Package 'BBCAnalyzer'

February 19, 2018

Type Package

Title BBCAnalyzer: an R/Bioconductor package for visualizing base counts

Version 1.9.0

Date 2016-01-26

Author Sarah Sandmann

Maintainer Sarah Sandmann <sarah.sandmann@uni-muenster.de>

Description BBCAnalyzer is a package for visualizing the relative or absolute number of bases, deletions and insertions at defined positions in sequence alignment data available as bam files in comparison to the reference bases. Markers for the relative base frequencies, the mean quality of the detected bases, known mutations or polymorphisms and variants called in the data may additionally be included in the plots.

License LGPL-3

Imports SummarizedExperiment, VariantAnnotation, Rsamtools, grDevices, GenomicRanges, IRanges, Biostrings

Suggests BSgenome.Hsapiens.UCSC.hg19

biocViews Sequencing, Alignment, Coverage, GeneticVariability, SNP

NeedsCompilation no

R topics documented:

`BBCAnalyzer-package` *BBCAnalyzer: an R/Bioconductor package for visualizing base counts*

Description

BBCAnalyzer is a package for visualizing the relative or absolute number of bases, deletions and insertions at defined positions in sequence alignment data available as bam files in comparison to the reference bases. Markers for the relative base frequencies, the mean quality of the detected bases, known mutations or polymorphisms and variants called in the data may additionally be included in the plots.

Details

Package:	BBCAnalyzer
Type:	Package
Title:	BBCAnalyzer: an R/Bioconductor package for visualizing base counts
Version:	1.9.0
Date:	2016-01-26
Author:	Sarah Sandmann
Maintainer:	Sarah Sandmann <sarah.sandmann@uni-muenster.de>
Description:	BBCAnalyzer is a package for visualizing the relative or absolute number of bases, deletions and in
License:	LGPL-3
Imports:	SummarizedExperiment, VariantAnnotation, Rsamtools, grDevices, GenomicRanges, IRanges, Bios
Suggests:	BSgenome.Hsapiens.UCSC.hg19
biocViews:	Sequencing, Alignment, Coverage, GeneticVariability, SNP
NeedsCompilation:	no

In the use case of medical diagnostics, a tool performing detailed analyses of those locations where mutations may be expected – but not always called – appears to be most useful. Low allele frequency and bad base quality do often explain a lacking call. Yet, this information is not included in a VCF-file and difficult to obtain from other existing tool. Furthermore, with regards to the comparison of different sequencing techniques, it seems helpful to have a tool for visualizing the background at a selection of locations where e.g. one technique calls a variant but another technique does not.

BBCAnalyzer (Bases By CIGAR Analyzer) is a tool for visualizing the number of counted bases, deletions and insertions at any given position in any genome in comparison to the reference bases. Relative frequencies, base qualities, known mutations or polymorphisms and called variants may be included into the plots as well.

Index of help topics:

`BBCAnalyzer`	`BBCAnalyzer: an R/Bioconductor package for visualizing base counts`
`analyzeBases`	`Analyze the bases at all previously defined positions`
`analyzeBasesPlotOnly`	`Plots the number of reads at all previously defined positions`

The package contains two functions - `analyzeBases` and `analyzeBasesPlotOnly`. The major use

of BBCAnalyzer is documented in the description of the function `analyzeBases`. The function `analyzeBasesPlotOnly` serves as an extension.

Author(s)

Sarah Sandmann

Maintainer: Sarah Sandmann <sarah.sandmann@uni-muenster.de>

References

More information on the bam format can be found at: http://samtools.github.io/hts-specs/SAMv1.pdf

See Also

`analyzeBases`, `analyzeBasesPlotOnly`

Examples

```
library("BSgenome.Hsapiens.UCSC.hg19")
ref_genome<-BSgenome.Hsapiens.UCSC.hg19

output<-analyzeBases(sample_names=system.file("extdata","SampleNames_small.txt",package="BBCAnalyzer"),
                     bam_input=system.file("extdata",package="BBCAnalyzer"),
                 target_regions=system.file("extdata","targetRegions_small.txt",package="BBCAnalyzer"),
                     vcf_input="",
                     output=system.file("extdata",package="BBCAnalyzer"),
                     output_pictures=system.file("extdata",package="BBCAnalyzer"),
                     known_file="",
                     genome=ref_genome,
                     MQ_threshold=60,
                     BQ_threshold=50,
                     frequency_threshold=0.01,
                     qual_lower_bound=58,
                     qual_upper_bound=63,
                     marks=c(0.01),
                     relative=TRUE,
                     per_sample=TRUE)
```

`analyzeBases` *Analyze the bases at all previously defined positions*

Description

BBCAnalyzer performs an analysis of the bases, deletions and insertions at defined positions in sequence alignment data and visualizes the results. The function `analyzeBases` performs the whole process of analyzing the data and plotting the results.

