Package 'genBaRcode'

September 21, 2017

Title Analysis and Visualization Tools for Genetic Barcode Data

Version 1.0.0

Author Lars Thielecke <lars.thielecke@tu-dresden.de>

Maintainer Lars Thielecke <lars.thielecke@tu-dresden.de>

Description Provides the necessary functions to identify and extract a selection of already available barcode constructs (Cornils, K. et al. (2014) <doi:10.1093/nar/gku081>) and freely choosable barcode designs from next generation sequence (NGS) data. Furthermore, it offers the possibility to account for sequence errors, the calculation of barcode similarities and provides a variety of visualisation tools (Thielecke, L. et al. (2017) <doi:10.1038/srep43249>).

Depends R (>= 3.4.0)

License LGPL

Encoding UTF-8

LazyData true

Suggests testthat

Imports methods, Biostrings, RColorBrewer, ShortRead, ape, ggnetwork, ggplot2, igraph, network, phangorn, stringdist, visNetwork, reshape2, S4Vectors, shiny

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-09-21 17:11:30 UTC

R topics documented:

as.BCdat 2
BCdat-class 3
BC_dat 3
checkBarcodeData 4
createGephiFile 4
errorCorrection 5
extractBarcodes 5

genBaRcode_app 6
generateKirchenplotBCdis 7
generateKirchenplotRCdis 7
generateTimeSeriesData 8
ggplotHammDistGraph 9
hybridsIdentification 10
ori_BCs 10
plotClusterTree 11
plotHammDistIgraph 11
plotHammDistVisNetwork 12
plotNucFrequency 13
plotQualityScoreDis 13
plotQualityScorePerCycle 14
plotTimeSeries 15
processingRawData 16
qualityFiltering 17
readBCdat 18

Index **19**

`as.BCdat` *Data Type Conversion*

Description

Converts a data.frame into a BCdat object.

Usage

```
as.BCdat(dat, label = "without_label", mask = "", resDir = getwd())
```

Arguments

`dat`	a data.frame object with two columns containing read counts and barcode sequences.
`label`	a optional character string used as label.
`mask`	a optional character string, describing the barcode backbone structure.
`resDir`	a optional character string, identifying the path to the results directory, default is current working directory.

Value

a BCdat object.

BCdat-class *BCdat class*

Description

S4 data class containing every relevant information.

Value

a BCdat object.

Slots

`reads` data.frame containing barcode sequences and their corresponding read counts.

`results_dir` character string of the working directory path.

`label` character string identifying the particular experiment (will be part of the names of any file created).

`mask` character string of the used barcode design.

BC_dat *Barcode distribution of an example experiment.*

Description

A dataset containing an example BCdat object which consists of 301 barcode sequences.

Usage

```
BC_dat
```

Format

A S4 data object with the following slots:

class sequence overview

barcode read counts a data frame consisting of read counts and barcode sequences

results dir path to a directory for any kind of results

barcode layout a string clarifying the barcode backbone structure

label character string, used as label for file names etc.

Details

BC_dat:

checkBarcodeData *checkBarcodeData*

Description

Checks data slots of BCdat object for correctness.

Usage

```
checkBarcodeData(object)
```

Arguments

object	a BCdat object.

createGephiFile *Creating a Gephi File*

Description

createGephiFile creates a data file usable with the free graph visualisation tool gephi. The nodes represent barcodes and its respective size reflects the corresponding read counts. Edges between nodes indicate a hamming distance between two barcodes of maximal minHD. If ori_BCs is provided the node color refelects the hamming distance of a particular barcode to one of the provided barcode sequences.

Usage

```
createGephiFile(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL,
  col_type = "rainbow")
```

Arguments

BC_dat	a BCdat object
minHD	an integer value representing the maximal hamming distance value for which the graph will contain edges
loga	a logical value indicating the use or non-use of logarithmic read count values
ori_BCs	a vector of character strings containing the initial barcode sequences
col_type	character sting, choosing one of the available color palettes

Examples

```
## Not run:

data(BC_dat)
createGephiFile(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL, col_type = "rainbow")


## End(Not run)
```

`errorCorrection` *Error Correction*

Description

Corrects a list of equally long (barcode) sequences. Based on calculated hamming distances as a measure of similarity, highly similar sequences are clustered together and the cluster label will be the respective sequence with the highest read count.

