Package 'NIPTeR'

March 9, 2016

Type Package

Encoding UTF-8

Title Fast and Accurate Trisomy Prediction in Non-Invasive Prenatal Testing

Version 1.0.2

Date 2016-03-09

Author Dirk de Weerd, Lennart Johansson

Maintainer Lennart Johansson <l.johansson@umcg.nl>

Description Fast and Accurate Trisomy Prediction in Non-Invasive Prenatal Testing.

Imports stats, Rsamtools, sets, S4Vectors

Depends R (>= 3.1.0),

License GNU Lesser General Public License

Suggests knitr, pander

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-09 18:55:22

R topics documented:

`add_samples_controlgroup` *Add a sample to an existing control group*

Description

This functions adds NIPTSample objects to an existing control group and returns a new NIPTControlGroup object.

Usage

```
add_samples_controlgroup(nipt_control_group, samples_to_add)
```

Arguments

`nipt_control_group`	The NIPTControlGroup to add the samples to
`samples_to_add`	A list with sample(s) to add. This always needs to be a list

Value

NIPTControlGroup object

Examples

```
## Not run:
##First bin the new sample
new_binned_sample <- bin_bam_sample(bam_filepath = "/path/to/file.bam",
                                    separate_strands = T)

##Then add the sample to the control group
new_control_group <- add_samples_controlgroup(nipt_control_group = my_control_group,
                                              samples_to_add = new_binned_sample)

## End(Not run)
```

`as_control_group` *Convert list of nipt samples to nipt control group*

Description

Convert list of nipt samples to nipt control group

Usage

```
as_control_group(nipt_samples, control_group_type = generic_control_group)
```

Arguments

`nipt_samples`	List of nipt_sample objects to be combined to a control group
`control_group_type`	Control group type, either 'generic control group' or 'fitted to sample'. Leave this argument blank

Details

This function returns an S3 object of class nipt_control_group. It is a list with 3 items:

- List **Samples** nipt_sample objects in the control group
- Character **Correction_status** Correction_status(es) in the control group
- Character **Samplenames** The sample names of samples present in the control group

Read count strategy should be uniform in all samples in a control group object; meaning samples where forward and reverse reads are counted separately cannot be in the same control group object as samples where forward and reverse reads are counted together.

A control group object with duplicate samples or samples with different correction statusses is possible but not recommended and will generate a warning message.

Value

NIPTControlGroup object

Examples

```
## Not run:
##Retrieve filenames
bam_filepaths <- list.files(path = "/Path/to/bamfiles/", pattern = ".bam", full.names = T)
##Load files and convert to control group
control_group  <- as_control_group(nipt_samples = lapply(X = bam_filepaths, bin_bam_sample,
                                                         do_sort = F, separate_strands = FALSE))
##Save control group for later
saveRDS(object = control_group, file = "/Path/to/directory/control_group.rds")

## End(Not run)
```

`bin_bam_sample` *Load and bin BAM file*

Description

Load a BAM file and count reads in bins of size 50.000 base pairs

Usage

```
bin_bam_sample(bam_filepath, do_sort = FALSE, separate_strands = FALSE,
  custom_name = NULL)
```

Arguments

`bam_filepath`	Character The location and filename on the file system where the bam file is stored
`do_sort`	Boolean Sort the bam file? If the bam is unsorted set to true, but the use of pre-sorted bam files is recommended.
`separate_strands`	Boolean If set to true, reads from forward and reverse strands are counted and stored separately. This option should be used if you are planning on using regression, since this doubles the number of predictors (F+R) and distributes predictive power more equally over prediction sets since F and R strand from the same chromosome cannot be both in one prediction set.
`custom_name`	String The name of sample. Default samplename is the filename of the bam file without the .bam suffix and filepath prefix.

Details

This function returns an object of class NIPTSample, the main 'currency' of this package. It is a list with 5 items:

- List **autosomal_chromosome_reads** Autosomal reads are stored in a matrix where the columns are the bins and rows (22) represent the autosomal chromosomes. The length of this list is either 1 or 2, depending if the forward and reverse reads are counted separately.
- Character **correction_status_autosomal_chromosomes** The correction status of the autosomal reads. The status can either be *Uncorrected* or *GC Corrected* and/or *Chi Corrected*
- List **sex_chromosome_reads** Sex chromosome reads are stored in a similar matrix(es) as the autosomal chromosome reads, now with 2 (X and Y) rows.
- Character **correction_status_autosomal_chromosomes** The status can either be *Uncorrected* or *GC Corrected* and/or *Chi Corrected*.
- Character **sample_name** Sample name

Value

Object NIPTSample

Examples

```
## Not run:
##To process a single sample
binned_sample <- bin_bam_sample(bam_filepath = "/path/to/file.bam",
                                separate_strands = T)

##To create a control group out of a set of bam files
bam_filepaths <- list.files(path = "/Path/to/bamfiles/",
                            pattern = ".bam", full.names = T)

control_group  <- as_control_group(nipt_samples = lapply(X = bam_filepaths,
                                    bin_bam_sample, do_sort = F,
                                    separate_strands = T))

## End(Not run)
```

`calculate_ncv_score` *Use an NCV template to calculate a NCV score for sample of interest*

Description

Use an NCV template to calculate a NCV score for sample of interest

Usage

```
calculate_ncv_score(nipt_sample, ncv_template)
```

Arguments

`nipt_sample`	nipt_sample object of interest
`ncv_template`	ncv_template object, result from `prepare_ncv`

Details

`prepare_ncv`

Value

ncv_result object

References

Sehnert et al.