Usage

```
analyzeBases(sample_names, bam_input, target_regions, vcf_input, output,
             output_pictures, known_file, genome, MQ_threshold, BQ_threshold,
             frequency_threshold, qual_lower_bound, qual_upper_bound,
             marks, relative, per_sample)
```

Arguments

sample_names	The txt-file containing the names of the samples to be analyzed.
bam_input	The folder containing the alignment data in bam- and bai format or BamFileList.
target_regions	The txt-file containing the target regions to be analyzed.
vcf_input	The folder containing the vcf files or a VcfFileList of the samples to be analyzed or a multiple-sample vcf file. The argument may be left blank.
output	The folder to write the output into.
output_pictures	The folder to write the plots into. The argument may be left blank. In this case, the plots are returned to the workspace.
known_file	The name of a tabix file containing e.g. known polymorphisms or mutations (dbSNP). The argument may be left blank.
genome	A BSgenome object defining reference genome that shall be used for comparison (e.g. BSgenome.Hsapiens.UCSC.hg19).
MQ_threshold	A PHRED-scaled value to be used as a mapping quality threshold. All reads with a mapping quality below this threshold are excluded from analysis. Every base in an excluded read gets marked in an output file.
BQ_threshold	A PHRED-scaled value to be used as a base quality threshold. All bases with a base quality below this threshold are excluded from analysis. Every excluded base gets marked in an output file. The number of excluded bases per position gets counted.
frequency_threshold	A frequency to be used as a threshold for variants to be reported ([0,1]).
qual_lower_bound	The lower bound for the mean quality that shall be color-coded in the plots. All bases with a mean quality below qual_lower_bound are colored with the lightest color defined for the corresponding base. If the bases shall not be color-coded according to their mean quality, qual_lower_bound has to be identical compared to qual_upper_bound.
qual_upper_bound	The upper bound for the mean quality that shall be color-coded in the plots. All bases with a mean quality above qual_upper_bound are colored with the darkest color defined for the corresponding base. If the bases shall not be color-coded according to their mean quality, qual_upper_bound has to be identical compared to qual_lower_bound.
marks	A vector of floats [0,1] defining the levels at which marks shall be drawn in the plot.
relative	A boolean defining whether the relative (true) or the absolute (false) number of reads shall be plotted.
per_sample	A boolean defining whether one plot per sample (true) or one plot per target (false) shall be created.

Details

About the input:

Names of the samples to be analyzed have to be provided by a file (`sample_names`). There has to be one sample name per line without the ".bam"-suffix.

The bam- and the corresponding bai files of the samples to be analyzed have to be provided in a folder (`bam_input`). The names of the files have to match the sample names provided by `sample_names`.

The target regions have to be provided by a file (`target_regions`). The file may either contain regions (chromosome, tab, first base of a region, tab, last base of a region) or positions (chromosome, tab, position). A mixture of both is not supported. Yet, a region may cover only one base, i.e. the first and last base of a region may be identical.

If vcf files shall be considered, the corresponding files of the samples to be analyzed have to be provided in a folder (`vcf_input`). There has to be one file per sample. The names of the files have to match the sample names provided by `sample_names`. If a vcf file is not available for one or more samples, empty vcf files may be used instead.

About the analysis of the data:

Determine target: If the `target_regions` file contains regions to be analyzed, the different positions covered by the regions are determined. If the file already contains single positions, the program directly proceeds with the next step. It is not necessary that the regions or positions are ordered. If a known insert is supposed to be analyzed, the position of the base succeeding the insert has to be given.

Analyze reads: The reads at every targeted position get analyzed. By the help of the CIGAR string the bases, deletions and insertions are determined. The output is saved as [Sample].bases.txt. For every base - also the inserted ones - the base quality is determined. The output is saved as [Sample].quality.txt. Reads with a mapping quality below `MQ_threshold` get excluded from the analysis. Instead of a base, "MQ" is noted in the base file. Instead of a quality value, "-2" is noted in the quality file. The function copes with uncovered positions ("NotCovered" in the base- and the quality file) and insertions >1bp (repeated analysis of the position).