Usage

```
errorCorrection(BC_dat, maxHD, save_it = FALSE)
```

Arguments

`BC_dat`	a BCdat object, containing the necessary sequences.
`maxHD`	an integer value representing the maximal hamming distance for which it is allowed to cluster two sequences together.
`save_it`	a logical value. If TRUE the data will be saved as csv-file.

Examples

```
errorCorrection(BC_dat, maxHD = 8, save_it = FALSE)
```

`extractBarcodes` *Barcode extraction*

Description

Extracts barcodes according to the given barcode design from a fastq file.

Usage

```
extractBarcodes(dat, label, results_dir, mismatch = 0, indels = FALSE,
  bc_pattern, cpus = 1)
```

Arguments

dat	a ShortReadQ object.
label	a character string.
results_dir	a character string which contains the path to the results directory.
mismatch	an positive integer value, default is 0, if greater values are provided they indicate the number of allowed mismatches when identifing the barcode constructe.
indels	under construction.
bc_pattern	a character string describing the barcode design, variable positions have to be marked with the letter 'N'.
cpus	a positive integer identifying the number of usable CPUs.

Value

a frequency table of barcode sequences.

Examples

```
## Not run:

bc_pattern <- "ACTNNCGANNCTTNNCGANNCTTNNGGANNCTANNACTNNCGANNCTTNNCGANNCTTNNGGANNCTANNACTNNCGANN"
source_dir <- system.file("extdata", package = "genBaRcode")
dat <- ShortRead::readFastq(dirPath = source_dir, pattern = "test_data.fastq")

extractBarcodes(dat, label = "test", results_dir = getwd(), mismatch = 0,
indels = FALSE, bc_pattern)


## End(Not run)
```

---

genBaRcode_app *Shiny App*

Description

Launches the corresponding shiny app.

Usage

```
genBaRcode_app(dat_dir = system.file("extdata", package = "genBaRcode"))
```

Arguments

dat_dir	a character string, identifying the path to one or more fast(q) files which shall be analysed, default is the path to the package inherent example fastq file

generateKirchenplotBCdis *Plotting a Kirchenplot*

Description

Generates a barplot based on read counts. If `ori_BCs` is provided the bar color reflects the hamming distance of a particular barcode to one of the provided initial barcode sequences.

Usage

```
generateKirchenplotBCdis(BC_dat, ori_BCs = NULL, loga = TRUE,
  col_type = NULL)
```

Arguments

`BC_dat`	a BCdat object.
`ori_BCs`	a vector of character strings containing the initial barcode sequences.
`loga`	a logical value, indicating the use or non-use of logarithmic read count values.
`col_type`	character sting, choosing one of the availabe color palettes, e.g. rainbow

Value

a ggplot2 object

Examples

```
data(BC_dat)
generateKirchenplotBCdis(BC_dat, ori_BCs, loga = TRUE, col_type = NULL)
```

generateKirchenplotRCdis *Plotting a Kirchenplot*

Description

Generates a barplot visualising the read count distribution.

Usage

```
generateKirchenplotRCdis(BC_dat, b = 30, show_it = FALSE)
```

Arguments

BC_dat	a BCdat object.
b	an integer value, defining the number of bins.
show_it	a logical vaue. If TRUE, the respective values are printed on the console?

Value

ggplot2 object

Examples

```
data(BC_dat)
generateKirchenplotRCdis <- function(BC_dat, b = 10, show_it = TRUE)
```

---

generateTimeSeriesData

Generating Time Series Data Object

Description

Generates a matrix containing barcodes sequences as rows and consecutive measurements at columns. It serves as the necessary data object for the plotting function 'plotTimeSeries'.

Usage

```
generateTimeSeriesData(BC_dat_list)
```

Arguments

BC_dat_list	a list of BCdat objects.

