all contained DNA k-mer counts are normalized. Because the normalization is intended to increase counts the value must be greater than all FASTQ file read counts (as reported by nReads). Therefore the standard value is chosen to the maximal number of reads recorded in this object. This normalization is necessary to compensate for systematic effects in the canberra distance.

Details

The distance between two DNA k-mer normalized count vectors is calculated by

$$df(X,Y) = \sum_{i=1}^{n} cbd(x_i, y_i)/4^k$$

where cb is given by

$$cbd(x,y) = |x-y|/(x+y).$$

Value

Square `matrix`. The number of rows equals the number of files (`=nFiles(object)`).

Note

The static size of the retured k-mer array is 4^k.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

hclust

Examples

```
basedir<-system.file("extdata",package="seqTools")
basenames<-c("g4_l101_n100.fq.gz","g5_l101_n100.fq.gz")
filenames<-file.path(basedir,basenames)
fq<-fastqq(filenames,6,c("g4","g5"))
dm<-cbDistMatrix(fq)
```

`collectDur` *collectDur: Returning elapsed time (in seconds) for collection of data from FASTQ files.*

Description

Objects of class `Fastqq` are created by reading data from FASTQ-files using the function `fastqq`. The `fastqq` function calls `Sys.time()` before and after execution of the core collecting routine. `collectDur` returns the number of seconds between these two times (as numeric value). `collectTime` returns the two timestamps inside a `list`.

Usage

```
collectDur(object)
collectTime(object)
```

Arguments

object	Fastqq. Object from which collection duration (or times) is returned.

Value

collectTime returns numeric. collectTime returns list.

Author(s)

Wolfgang Kaisers

See Also

fastqq

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq<-fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
            probeLabel=c("g4", "g5"))

collectTime(fq)
collectDur(fq)
```

---

countDnaKmers *countDnaKmers: Counting k-mers in DNA sequence.*

Description

Counts occurrence of DNA k-mers in given DNA sequence. The k-mers are searched in a set of search windows, which are defined by start and width parameter. From each position of the search window, a DNA k-mer is identified on the right hand side on the given DNA sequence. Each value in the start vector defines the left border of a search window. The size of the search window is given by the appropriate value in the width vector. The function is intended to count DNA k-mers in selected regions (e.g. exons) on DNA sequence.

Usage

```
countDnaKmers(dna,k,start,width)
```

Arguments

dna	character. Single DNA sequence (vector of length 1). dna must not contain other characters than "ATCGN". Capitalization does not matter. When a 'N' character is found, the current DNA k-mer is skipped.
k	numeric. Number of nucleotides in tabled DNA motifs.
start	numeric. Vector of (1-based) start positions for reading frames. Reading frame is counted to the right side of the DNA string.
width	numeric. Defines size of search window for each start position. Must have the same length as start or length 1 (in which case the values of width are recycled.

Details

The start positions for counting of DNA k-mers are all positions in {start,...,start+width-1}. As the identification of a DNA k-mer scans a sequence window of size k, the last allowed start position counting a k-mer is nchar(dna)-k+1. The function throws the error 'Search region exceeds string end' when a value start + width + k > nchar(dna) + 2 occurs.

Value

matrix. Each colum contains the motif-count values for one frame. The column names are the values in the start vector. Each row represents one DNA motif. The DNA sequence of the DNA motif is given as row.name.

Author(s)

Wolfgang Kaisers

See Also

countGenomeKmers

Examples

```
seq <- "ATAAATA"
countDnaKmers(seq, 2, 1:3, 3)
```

---

countFastaKmers *countFastaKmers function: Counts DNA k-mers from (compressed) fasta files.*

Description

Reads (compressed) fasta files and counts for DNA k-mers in the sequence.

Usage

```
countFastaKmers(filenames,k=4)
```

Arguments

filenames	character: Vector of fasta file names. Files can be gz compressed.
k	Length of counted DNA k-mers.

Details

Maximal allowed value for k is 12.

Value

matrix.

Note

The static size of the retured k-mer array is 4^k.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-177

Examples

```
basedir <- system.file("extdata", package="seqTools")
filename <- file.path(basedir,"small.fa")
## Not run: writeFai(filename, "small.fa.fai")
res <- countFastaKmers(filename, k=2)
```

---

countGenomeKmers *countGenomeKmers: Counting K-mers in DNA sequences.*

Description

Counts K-mers of DNA sequences inside a vector of DNA sequences. The k-mers are searched in a set of search windows, which are defined by start and width parameter. From each position of the search window, a DNA k-mer is identified on the right hand side on the given DNA sequence. Each value in the start vector defindes the left border of a search window. The size of the search window is given by the appropriate value in the width vector. The function is intended to count DNA k-mers in selected regions (e.g. exons) on DNA chromosomes while respecting strand orientation.

Usage

```
countGenomeKmers(dna, seqid, start, width, strand, k)
```

Arguments

dna	character. Vector of DNA sequences. dna must not contain other characters than "ATCGN". Capitalization does not matter. When a 'N' character is found, the current DNA k-mer is skipped.
seqid	numeric. Vector of (1-based) values describing the index of the analyzed sequences inside the given dna vector.
start	numeric. Vector of (1-based) start positions for reading windows.
width	numeric. Vector of window width values.
strand	factor or numeric. First factor level (or numeric: 1) value will be interpreted as (+)-strand. For any other values, the reversed complement sequence will be counted (in left direction from start value).
k	numeric. Number of nucleotides in tabled DNA motifs. Only a single value is allowed (length(n) = 1!)

Details

The function returns a matrix. Each colum contains the motif-count values for one frame. Each row represents one DNA motif. The DNA sequence of the DNA motif is given as row.name.

Value

matrix.

Author(s)

Wolfgang Kaisers

Examples

```
sq <- "TTTTTCCCCGGGGAAAA"
seqid <- as.integer(c(1, 1))
start <- as.integer(c(6, 14))
width <- as.integer(c(4, 4))
strand <- as.integer(c(1, 0))
k <- 2
countGenomeKmers(sq, seqid, start, width, strand, k)
```

countSpliceKmers *countSpliceKmers: Counting K-mers on donor (5', upstream) sides (exonic) of splice sites.*

Description

The function regards the given string as DNA sequence bearing a collection of splice sites. The given `lEnd` and `rStart` positions act as (1-based) coordinates of the innermost exonic nucleotides. They reside on exon-intron boundaries and have one exonic and one intronic adjacent nucleotide. The function counts `width` k-mers upstream on exonic DNA in reading direction (left -> right on (+) strand, right -> left on (-) strand).

Usage