Analyze frequency: The number of detected bases, deletions and insertions at every position is summed up. Additionally, the mean quality of the detected bases - including the insertions and for the inserted bases only - is calculated. Bases with a quality below `BQ_threshold` are excluded and counted separately. The output is saved as [Sample].frequency.txt. If the analysis shall consider vcf files as well (`vcf_input` not blank), the alternate alleles and the genotypes - as far as they are available for the positions analyzed - are written out as well. Furthermore, for every called variant it is noted whether it is an insert. The function copes with up to two different alternate alleles per position.

Report variants: The ratios of the detected bases, deletions and insertions (additionally) at every position are determined. According to the determined ratios, up to six different calls get reported. If `frequency_threshold` is set, minor variants with ratios below this threshold do not get reported. The output is saved as [Sample].calling.txt. The function copes with insertions >1bp even if the position at which an insert is detected is not covered by all samples being analyzed. If the analysis shall consider vcf files as well (`vcf_input` not blank), the call - taking the reference allele, the alternate allele(s) and the genotype into account - is written out as well. For every called variant it is noted whether it is an insert and whether the genotype is heterozygous.

About plotting the results:

The absolute number of the detected bases, deletions and insertions for each sample at each targeted position is plotted if `relative==FALSE`. Otherwise the relative frequencies of the detected bases, deletions and insertions for each sample at each targeted position get plotted. The bars are colored

according to the base (adenine: green; cytosine: blue; guanine: yellow; thymine: red; deletion: black; insertion: lilac edge) and their mean quality (high mean quality: dark color; low mean quality: light color). The reference bases (using the defined package `ref_genome`) are plotted on the negative y-axis below each position. If a file containing known variants or mutations is provided (`known_file`), more than one reference base is plotted at the corresponding position. For each position to be analyzed, lines are drawn at the heights of the ratios defined in `marks`. Every targeted position is labelled according to the chromosome and the position. The function copes with different inserted bases at one position (stacked bars) and insertions $>$1bp, even if these are not covered by all samples. If the analysis shall consider vcf files as well (`vcf_input` not blank), the expected number of detected bases, deletions and insertions – according to the vcf file – is added to the plot using dashed lines.

Plot per sample: One barplot per sample is created. The output is saved as [Sample].png.

Plot per target: One barplot per targeted position is created. The output is saved as chr[number];[position].png.

Value

A list is returned:

`bases`	A list containing the bases for all samples and positions being analyzed (identical compared to [Sample].bases.txt).
`quality`	A list containing the corresponding qualities for all samples and positions being analyzed (identical compared to [Sample].quality.txt).
`frequency`	A list containing the summed up number of bases for all samples and positions being analyzed (identical compared to [Sample].frequency.txt).
`calling`	A list containing the relative frequencies and the potential calls for all samples and positions being analyzed (identical compared to [Sample].calling.txt).

Author(s)

Sarah Sandmann <sarah.sandmann@uni-muenster.de>

References

More information on the bam format can be found at: http://samtools.github.io/hts-specs/SAMv1.pdf

db SNP – Short Genetic variations: http://www.ncbi.nlm.nih.gov/SNP/

See Also

BBCAnalyzer, analyzeBasesPlotOnly

Examples

```
library("BSgenome.Hsapiens.UCSC.hg19")
ref_genome<-BSgenome.Hsapiens.UCSC.hg19

output<-analyzeBases(sample_names=system.file("extdata","SampleNames_small.txt",package="BBCAnalyzer"),
                     bam_input=system.file("extdata",package="BBCAnalyzer"),
                 target_regions=system.file("extdata","targetRegions_small.txt",package="BBCAnalyzer"),
                     vcf_input="",
                     output=system.file("extdata",package="BBCAnalyzer"),
                     output_pictures=system.file("extdata",package="BBCAnalyzer"),
                     known_file="",
```

```
                    genome=ref_genome,
                    MQ_threshold=60,
                    BQ_threshold=50,
                    frequency_threshold=0.01,
                    qual_lower_bound=58,
                    qual_upper_bound=63,
                    marks=c(0.01),
                    relative=TRUE,
                    per_sample=TRUE)
```

`analyzeBasesPlotOnly` *Plots the number of reads at all previously defined positions*

Description

To allow for a quick change in the way the analysis of the bases, deletions and insertions at defined positions in sequence alignment data is visualized, the function `analyzeBasesPlotOnly` may be used. It solely performs the last step of the whole analysis pipeline – the plotting of the results.