Value

a data.frame containing every identified barcode and its read count per time point/measurement.

ggplotHammDistGraph *Plotting a Hamming Distance Network*

Description

ggplotHammDistGraph will create a graph-like visualisation (ripple plot) of the corresponding barcode sequences and their similarity based on the ggplot2 and the ggnetwork packages. The nodes represent the barcode sequences and their respective size reflects the corresponding read counts. Edges between nodes indicate a hamming distance between two barcodes of maximal `minHD`. If `ori_BCs` is provided the node color also refelects the hamming distance of a particular barcode to one of the initial barcodes.

Usage

```
ggplotHammDistGraph(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL,
  lay = "fruchtermanreingold", complete = TRUE, col_type = "rainbow",
  outline = 0.5)
```

Arguments

`BC_dat`	a BCdat object.
`minHD`	an integer value representing the maximal hamming distance for which the graph will contain edges.
`loga`	a logical value, indicating the use or non-use of logarithmic read count values.
`ori_BCs`	a vector of character strings containing the initial barcode sequences.
`lay`	a character string, identifying the prefered layout algorithm (see ggnetwork layout option).
`complete`	a logical value. If TRUE, every node will have at least one edge.
`col_type`	a character sting, choosing one of the available color palettes.
`outline`	an integer value which adjusts the thickness of the black outline of each node.

Value

a ggplot2 object

Examples

```
## Not run:

data(BC_dat)
ggplotHammDistGraph(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL, lay = "fruchtermanreingold",
complete = FALSE, col_type = "rainbow")


## End(Not run)
```

hybridsIdentification *Identifies hybrid barcodes*

Description

Experimental function to identify hybrid barcodes which can occure due to unfinished synthesis of a template in-between PCR cycles.

Usage

```
hybridsIdentification(dat, min_seq_length = 2)
```

Arguments

dat	a character vector containing barcode sequences or a BCdat object.
min_seq_length	a positive integer value indicating the minimal length of the two barcodes which give rise to a hybrid barcode.

Value

a hybrid-free frequency table of barcode sequences

Examples

```
data(BC_dat)
hybridsIdentification(BC_dat, min_seq_length = 2)
```

ori_BCs *List of Barcodes.*

Description

An object called ori_BCs containing specific predefined barcode sequences.

Usage

```
ori_BCs
```

Format

a character vector:

Details

ori_BCs:

plotClusterTree *Plotting a Cluster Tree*

Description

Generates a tree plot based on a herachical clustering of the complete hamming distance matrix.

Usage

```
plotClusterTree(BC_dat, tree_est = c("NJ", "UPGMA"), type = c("unrooted",
  "phylogram", "cladogram", "fan", "radial"), tipLabel = FALSE)
```

Arguments

BC_dat	a BCdat object.
tree_est	a character string, indicating the particular cluster algorithm, possible algorithms are "Neighbor-Joining" ("NJ") and "Unweighted Pair Group Method" ("UPGMA").
type	a character string, indication the graph layout style.
tipLabel	a logical value, indication the use of labeled tree leaves.

Value

a ggtree object.

Examples

```
data(BC_dat)
plotClusterTree(BC_dat, tree_est = "UPGMA", type = "unrooted", tipLabel = FALSE)
```

plotHammDistIgraph *Plotting a Hamming Distance Network*

Description

plotHammDistIgraph will create a graph-like visualisation (ripple plot) of the corresponding barcode sequences and their similarity based on the igraph package. The nodes represent the barcode sequences and their respective size reflects the corresponding read counts. Edges between nodes indicate a hamming distance between two barcodes of maximal `minHD`. If `ori_BCs` is provided the node color also refelects the hamming distance of a particular barcode to one of the initial barcodes.

Usage

```
plotHammDistIgraph(BC_dat, minHD = 1, loga = TRUE, ori_BCs,
  threeD = FALSE, complete = TRUE, col_type = "rainbow")
```

Arguments

BC_dat	a BCdat object.
minHD	an integer value representing the maximal hamming distance value for which the graph will contain edges.
loga	a logical value, indicating the use or non-use of logarithmic read count values.
ori_BCs	a vector of character strings containing the initial barcode sequences.
threeD	a logical value to chose between 2D and 3D visualisation.
complete	a logical value. If TRUE, every node will have at least one edge.
col_type	a character sting, choosing one of the available color palettes.