```
countSpliceKmers(dna, seqid, lEnd, rStart, width, strand, k)
```

Arguments

`dna`	`character`. Vector of DNA sequences. `dna` must not contain other characters than "ATCGN". Capitalization does not matter. When a 'N' character is found, the current DNA k-mer is skipped.
`seqid`	`numeric`. Vector of (1-based) values coding for one of the given sequences.
`lEnd`	`numeric`. Vector of (1-based) left-end positions. Will be used as rightmost window position.
`rStart`	`numeric`. Vector of (1-based) right-start positions. Will be used as leftmost window positions (over which(n-1) positions overhang will be counted as part of frame).
`width`	`numeric`. Vector of window width values.
`strand`	`factor` or `numeric`. First factor level (or numeric: 1) value will be interpreted as (+) strand For any other values, the reversed complement sequence will be counted (in left direction from start value). For (+) strand, the lEnd value will be used as starting position. For (-) strand, the rStart position will be used as starting positions.
`k`	`numeric`. Number of nucleotides in tabled DNA motifs. Only a single value is allowed (length(n) = 1 !)

Details

The function returns a matrix. Each colum contains the motif-count values for one frame. Each row represents one DNA motif. The DNA sequence of the DNA motif is given as row.name.

Value

`matrix`.

Author(s)

Wolfgang Kaisers

Examples

```
seq <- "acgtGTccccAGcccc"
countSpliceKmers(seq, seqid=1, lEnd=4, rStart=10, width=2, strand=1, k=3)
#
sq1 <- "TTTTTCCCCGGGGAAAA"
sq2 <- "TTTTTTTCCCCGGGGAAAA"
sq <- c(sq1, sq2)
seqid <- c( 1, 1, 2, 2)
lEnd  <- c( 9, 9, 11, 11)
rStart <- c(14, 14, 16, 16)
width <- c( 4, 4, 4, 4)
strand <- c( 1, 0, 1, 0)
countSpliceKmers(sq, seqid, lEnd, rStart, width, strand, k=2)
```

---

fastqKmerLocs *fastqKmerLocs function: Counts DNA k-mers position wise from FASTQ files.*

Description

Reads (compressed) FASTQ files and counts for DNA k-mers for each position in sequence.

Usage

```
fastqKmerLocs(filenames, k=4)
```

Arguments

filenames	Vector of FASTQ file names. Files can be gz compressed.
k	Length of counted DNA k-mers.

Details

Maximal allowed value for k is 12.

Value

list. The length of the list equals the number of given filenames. Contains for each given file a matrix with 4^k rows and (maxSeqLen - k + 1) columns (maxSeqLen= maximum read length). The matrix contains for each k-mer and k-mer-start position the counted values.

Note

The static size of the retured k-mer array is 4^k.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
res <- fastqKmerLocs("test_l10_ATCGN.fq", k=2)
res <- fastqKmerLocs("test_l10_atcg.fq", k=2)
res <- fastqKmerLocs("test_l10_ATCGN.fq", k=2)
res <- fastqKmerLocs("test_l6_multi_line.fq", k=2)
```

---

fastqKmerSubsetLocs *fastqKmerSubsetLocs function: Counts for a given DNA k-mer subset position wise from FASTQ files.*

Description

Reads (compressed) FASTQ files and counts for given DNA k-mer subset for each position in sequence. The k-mer subset is given by a vector of k-mer indices. k-mer indices can be obtained from DNA k-mers with the function `kMerIndex`.

Usage

```
fastqKmerSubsetLocs(filenames, k=4, kIndex)
```

Arguments

`filenames`	`character`. Vector of fastqKmerSubsetLocs file names. Files can be gz compressed.
`k`	`integer`. Length of counted DNA k-mers.
`kIndex`	`integer`. Numeric values which represent indices of DNA-k mers.

Details

Maximal allowed value for k is 12.

Value

`list`. The length of the list equals the number of given filenames. Contains for each given file a matrix. Each matrix has one row for each given `kIndex` and an additional row with counts for all other DNA k-mers (labeled `other`). The number of columns equals the maximal sequence length in the FASTQ file.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
k <- 4
kMers <- c("AAAA", "AACC", "AAGG")
kIdx <- kMerIndex(kMers)
res <- fastqKmerSubsetLocs("test_l6.fq", k, kIdx)
```

---

fastqq *fastqq function: Reading summarizing information from FASTQ files.*

Description

Reads read numbers, read lengths, counts per position alphabet frequencies, phred scores and counts per file DNA k-mers from (possibly compressed) FASTQ files.

Usage

```
fastqq(filenames, k=6, probeLabel)
```

Arguments

filenames	Vector of FASTQ file names. Files can be gz compressed.
k	Length of counted DNA k-mers.
probeLabel	character: Textual label for each probe. When probeLabel and filenames have different length, a warning is thrown and the given labels are discarded.

Details

Maximal allowed value for k is 12.

Value

S4 Object of class 'Fastqq'.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq <- fastqq("test_l6.fq")
fq <- fastqq("test_l6_multi_line.fq")
fq <- fastqq("non_exist.fq")
fq <- fastqq("test_l10_ATCGN.fq")
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
                                                    probeLabel=c("g4", "g5"))
```

Fastqq-class *Class* "Fastqq"

Description

Contains quality related summarizing data on FASTQ files.

Objects from the Class

Objects can be created by calls of the form `fastqq("test.fq")`.

Slots

`filenames`: "character": Vector of Fastqq file names.

`probeLabel`: "character": Vector of probe labels.

`nFiles`: "integer": Length of fileNamess.

`k`: "integer": Length of counted DNA k-mers.

`maxSeqLen`: "integer" Maximum sequence length found in FASTQ files. Determines row-number in 'seqLenCount' matrix and column-number in 'nac' and 'phred' slot.

`kmer`: "matrix" Matrix containing DNA k-mers counts.

`firstKmer`: "matrix" Matrix containing count of incipient DNA k-mers.

`nReads`: "integer" Vector containing number of reads per file.

`seqLenCount`: "matrix" Matrix containing Counts of read lengths.

`gcContent`: "matrix" Matrix containing GC content (in percent).

`nN`: "integer" Vector containing Number of N nucleotide entries per file.

`nac`: `"list"` Contains counted per position alphabet frequencies.