Usage

```
analyzeBasesPlotOnly(sample_names, vcf_input, output, known_file,
                     output_list, qual_lower_bound, qual_upper_bound,
                     marks, relative, per_sample)
```

Arguments

`sample_names`	The file containing the names of the samples to be analyzed.
`vcf_input`	The folder containing the vcf files or a VcfFileList of the samples to be analyzed or a multiple-sample vcf file. The argument may be left blank.
`output`	The folder to write the output (plots) into. The argument may be left blank. In this case, the plots are returned to the workspace.
`known_file`	The name of a tabix file containing e.g. known polymorphisms or mutations (dbSNP). The argument may be left blank.
`output_list`	The name of the list that is returned by the function `analyzeBases`.
`qual_lower_bound`	The lower bound for the mean quality that shall be color-coded in the plots. All bases with a mean quality below `qual_lower_bound` are colored with the lightest color defined for the corresponding base. If the bases shall not be color-coded according to their mean quality, `qual_lower_bound` has to be identical compared to `qual_upper_bound`.
`qual_upper_bound`	The upper bound for the mean quality that shall be color-coded in the plots. All bases with a mean quality above `qual_upper_bound` are colored with the darkest color defined for the corresponding base. If the bases shall not be color-coded according to their mean quality, `qual_upper_bound` has to be identical compared to `qual_lower_bound`.

marks	A vector of floats [0,1] defining the levels at which marks shall be drawn in the plot.
relative	A boolean defining whether the relative (true) or the absolute (false) number of reads shall be plotted.
per_sample	A boolean defining whether one plot per sample (true) or one plot per target (false) shall be created.

Details

About the input:

Names of the samples to be analyzed have to be provided by a file (sample_names). There has to be one sample name per line without the ".bam"-suffix.

If vcf files shall be considered, the corresponding files of the samples to be analyzed have to be provided in a folder (vcf_input). There has to be one file per sample. The names of the files have to match the sample names provided by sample_names. If a vcf file is not available for one or more samples, empty vcf files may be used instead.

The value that gets returned by the function analyzeBases has to be provided (output_list). Otherwise, analyzeBasesPlotOnly will not be able to use the previously generated output.

About plotting the results:

The absolute number of the detected bases, deletions and insertions for each sample at each targeted position is plotted if relative==FALSE. Otherwise the relative frequencies of the detected bases, deletions and insertions for each sample at each targeted position get plotted. The bars are colored according to the base (adenine: green; cytosine: blue; guanine: yellow; thymine: red; deletion: black; insertion: lilac edge) and their mean quality (high mean quality: dark color; low mean quality: light color). The reference bases (using the defined package ref_genome) are plotted on the negative y-axis below each position. If a file containing known variants or mutations is provided (known_file), more than one reference base is plotted at the corresponding position. For each position to be analyzed, lines are drawn at the heights of the ratios defined in marks. Every targeted position is labelled according to the chromosome and the position. The function copes with different inserted bases at one position (stacked bars) and insertions $>$1bp, even if these are not covered by all samples. If the analysis shall consider vcf files as well (vcf_input not blank), the expected number of detected bases, deletions and insertions – according to the vcf file – is added to the plot using dashed lines.

Plot per sample: One barplot per sample is created. The output is saved as [Sample].png.

Plot per target: One barplot per targeted position is created. The output is saved as chr[number];[position].png.

Value

No value is returned.

Author(s)

Sarah Sandmann <sarah.sandmann@uni-muenster.de>

References

db SNP – Short Genetic variations: http://www.ncbi.nlm.nih.gov/SNP/

See Also

BBCAnalyzer, analyzeBases

Examples

```
library("BSgenome.Hsapiens.UCSC.hg19")
ref_genome<-BSgenome.Hsapiens.UCSC.hg19

output<-analyzeBases(sample_names=system.file("extdata","SampleNames_small.txt",package="BBCAnalyzer"),
                     bam_input=system.file("extdata",package="BBCAnalyzer"),
                 target_regions=system.file("extdata","targetRegions_small.txt",package="BBCAnalyzer"),
                     vcf_input="",
                     output=system.file("extdata",package="BBCAnalyzer"),
                     output_pictures=system.file("extdata",package="BBCAnalyzer"),
                     known_file="",
                     genome=ref_genome,
                     MQ_threshold=60,
                     BQ_threshold=50,
                     frequency_threshold=0.01,
                     qual_lower_bound=58,
                     qual_upper_bound=63,
                     marks=c(0.01),
                     relative=TRUE,
                     per_sample=TRUE)

analyzeBasesPlotOnly(sample_names=system.file("extdata","SampleNames_small.txt",package="BBCAnalyzer"),
                     vcf_input="",
                     output=system.file("extdata",package="BBCAnalyzer"),
                     known_file="",
                     output_list=output,
                     qual_lower_bound=58,
                     qual_upper_bound=63,
                     marks=c(0.25,0.5,0.75,1),
                     relative=FALSE,
                     per_sample=TRUE)
```

Index