Value

an igraph object.

Examples

```
data(BC_dat)
plotHammDistIgraph(BC_dat, minHD = 1, loga = TRUE, ori_BCs, threeD = FALSE,
complete = TRUE, col_type = "rainbow")
```

---

plotHammDistVisNetwork

Plotting a Hamming Distance Network

Description

plotHammDistVisNetwork will create a graph-like visualisation (ripple plot) of the corresponding barcode sequences and their similarity based on the ggplot2 and the ggnetwork packages. The nodes represent the barcode sequences and their respective size reflects the corresponding read counts. Edges between nodes indicate a hamming distance between two barcodes of maximal `minHD`. If `ori_BCs` is provided the node color also refelects the hamming distance of a particular barcode to one of the initial barcodes.

Usage

```
plotHammDistVisNetwork(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL,
  complete = TRUE, col_type = "rainbow")
```

Arguments

BC_dat	a BCdat object.
minHD	an integer value representing the maximal hamming distance value for which the graph will contain edges.
loga	a logical value indicating the use or non-use of logarithmic read count values.
ori_BCs	a vector of character strings containing the initial barcode sequences.
complete	a logical value. If TRUE, every node will have at least one edge.
col_type	a character sting, choosing one of the available color palettes.

Value

a visNetwork object.

Examples

```
data(BC_dat)
plotHammDistVisNetwork(BC_dat, minHD = 1, loga = TRUE, ori_BCs = NULL,
complete = TRUE, col_type = "rainbow")
```

---

plotNucFrequency *Plotting Nucleotide Frequency*

Description

Creates a plot visualising the nucleotide frequency within the entire fastq file.

Usage

```
plotNucFrequency(source_dir, file_name)
```

Arguments

source_dir	a character string containing the path to the sequencing file.
file_name	a character string containng the name of the sequencing file.

Value

a ggplot2 object.

---

plotQualityScoreDis *Plotting Quality Score Distribution*

Description

Creates a plot of the quality values accommodated by the fastq file.

Usage

```
plotQualityScoreDis(source_dir, file_name, type)
```

Arguments

source_dir	a character string of the path to the source directory.
file_name	a character string of the file name.
type	a character string, possible values are "mean" and "median".

Value

a ggplot2 object.

Examples

```
## Not run:

source_dir <- system.file("extdata", package = "genBaRcode")

plotQualityScoreDis(source_dir, file_name = "test_data.fastq", type = "mean")


## End(Not run)
```

`plotQualityScorePerCycle`
Plotting Quality Score per Cycle

Description

Visualises the mean, median, 25

Usage

```
plotQualityScorePerCycle(source_dir, file_name)
```

Arguments

`source_dir`	a character string containing the path to the sequencing file.
`file_name`	a character string containng the name of the sequencing file.

Value

a ggplot2 object.

plotTimeSeries *Plotting Time Series Data*

Description

Uses the result of the generateTimeSeriesData function as inout and generates a visualisation of the clonal contributions over a number of given time points (similar to a stacked barplot).

Usage

```
plotTimeSeries(ov_dat, colr = NULL, tp = NULL, bar_width = 0.05,
  labs = NULL, x_label = "time", y_label = "contribution")
```

Arguments

ov_dat	a numeric matrix consisting of all time points as columns and all barcode sequences as rows and the corresponding read counts as numerical values (see function generateTimeSeriesData()).
colr	a vector of character strings identifying a certain color palette.
tp	a numeric vector containing the time points of measurement (in case of unequally distributed time points).
bar_width	a numeric value specifying the visual space between two plotted measurements.
labs	a character vector containing sample labels.
x_label	a character string providing the x-axis label.
y_label	a character string providing the y-axis label.

Value

a ggplot2 object.