`phred`: `"list"` Contains per position phred count tables (one per Fastqq file).

`seqLen`: `"matrix"` Contains minimal and maximal sequence length (one column per file).

`collectTime`: `"list"` Contains start and end time of FASTQ reading as 'POSIXct'.

Methods

The following methods are defined for class `Fastqq`:

Basic accessors:

getK `signature(object="Fastqq")`: Returns k-value (length of DNA k-mers) as `integer`.

kmerCount `signature(object="Fastqq")`: Returns `matrix` with 4^k rows anc `nFiles` columns. For each k-mer and FASTQ-file, the absolute count value of the k-mer in the FASTQ file is given.

nFiles `signature(object="Fastqq")`: Returns number of Files from which data has been collected as `integer`.

nNnucs `signature(object="Fastqq")`: Returns `integer` vector of length `nFiles`. For each FASTQ file, the absolute number of containes 'N' nucleotide entries is given.

nReads `signature(object="Fastqq")`: Returns number of reads in each FASTQ file as `integer`.

fileNames `signature(object="Fastqq")`: Returns number names of FASTQ files from which data has been collected as `character`.

maxSeqLen `signature(object="Fastqq")`: Returns maximum sequence length which has been found in all FASTQ files as `integer`.

seqLenCount `signature(object="Fastqq")`: Returns matrix which tables counted read length in all FASTQ files.

gcContent `signature(object="Fastqq",i="numeric")`: Returns `integer` vector of length 100 which countains absolute read count numbers for each percentage of GC-content. `i` is the index of the FASTQ file for wich the values are returned. The GC content values for all files together can be obtained using `gcContentMatrix`.

nucFreq `signature(object="Fastqq",i="integer")`: Returns `matrix` which contains the absolute nucleotide count values for each nucleotide and read position. `i` is the index of the FASTQ file for wich the values are returned.

seqLen `signature(object="Fastqq")`: Returns `matrix` with two rows and `nFiles` columns. For each file the minimum and maximum read length is given.

kmerCount `signature(object="Fastqq")`: Returns a `matrix` with 4^k rows and `nFiles` columns. Each entry gives the absolute count of the k-mer (given as row name) in each file (given as column name).

phred `signature(object="Fastqq",i="integer")`: Returns a `matrix` with 93 rows and `maxSeqLen` columns. The matrix gived the absolute counts of each phred value for each sequence position. `i` is the index of the FASTQ file for wich the values are returned.

phredQuantiles `signature(object="Fastqq",quantiles="numeric",i="integer")`: Returns a `data.frame`. The data.frame has one row for each given quantile and `maxSeqLen` columns. Each value gives the quantile (given by row name) of the phred values at the sequence position (given by column name). For the `quantiles` argument, a numeric vector with values in [0,1] must be given. For the `i` argument, a single integer value must be given which denotes the index of the FASTQ file from which values are returned (value must be in {1,...,nFiles}).

probeLabel `signature(object="Fastqq")`: Returns `character` vector which contains the `probeLabel` entries for given `Fastqq` object.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

fastqq

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq <- fastqq(c("g4_l101_n100.fq.gz","g5_l101_n100.fq.gz"),
                                k=4, probeLabel=c("g4","g5"))
#
fileNames(fq)
getK(fq)
nNnucs(fq)
nFiles(fq)
nReads(fq)
maxSeqLen(fq)
collectTime(fq)
collectDur(fq)
slc<-seqLenCount(fq)
nf<-nucFreq(fq,1)
nf[1:4,1:10]
seqLen(fq)
probeLabel(fq)
probeLabel(fq) <- 1:nFiles(fq)
#
kc<-kmerCount(fq)
kc[1:10, ]
plotKmerCount(fq)
#
ph<-phred(fq, 1)
ph[25:35,1:15]
pq <- phredQuantiles(fq,c(0.25, 0.5, 0.75), 1)
plotNucFreq(fq, 1)
# Nucleotide count
plotNucCount(fq, 2:3)
# GC content
gcContent(fq, 1)
#
```

```
fqq<-fq[1]
```

`gcContentMatrix` *gcContentMatrix: Returns matrix with read counts for GC content.*

Description

Returns a matrix with read counts. getGCcontent returns a numeric vector with the GC contend (in percent) for each fastq file.

Usage

```
gcContentMatrix(object)
```

Arguments

`object`	`Fastqq`: Object from wich data is copied.

Details

The matrix contains one column for each FASTQ file. Rows labeled from 0 to 100 which represents percent (%) GC content. The matrix contains numbers of reads with the respective proportion of GC (Row 2 contains number of reads with 2% GC content).

Value

`matrix`.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

gcContent

Examples

```
basedir <- system.file("extdata",package="seqTools")
setwd(basedir)
#
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4","g5"))
fqm<-gcContentMatrix(fq)
getGCcontent(fq)
```

---

kMerIndex *kMerIndex function: Returns array index for given DNA k-mers.*

Description

For each k, there exist 4^k DNA k-mers. Many functions inside this package return values where DNA k-mers appear as array indices. `kMerIndex` can be used for extraction of count values for special k-mers by provision of index values.

Usage

```
kMerIndex(kMers, k=nchar(kMers)[1], base=1)
```

Arguments

`kMers`	`character`. Vector of equal sized character strings. The number of characters in each string must be =k (i.e. `all(nchar(kMers)==k)`)
`k`	`integer`. Length of k-mer.
`base`	`integer`. Value must be 0 or 1 (i.e. length(base)==1). For `base=0` the returned index is 0-based (i.e. the index of the first k-mer (AAA..)) is 0. Otherwise the index is 1-based.

Details

Maximal allowed value for k is 12.

Value

`integer`.

Author(s)

Wolfgang Kaisers

Examples

```
kMerIndex(c("AACC", "ATAA"))
kMerIndex(c("AA","AC"), base=1)
kMerIndex(c("AA","AC"), base=0)
```

`meltDownK` *meltDownK: Condensing DNA k-mer count data to lower k-value (i.e. shorter DNA motifs).*

Description

Returns a copy of given object where DNA k-mer counts and first DNA k-mer count table are reduced in size.

Usage

```
meltDownK(object, newK)
```

Arguments

`object`	`Fastqq`: Object from wich data is copied.
`newK`	`integer`: New value for k. Must be >=1 and <= old k.

Details

The function sums all count values which belong the the new motif up. The new motif is the new-k sized prefix of the given k-mer motif.

Value

S4 Object of class 'Fastqq'.

Note

The `meltDownK` mechanism is assotiated with a change of DNA k-mer count values (by its accumulative character). Also, count values from down-melted tables are not identical to directly counted values for lower k. For example counting 'AAAA' with k=1 yields four 'A'. Counting 'AAAA' with k=2 yields three 'AA'. As `meltDownK` sums up count values by prefix k-mers, the melted count table for the second (k=2) count will return three 'A'. Another source for differences may be N-nucleotides. Counting 'AANA' returns three 'A' (using k=1) but only one 'AA' for k=2.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq<-fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4","g5"))
fqm <- meltDownK(fq, 2)
```

`mergedPhred` *mergedPhred functions: Retrieving and plotting of phred quantities from whole* `Fastqq` *objects.*

Description

The `Fastqq` objects contain position-wise counted phred values. The `mergedPhred` function adds the counted values for all FASTQ files together into a single matrix. The matrix then again contains position-wise counted phred values. The `mergedPhredQuantiles` and `plotMergedPhredQuant` are analogues to the `phredQuantiles` and `plotPhredQuant` functions.

Usage

```
mergedPhred(object)
mergedPhredQuantiles(object, quantiles)
plotMergedPhredQuant(object, main, ...)
```

Arguments

`object`	`Fastqq`: Object which contains collected values from `nFiles` FASTQ files.
`quantiles`	`numeric`: Vector of quantiles. All values must be in [0,1].
`main`	`character`: String wich is used as figure caption. Passed internally to plot function.
`...`	Optional arguments which are passed to the `plot` function in `plotMergedPhredQuant`.

Details

The function adds the phred values from all contained FASTQ data.

Value

`mergedPhred` returns a `matrix` with 94 rows and (maxSeqLen + 1) columns. `mergedPhredQuantiles` returns a `data.frame` with one row for each given quantile and max(seqLen(.)) columns. `plotMergedPhredQuant` returns nothing.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771\ Ewing B, Green P Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Research 1998 Vol. 8 No. 3 186-194

Examples

```
basedir <- system.file("extdata",package="seqTools")
setwd(basedir)
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4", "g5"))
#
ph <- mergedPhred(fq)
ph[25:35, 1:15]
pq <- mergedPhredQuantiles(fq, c(0.25, 0.5, 0.75))
plotMergedPhredQuant(fq)
#
```

`mergeFastqq` *mergeFastqq: Merges two Fastqq object into one.*

Description

The contents of two given Fastqq objects are merged together into one resulting Fastqq object.

Usage

```
mergeFastqq(lhs,rhs)
```

Arguments

`lhs`	`Fastqq`.
`rhs`	`Fastqq`.

Details

The data on all FASTQ files in the two incoming objects is merged together. The object has the same internal structure as if the data from all FASTQ files had been collected by a separate call of `fastqq` on the merged FASTQ file names of the arguments. Duplicated probeLabel's are separated by adding of consecutive numbers as suffix to all probeLabel's. When `lhs` and `rhs` contain kmer-counts for different `k` (getK), the function uses the `meltDownK` mechanism in order to equalize the `k` values. Therefore it is possible to compare samples which were counted with different `k` (i.e. k-mer resolution).

Value

S4 Object of class 'Fastqq'.

Note

Note that the meltDownK mechanism is assotiated with a change of DNA k-mer count values. See 'meltDownK' help (note) for more information.

Author(s)

Wolfgang Kaisers

Examples

```
basedir<-system.file("extdata",package="seqTools")
setwd(basedir)
#
lhs<-fastqq("g4_l101_n100.fq.gz",k=4,"g4")
rhs<-fastqq("g5_l101_n100.fq.gz",k=4,"g5")
fq<-mergeFastqq(lhs,rhs)
```

---

phredDist *phredDist: Global relative content of Phred values in Fastqq objects (or subsets).*

Description

The phredDist function returns a named vector with relative Phred content from the whole Fastqq object or a subset which is denoted by a index i. The plotPhredDist function produces a plot of the phredDist values.

Usage

```
phredDist(object, i)
plotPhredDist(object, i, maxp=45, col, ...)
```

Arguments

object	Fastqq: Object which contains collected values from nFiles FASTQ files.
i	integer(optional): Index of FASTQ file(s) from which Phred values are counted. When value is missing, Phred counts for all contained data is returned.
maxp	numeric(optional): Value of maximal plotted phred value (right limit of x-axis).
col	Colour encoding for plotted lines.
...	Additional values passed to plot function.

Details

i must be a numerical vector with values in {1,...,nFiles}. The plotPhredDist function is also prepared for additional arguments: The maxp value denotes the maximal Phred value until which the Phred values are plotted (possibly shrinks the x-Axis). The standard line color is topo.colors(10)[3]. Additional arguments (e.g. main="") can be passed to the plot function.

Value

phredDist returns numeric. plotPhredDist returns nothing.

Author(s)

Wolfgang Kaisers

References

Ewing B, Green P Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Research 1998 Vol. 8 No. 3 186-194

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
        probeLabel=c("g4","g5"))
#
phredDist(fq)
plotPhredDist(fq, main="g4 and g5")
#
```

---

phredTable *phredTable: Returns a data.frame with phred encodings.*

Description

The function calculates characters and corresponding ascii values for a given range of phred values. As default, a data.frame with all valid phred values {0,...,93} is returned.

Usage

```
phredTable(phred)
```

Arguments

phred	numeric. Vector with phred values. All values must be in 0:93

Value

data.frame. The data.frame has three columns: "ascii","phred" and "char"

Author(s)

Wolfgang Kaisers

References

Ewing B, Green P Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Research 1998 Vol. 8 No. 3 186-194

See Also

char2ascii

Examples

```
phredTable()
```

`plotGCcontent` *plotGCcontent: Plots the proportions of relative GC content for all FASTQ files.*

Description

The function creates plots on proportions of relative GC content. For each FASTQ file from wich data is contained, one separate line is plotted. A value of 0.1 at the proportion of 40 says that 0.1 % of the reads have 40 % GC content.

Usage

```
plotGCcontent(object, main, ...)
```

Arguments

`object`	`Fastqq`: Object which contains collected values from `nFiles` FASTQ files.
`main`	`integer`(optional): The main title displayed on top of the plot. When missing, a standard text is printed.
`...`	Other arguments which are passed to the internally called plot function.

Details

The area under each plotted line adds up to 1.

Value

None.

Author(s)

Wolfgang Kaisers

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4", "g5"))
#
plotGCcontent(fq)
```

`plotKmerCount` *plotKmerCount: Creation of plots DNA for k-mer counts from Fastqq objects.*

Description

The function creates plots from counted DNA k-mers from Fastqq objects.