Examples

```
ov_dat <- matrix(round(runif(1:100, min = 0, max = 1000)), ncol = 5)
rownames(ov_dat) <- paste("barcode", 1:20)
plotTimeSeries(ov_dat)
```

processingRawData *Data processing*

Description

Reads the corresponding fast(q) file(s), extracts the defined barcode constructs and counts them. Optionally, a Phred-Score based quality filtering will be conducted and the results will be saved within a csv file.

Usage

```
processingRawData(file_name, source_dir, results_dir, mismatch = 0,
  label = "", bc_pattern, quality_filtering = FALSE, min_score = 30,
  min_reads = 2, unix = FALSE, save_it = TRUE, cpus = 1)
```

Arguments

file_name	a character string which will serve as file name.
source_dir	a character string which contains the path to the source files.
results_dir	a character string which contains the path to the results directory.
mismatch	an positive integer value, default is 0, if greater values are provided they indicate the number of allowed mismtaches when identifying the barcode constructes.
label	a character string which serves as a label for every kind of created output file.
bc_pattern	a character string describing the barcode design, variable positions have to be marked with the letter 'N'.
quality_filtering	a logical value. If TRUE a quality filtering will be applied before extracting the barcode sequences
min_score	a positive integer value, only relevant if quality_filtering is TRUE, all fastq sequence with an average score smaller then min_score will be excluded
min_reads	positive integer value, all extracted barcode sequences with a read count smaller than min_reads will be excluded from the results
unix	under construction
save_it	a logical value. If TRUE, the raw data will be saved as a csv-file.
cpus	a positive integer identifying the number of usable CPUs.

Value

a BCdat object which includes reads, seqs, directories, masks.

Examples

```
## Not run:

bc_pattern <- "ACTNNCGANNCTTNNCGANNCTTNNGGANNCTANNACTNNCGANNCTTNNCGANNCTTNNGGANNCTANNACTNNCGANN"

source_dir <- system.file("extdata", package = "genBaRcode")

processingRawData(file_name = "test_data.fastq", source_dir, results_dir = getwd(), mismatch = 0,
label = "test", bc_pattern, quality_filtering = FALSE, min_score = 30,
min_reads = 2, unix = FALSE, save_it = TRUE)


## End(Not run)
```

---

qualityFiltering *Quality Filtering*

Description

Excludes all sequences of a given fastq file below a certain quality value.

Usage

```
qualityFiltering(file_name, source_dir, results_dir, min_score = 30)
```

Arguments

file_name	a character string containing the name of the source file.
source_dir	a character string containing the path to the source directory.
results_dir	a character string containing the path to the directory of the results.
min_score	an integer value representing the minimal average phred score a read has to achieve in order to be accepted.

Value

a ShortRead object.

Examples

```
source_dir <- system.file("extdata", package = "genBaRcode")
qualityFiltering(file_name = "test_data.fastq", source_dir, results_dir = getwd(), min_score = 30)
```

`readBCdat` *Data Input*

Description

Reads in a data table and returns a BCdat objects.

Usage

```
readBCdat(path = "./", label = "", mask = "", file_name, s = ";")
```

Arguments

`path`	a character string containing the path to a saved read count table (two columns containing read counts and barcode sequences).
`label`	a character string containing a label of the data set.
`mask`	a character string containing the barcode structure information.
`file_name`	a character string containing the name of the file to read in.
`s`	a character value, identifying the column separating char.

Value

a BCdat object.

Index

∗Topic **datasets**
BC_dat, 3
ori_BCs, 10

as.BCdat, 2

BC_dat, 3
BCdat *(*BCdat-class*)*, 3
BCdat-class, 3

checkBarcodeData, 4
createGephiFile, 4

errorCorrection, 5
extractBarcodes, 5

genBaRcode_app, 6
generateKirchenplotBCdis, 7
generateKirchenplotRCdis, 7
generateTimeSeriesData, 8
ggplotHammDistGraph, 9

hybridsIdentification, 10

ori_BCs, 10

plotClusterTree, 11
plotHammDistIgraph, 11
plotHammDistVisNetwork, 12
plotNucFrequency, 13
plotQualityScoreDis, 13
plotQualityScorePerCycle, 14
plotTimeSeries, 15
processingRawData, 16

qualityFiltering, 17

readBCdat, 18