Usage

```
plotKmerCount(object,index,mxey,main="K-mer count",...)
```

Arguments

`object`	`Fastqq`: Object which contains collected values from `nFiles` FASTQ files.
`index`	`integer`(optional): Index of FASTQ file(s) for which data is plotted. When value is missing, k-mer counts for all contained data is plotted.
`mxey`	`integer`(optional): Maximal value for y axis, given by power of 2 (when mxey=4, then maximal ylim value is 2^4 = 16). Allows overriding of automatic calculated values.
`main`	`character`(optional): Caption text which printed into the output.
`...`	Additional parameters which are passed down to the plot function.

Details

Values for i must be in {1,...,nFiles}. The function shrinks the k-mer count table down to size of 4096 (k = 6) when k > 6 in order to limit the complexity of the plot.

Value

None.

Note

The static size of the retured k-mer array is 4^k.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata",package="seqTools")
setwd(basedir)
#
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4", "g5"))
#
plotKmerCount(fq)
plotKmerCount(fq,1)
plotKmerCount(fq, 1:2)
#
```

`plotNucCount` *plotNucCount: Plots nucleotide counts from Fastqq objects.*

Description

The function creates plots from nucleotide counts from Fastqq objects.

Usage

```
plotNucCount(object, nucs=16, maxx,...)
```

Arguments

`object`	`Fastqq`: Object which contains collected values from `nFiles` FASTQ files.
`nucs`	`integer`(optional): Index of nucleotides for which data is plotted. When value is missing, k-mer counts for all contained data is plotted.
`maxx`	`integer`(optional): When given, nucleotide counts are plotted for the first maxx nucleotide positions. This option is used for displaying detailed plots from the first read nucleotide positions (which are sometimes not equally distributed).
`...`	(currently unused).

Details

Values for i must be in {1,...,nFiles}. The nucs index encodes for IUPAC characters as shown in the following table.

1	A	|	6	R	|	11	M	|	16	N
2	C	|	7	Y	|	12	B	|	17	.
3	G	|	8	S	|	13	D	|	18	-
4	T	|	9	W	|	14	H	|	19	=
5	U	|	10	K	|	15	V	|	20	”

When count values for 'A' are to be plotted, 'nucs' must be =1. When count values for 'GC' are to be plotted, 'nucs' must be c(2,3).

Value

None.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
#
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4", "g5"))
#
plotNucCount(fq)
plotNucCount(fq, 1)
plotNucCount(fq, 1:2)
#
```

`plotNucFreq` *plotNucFreq: Plots the position wise relative nucleotide content for nucleotides 'A','C','G','T'.*

Description

The function creates plots on position wise relative nucleotide content single FASTQ files.

Usage

```
plotNucFreq(object, i, main, maxx, ...)
```

Arguments

object	Fastqq: Object which contains collected values from nFiles FASTQ files.
i	integer(optional): Index FASTQ file for which nucleotide frequencies are plotted.
main	integer(optional): The main title displayed on top of the plot. When missing, a standard text is printed.
maxx	integer(optional): Determines the maximum sequence position for which counts are plotted. Small values (e.g. 15) allow plotting the distribution on the first nucleotides at larger resolution (see reference).
...	Other arguments which are passed to the internally called plot function.

Value

None.

Author(s)

Wolfgang Kaisers

References

Hansen KD, Brenner SE, Dudoit S. Biases in Illumina transcriptome sequencing caused by random hexamer priming. Nucleic Acids Research 2010 Vol.38 No.12 e131, doi: 10.1093/nar/gkq224

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
#
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4","g5"))
#
plotNucFreq(fq, 2)
# Same plot without x-axis
plotNucFreq(fq, 2, xaxt="n")
#
plotNucFreq(fq, 1, maxx=15)
```

plotPhredQuant *plotPhredQuant: Plots the position wise 10%, 25 %, 50%, 75% and 90 % quantiles of phred values.*

Description

The function creates plots which describes the position wise distribution of phred quantiles in single FASTQ files.

Usage

```
plotPhredQuant(object, i, main, ...)
```

Arguments

object	Fastqq: Object which contains collected values from nFiles FASTQ files.
i	integer(optional): Index FASTQ file for which phred quantiles are plotted.
main	integer(optional): The main title displayed on top of the plot. When missing, a standard text is printed.
...	Other arguments which are passed to the internally called plot function.

Value

None.

Author(s)

Wolfgang Kaisers

See Also

Fastqq-class

Examples

```
basedir <- system.file("extdata",package="seqTools")
#
setwd(basedir)
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
        probeLabel=c("g4", "g5"))
#
plotPhredQuant(fq, 2)
# Same plot without x-axis
plotPhredQuant(fq,2, xaxt="n")
```

propPhred *propPhred: Lane specific proportion of reads in a specified Phred-region.*

Description

The propPhred function returns a named vector with relative Phred content for all contained lanes.

Usage

```
propPhred(object, greater = 30, less = 93)
```

Arguments

object	Fastqq: Object which contains collected values from FASTQ files.
greater	numeric: Limits the counted proportion of phred to values which are greater than the given value (default: 30).
less	numeric: Limits the counted proportion of phred to values which are less than the given value (default: 93).

Details

The greater and less arguments must be numeric, have length 1 and be >0 and < 94. greater must be less than less. With the default settings the reported proportions should be >50 % for all lanes in order to be acceptable (see 't Hoen et. al.).

Value

Numeric.

Author(s)

Wolfgang Kaisers

References

't Hoen et.al Reproducibility of high-throughput mRNA and small RNA sequencing across laboratories Nature Biotechnology 2013 Vol. 31 1015 - 1022 (doi:10.1038/nbt.2702)

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
#
fq <- fastqq(c("g4_l101_n100.fq.gz", "g5_l101_n100.fq.gz"), k=4,
    probeLabel=c("g4", "g5"))
# Proportion of phred Values >30
propPhred(fq)
```

```
# Proportion of phred Values >10 and < 30
propPhred(fq, greater=10, less=30)
```

`revCountDnaKmers` *revCountDnaKmers: Counting K-mers in DNA sequence.*

Description

Counts DNA K-mers for reverse complement of given DNA sequence. The k-mers are counted in a set of search windows, which are defined by `start` and `width` parameter. From each position of the search window, a DNA k-mer is identified on the left hand side on the reverse complement of the given DNA sequence. Each value in the `start` vector defines the right border of a search window. The size of the search window is given by the appropriate value in the `width` vector.

Usage

```
revCountDnaKmers(dna, k ,start, width)
```

Arguments

`dna`	`character`. Single DNA sequence (vector of length 1). `dna` must not contain other characters than "ATCGN". Capitalization does not matter. When a 'N' character is found, the ongoing identification of a DNA k-mer is terminated.
`k`	`numeric`. Number of nucleotides in tabled DNA motifs.
`start`	`numeric`. Vector of (1-based) start positions for reading frames.
`width`	`numeric`. Defines number of k-mers (size of search window) for each start position. Must have the same length as `start` or length 1 (in which case the values of `width` are recycled.)

Details

The start positions for identification of DNA k-mers are all positions in {start-width+1,...,start}. In order to prevent counting before the first nucleotide of the DNA sequence, all start values must be >= width + k. The function throws an error when this border is exceeded.

Value

`matrix`. Each colum contains the motif-count values for one frame. Each row represents one DNA motif. The DNA sequence of the DNA motif is given as row.name.

Author(s)

Wolfgang Kaisers

Examples

```
rseq <- "TATTAT"
revCountDnaKmers(rseq, 2,6:4, 2)
```

`simFastqqRunTimes` *simFastqqRunTimes: For given values of k and nSeq the function creates FASTQ files with simulated data, collects k-mer data with the fastqq function and reports the run times for the data collection.*

Description

For each combination of the parameters k and nSeq, the function writes one FASTQ file and collects the data. The FASTQ files are equally structured: Each read contains 17 randomly selected DNA 6-mers. Therefore the read-length is always 102.

Usage

```
simFastqqRunTimes(k, nSeq, filedir=".")
```

Arguments

`k`	`numeric`. k-mer sizes which are passed to `fastqq`. Default value is 2:15.
`nSeq`	`numeric`. Number of simulated reads in FASTQ-file. Default value is (100, 1000, ..., 10000000).
`filedir`	`character`. The output can be placed in a separate directory. When not existant, the function tries to create 'filedir'. The function throws an error when writing is not permitted in the given directory (Could not open file ...).

Details

The FASTQ files contain the parameter settings inside their filename. The files are created with 'writeSimFastq'.

Value

`data.frame`. The data frame has four columns: id, k, nSeq and runtime.

Author(s)

Wolfgang Kaisers

Examples

```
## Not run:
res <- simFastqqRunTimes(k=2:9, nSeq=100000)
plot(runtime~k,res,type="b")

## End(Not run)
```

`sim_fq` *sim_fq: Performs an experimental series of separation capabilities of hierarchical clustering (HC) based on DNA k-mers in FASTQ files using simulated DNA content.*

Description

Writes compressed FASTQ files where sequence sections contain concatenated k-mers which are uniformly distributed in the range of k-mers for given k. The function first writes a batch of randomly FASTQ files containing randomly simulated DNA sequence. In a second step the function repeatedly writes FASTQ files with random DNA sequence where a fraction of the reads is 'contaminated' with given DNA k-mers. In a third step, for each set of simulated and contaminated files, a hierarchical cluster (HC) tree based on DNA k-mers is calculated. For each set of files, the size of the smaller fraction in the first half of the tree is counted (perc). The value can be used as measure for separation capability of the HC algorithm.

Usage

```
sim_fq(nRep=2, nContamVec=c(100, 1000), grSize=20, nSeq=1e4,
    k=6, kIndex=1365, pos=20)
```

Arguments

`nRep`	`numeric`. Number of replicates for each combination of each nContamVec value
`nContamVec`	`numeric`. Vector with nContam (absolute number of contaminated reads) values.
`grSize`	`numeric`. Number FASTQ files in control and contamination group.
`nSeq`	`numeric`. Number of reads per FASTQ file.
`k`	`numeric`. k value used in fastqq function.
`kIndex`	`numeric`. k-mer index of inserted k-mer(s). The k-mer index can be retreaved for a given k-mer with 'kMerIndex'. Default value is 1365 (="CCCCCC").
`pos`	`numeric`. Determines at which position in sequence the k-mer is inserted. 1-based (1=first position).

Details

The function is intended to be used as explorative tool (not for routine quality assessment). There are some files written and there will be a lot of output on the terminal. It is therefore recommended to switch to a separate working directory and to run this function on a separate terminal. The function is not exported.

Value

`data.frame` containing results of the counted perc values for each repetition of the simulation.

Author(s)

Wolfgang Kaisers

Examples

```
kMerIndex("CCCCCC")
## Not run: res <- seqTools:::sim_fq(nRep=2, nContamVec=c(10, 100),
                    grSize=4, nSeq=1e2)
## End(Not run)
```

`trimFastq` *trimFastq: Performs sequence removal, trimming (fixed and quality based) and nucleotide masking on FASTQ files.*

Description

Fastq files sometimes need to be preprocessed before alignment. Three different mechanisms come into use here: Discarding whole reads, trimming sequences and masking nucleotides. This function performs all three mechanisms together in one step. All reads with insufficient phred are discarded. The reads can be trimmed ad each terminal side (on trim of fixed size and a trim based on quality thresholds).

Usage

```
trimFastq(infile, outfile="keep.fq.gz", discard="disc.fq.gz",
        qualDiscard=0, qualMask=0, fixTrimLeft=0,
        fixTrimRight=0, qualTrimLeft=0, qualTrimRight=0,
        qualMaskValue=78, minSeqLen=0)
```

Arguments

`infile`	`character`. Input FASTQ file. Only one infile is allowed per function call.
`outfile`	`character`. Output FASTQ file.
`discard`	`character`. Output file in which discarded reads are written.
`qualDiscard`	`numeric`. All reads which contain one or more phred scores <qualDiscard will be discarded (i.e. output to discard).
`qualMask`	`numeric`. All nucleotides for which phred score < qualMask will be overwritten with `qualMaskValue`.
`fixTrimLeft`	`numeric`. Prefix of this size will be trimmed.
`fixTrimRight`	`numeric`. Suffix of this size will be trimmed.
`qualMaskValue`	`numeric`. ASCII replace value for masked nucleotides
`qualTrimLeft`	`numeric`. Prefix where all phred scores are < qualTrimLeft will be trimmed.
`qualTrimRight`	`numeric`. Suffix where all phred scores are < qualTrimRight will be trimmed.
`minSeqLen`	`numeric`. All reads where sequence length after (fixed and quality based) trimming is <minSeqLen will be descarded (i.e. output to discard).

Details

The function divides the input file into two outputs: The output file (contains the accepted reads) and the discard file (contains the excluded reads). After trim operations, the function checks for remaining read length. When the read length is smaller than minSeqLen, the read will be discarded.

Value

Numeric. A vector of length 2 which contains the number of reads which are written to output and to discard

Author(s)

Wolfgang Kaisers

References

Ewing B, Green P Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Research 1998 Vol. 8 No. 3 186-194

Examples

```
basedir <- system.file("extdata", package="seqTools")
setwd(basedir)
trimFastq("sim.fq.gz", qualDiscard=10, qualMask=15, fixTrimLeft=2,
    fixTrimRight=2, qualTrimLeft=28, qualTrimRight=30, minSeqLen=5)
```

---

writeFai *writeFai: Create FASTA index file.*

Description

The function reads a FASTA file and produces a FASTA index file as output.

Usage

```
writeFai(infiles, outfiles)
```

Arguments

infiles	character. Vector of FASTA file names for which FASTA index is to be written.
outfiles	character. Vector file names for writing FASTA index to.

Value

None.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
## Not run:
infile <- system.file("extdata", "small.fa", package="seqTools")
writeFai(infile, "small.fa.fai")

## End(Not run)
```

`writeSimContFastq` *writeSimContFastq: Create FASTQ files with simulated k-mer sequences*

Description

Writes compressed FASTQ files where sequence sections contain concatenated k-mers which are uniformly distributed in the range of k-mers for given k. A fraction of the reads can be contaminated with one or more deterministic k-mers.

Usage

```
writeSimContFastq(k=6, nk=5, nSeq=10, pos=1,
       kIndex=1, nContam=nSeq, filename="simc.fq.gz")
```

Arguments

`k`	`numeric`. Length of k-mer. Default value is 6.
`nk`	`numeric`. Number of k-mers in each FASTQ read. Default value is 5.
`nSeq`	`numeric`. Number of simulated reads in FASTQ-file. Default value is 10.
`pos`	`numeric`. Determines at which position in sequence the k-mer is inserted. 1-based (1=first position).
`kIndex`	`numeric`. k-mer index of inserted k-mer. The k-mer index can be retreaved for a given k-mer with 'kMerIndex'.
`nContam`	`numeric`. Absolute number of contaminated reads. The k-mer's are inserted at the firsts 'nContam' reads of the sequence array.
`filename`	`character`. Name of written (compressed) FASTQ file.

Details

The read headers are consequtive numbered. The phred quality values are equally set to 46 (='.') which represents a phred value of 13. This function is not designed for routine use. The random content FASTQ files can be used in order to measure the separation capabilities of hierarchical clustering mechanisms.

Value

None.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
## Not run: writeSimContFastq()
```

`writeSimFastq` *writeSimFastq: Create FASTQ files with simulated DNA k-mer sequences*

Description

Writes compressed FASTQ files where sequence sections contain concatenated k-mers which are uniformly distributed in the range of k-mers for given k.

Usage

```
writeSimFastq(k=6, nk=5, nSeq=10, filename="sim.fq.gz")
```

Arguments

`k`	`numeric`. Length of k-mer. Default value is 6.
`nk`	`numeric`. Number of k-mers in each FASTQ read. Default value is 5.
`nSeq`	`numeric`. Number of simulated reads in FASTQ-file. Default value is 10.
`filename`	`character`. Name of written (compressed) FASTQ file.

Details

The read headers are consequtive numbered. The phred quality values are equally set to 46 (='.') which represents a phred value of 13. This function is not designed for routine use. The random content FASTQ files can be used in order to measure the separation capabilities of hierarchical clustering mechanisms.

Value

None.

Author(s)

Wolfgang Kaisers

References

Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM The sanger FASTQ file format for sequences with quality scores and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 2010 Vol.38 No.6 1767-1771

Examples

```
writeSimFastq()
```

Index

∗ **FASTQ**
meltDownK, 19
seqTools-package, 2
∗ **Fastqq**
fastqq, 13
meltDownK, 19
∗ **ascii2char**
ascii2char, 3
∗ **cbDistMatrix**
cbDistMatrix, 4
∗ **classes**
Fastqq-class, 14
∗ **collectDur**
collectDur, 5
∗ **countDnaKmers**
countDnaKmers, 6
∗ **countFastaKmers**
countFastaKmers, 7
∗ **countGenomeKmers**
countGenomeKmers, 8
∗ **countSpliceKmers**
countSpliceKmers, 10
∗ **fasta**
seqTools-package, 2
∗ **fastqKmerLocs**
fastqKmerLocs, 11
∗ **fastqKmerSubsetLocs**
fastqKmerSubsetLocs, 12
∗ **fastqq**
fastqq, 13
Fastqq-class, 14
gcContentMatrix, 17
∗ **gcContentMatrix**
gcContentMatrix, 17
∗ **kMerIndex**
kMerIndex, 18
∗ **kmer**
cbDistMatrix, 4
countFastaKmers, 7
fastqKmerLocs, 11
fastqKmerSubsetLocs, 12
fastqq, 13
Fastqq-class, 14
gcContentMatrix, 17
meltDownK, 19
plotGCcontent, 24
plotKmerCount, 25
plotNucCount, 26
plotNucFreq, 27
plotPhredQuant, 29
∗ **meltDownK**
meltDownK, 19
∗ **mergeFastqq**
mergeFastqq, 21
∗ **mergedPhredQuantiles**
mergedPhred, 20
∗ **mergedPhred**
mergedPhred, 20
∗ **phredDist**
phredDist, 22
∗ **phredTable**
phredTable, 23
∗ **plotGCcontent**
plotGCcontent, 24
∗ **plotKmerCount**
plotKmerCount, 25
∗ **plotMergedPhredQuant**
mergedPhred, 20
∗ **plotNucCount**
plotNucCount, 26
∗ **plotNucFreq**
plotNucFreq, 27
∗ **plotPhredDist**
phredDist, 22
∗ **plotPhredQuant**
plotPhredQuant, 29
∗ **propPhred**
propPhred, 30

∗ **revCountDnaKmers**
revCountDnaKmers, 31
∗ **simFastqqRunTimes**
simFastqqRunTimes, 32
∗ **sim_fq**
sim_fq, 33
∗ **trimFastq**
trimFastq, 34
∗ **writeFai**
writeFai, 35
∗ **writeSimContFastq**
writeSimContFastq, 36
∗ **writeSimFastq**
writeSimFastq, 37
[,Fastqq-method *(*Fastqq-class*)*, 14
[-methods *(*Fastqq-class*)*, 14

ascii2char, 3

cbDistMatrix, 4
cbDistMatrix,Fastqq-method *(*cbDistMatrix*)*, 4
cbDistMatrix-methods *(*cbDistMatrix*)*, 4
char2ascii *(*ascii2char*)*, 3
collectDur, 5
collectDur,Fastqq-method *(*collectDur*)*, 5
collectDur-methods *(*collectDur*)*, 5
collectTime *(*collectDur*)*, 5
collectTime,Fastqq-method *(*collectDur*)*, 5
collectTime-methods *(*collectDur*)*, 5
countDnaKmers, 6
countFastaKmers, 7
countGenomeKmers, 8
countSpliceKmers, 10

fastqKmerLocs, 11
fastqKmerSubsetLocs, 12
fastqq, 13
Fastqq-class, 14
fileNames *(*Fastqq-class*)*, 14
fileNames,Fastqq-method *(*Fastqq-class*)*, 14
fileNames-methods *(*Fastqq-class*)*, 14

gcContent *(*Fastqq-class*)*, 14
gcContent,Fastqq-method *(*Fastqq-class*)*, 14
gcContent-methods *(*Fastqq-class*)*, 14
gcContentMatrix, 17
gcContentMatrix,Fastqq-method *(*gcContentMatrix*)*, 17
gcContentMatrix-methods *(*gcContentMatrix*)*, 17
getGCcontent *(*gcContentMatrix*)*, 17
getGCcontent,Fastqq-method *(*gcContentMatrix*)*, 17
getGCcontent-methods *(*gcContentMatrix*)*, 17
getK *(*Fastqq-class*)*, 14
getK,Fastqq-method *(*Fastqq-class*)*, 14
getK-methods *(*Fastqq-class*)*, 14

kmerCount *(*Fastqq-class*)*, 14
kmerCount,Fastqq-method *(*Fastqq-class*)*, 14
kmerCount-methods *(*Fastqq-class*)*, 14
kMerIndex, 18

maxSeqLen *(*Fastqq-class*)*, 14
maxSeqLen,Fastqq-method *(*Fastqq-class*)*, 14
maxSeqLen-methods *(*Fastqq-class*)*, 14
meltDownK, 19
meltDownK,Fastqq-method *(*meltDownK*)*, 19
meltDownK-methods *(*meltDownK*)*, 19
mergedPhred, 20
mergedPhred,Fastqq-method *(*mergedPhred*)*, 20
mergedPhred-methods *(*mergedPhred*)*, 20
mergedPhredQuantiles *(*mergedPhred*)*, 20
mergedPhredQuantiles,Fastqq-method *(*mergedPhred*)*, 20
mergedPhredQuantiles-methods *(*mergedPhred*)*, 20
mergeFastqq, 21
mergeFastqq,Fastqq-method *(*mergeFastqq*)*, 21
mergeFastqq-methods *(*mergeFastqq*)*, 21

nFiles *(*Fastqq-class*)*, 14
nFiles,Fastqq-method *(*Fastqq-class*)*, 14
nFiles-methods *(*Fastqq-class*)*, 14
nNnucs *(*Fastqq-class*)*, 14
nNnucs,Fastqq-method *(*Fastqq-class*)*, 14
nNnucs-methods *(*Fastqq-class*)*, 14
nReads *(*Fastqq-class*)*, 14
nReads,Fastqq-method *(*Fastqq-class*)*, 14

nReads-methods (Fastqq-class), 14
nucFreq (Fastqq-class), 14
nucFreq,Fastqq-method (Fastqq-class), 14
nucFreq-methods (Fastqq-class), 14

phred (Fastqq-class), 14
phred,Fastqq-method (Fastqq-class), 14
phred-methods (Fastqq-class), 14
phredDist, 22
phredDist,Fastqq-method (phredDist), 22
phredDist-methods (phredDist), 22
phredQuantiles (Fastqq-class), 14
phredQuantiles,Fastqq-method (Fastqq-class), 14
phredQuantiles-methods (Fastqq-class), 14
phredTable, 23
plotGCcontent, 24
plotGCcontent,Fastqq-method (plotGCcontent), 24
plotGCcontent-methods (plotGCcontent), 24
plotKmerCount, 25
plotKmerCount,Fastqq-method (plotKmerCount), 25
plotKmerCount-methods (plotKmerCount), 25
plotMergedPhredQuant (mergedPhred), 20
plotMergedPhredQuant,Fastqq-method (mergedPhred), 20
plotMergedPhredQuant-methods (mergedPhred), 20
plotNucCount, 26
plotNucCount,Fastqq-method (plotNucCount), 26
plotNucCount-methods (plotNucCount), 26
plotNucFreq, 27
plotNucFreq,Fastqq-method (plotNucFreq), 27
plotNucFreq-methods (plotNucFreq), 27
plotPhredDist (phredDist), 22
plotPhredDist,Fastqq-method (phredDist), 22
plotPhredDist-methods (phredDist), 22
plotPhredQuant, 29
plotPhredQuant,Fastqq-method (plotPhredQuant), 29
plotPhredQuant-methods (plotPhredQuant), 29

probeLabel (Fastqq-class), 14
probeLabel,Fastqq-method (Fastqq-class), 14
probeLabel-methods (Fastqq-class), 14
probeLabel<- (Fastqq-class), 14
probeLabel<-,Fastqq-method (Fastqq-class), 14
probeLabel<--methods (Fastqq-class), 14
propPhred, 30
propPhred,Fastqq-method (propPhred), 30
propPhred-methods (propPhred), 30

revCountDnaKmers, 31

seqLen (Fastqq-class), 14
seqLen,Fastqq-method (Fastqq-class), 14
seqLen-methods (Fastqq-class), 14
seqLenCount (Fastqq-class), 14
seqLenCount,Fastqq-method (Fastqq-class), 14
seqLenCount-methods (Fastqq-class), 14
seqTools (seqTools-package), 2
seqTools-package, 2
sim_fq, 33
simFastqqRunTimes, 32

trimFastq, 34

writeFai, 35
writeSimContFastq, 36
writeSimFastq, 37