Examples

```
## Not run:
##Use NCVTemplate to get NCV scores for the sample of interest
ncv_score_13 <- calculate_ncv_score(nipt_sample = sample_of_interest,
                                    ncv_template = new_ncv_template_13)

## End(Not run)
```

`calculate_z_score` *Calculate 'standard' Z-score*

Description

Calculate 'standard' Z-score

Usage

```
calculate_z_score(nipt_sample, nipt_control_group, chromo_focus)
```

Arguments

`nipt_sample`	The NIPTSample object that is the focus of the analysis
`nipt_control_group`	The NIPTControlGroup object used in the analysis
`chromo_focus`	The chromosome of interest. Most commonly chromosome 13, 18 or 21. However, every autosomal chromosome can be predicted

Details

In the Z-score approach, introduced by Chiu et al in 2008, the chromosomal fraction of interest of a sample is compared to the chromosomal fractions of interest of the reference samples, the 'NIPTControlGroup' object. The output of the function is an object of class 'ZscoreResult'. It is a named list containing seven fields:

- numeric **sample_Zscore** The Z score for the sample of interest for the sample of interest
- named num **control_group_statistics** Named num of length 3, the first field being the mean (name mean), the second field is the standard deviation (name SD) and the third field is the P value of the Shapiro-Wilk test (name Shapiro_P_value)
- matrix **control_group_Zscores** containing the Z scores of the chromosome of interest for all used control samples
- integer **focus_chromosome** The chromosome of interest. Most commonly chromosome 13, 18 or 21. However, every autosomal chromosome can be predicted
- string **control_group_sample_names** The sample names of all control group samples used in the analysis
- string **correction_status** The correction status of the control group
- string **sample_name** The sample_name of the sample of interest

Value

ZscoreResult object

Examples

```
## Not run:
z_score_result_13 <- calculate_z_score(nipt_sample = sample_of_interest,
                                       nipt_control_group = control_group,
                                       chromo_focus = 13)

## End(Not run)
```

`chi_correct` *Performs chi-square based variation reduction*

Description

Performs chi-square based variation reduction

Usage

```
chi_correct(nipt_sample, nipt_control_group, chi_cutoff = 3.5,
  include_XY = F)
```

Arguments

`nipt_sample`	The NIPTSample object that is the focus of the analysis
`nipt_control_group`	The NIPTControlGroup object used in the analysis
`chi_cutoff`	The Z-score cutoff. If a bin has a Z-score above this threshold, it will be corrected
`include_XY`	Also apply correction to X and Y chromosomes?

Details

The chi-squared based variation reduction identifies overdispersed bins within the control group and corrects these bins in both the sample of interest and the control group. The function takes in a 'NIPTSample' and a 'NIPTControlGroup' object, both to be corrected. For every corresponding bin in the control group a chi-squared score is calculated and this total score is converted to a normal distribution. Corresponding bins with a normalized score above _chi_cutoff_ (default 3.5) are corrected by dividing the number of reads by the total chi-squared score divided by degrees of freedom

Value

Named list of length 2. The corrected nipt_sample is in index 1 and the corrected control group in index 2 to extract the corrected sample use `$sample` or `[[1]]`. To extract the control group from the list use `$control_group` or `[[2]]`

Examples

```
## Not run:
##Apply chi-squared based variation reduction method
chi_corrected_data <- chicorrect(nipt_sample = gc_LOESS_corrected_sample,
                                 nipt_control_group = subset_loess_corrected_control_group)
##Extract sample and control group
loess_chi_corrected_sample <- chi_corrected_data$sample
subset_loess_chi_corrected_control_group <- chi_corrected_data$control_group

## End(Not run)
```

---

chrfractions *Calculate chromosomal fraction*

Description

Calculate chromosomal fraction

Usage

```
chrfractions(nipt_sample)
```

Arguments

`nipt_sample`	NIPTSample to retrieve chromosomal fraction for

---

diagnose_control_group *Diagnose control group*

Description

Compute a regular Z-score for every chromosome of every sample in a NIPTControlGroup object

Usage

```
diagnose_control_group(nipt_control_group)
```

Arguments

`nipt_control_group`
: The NIPTControlGroup object to diagnose

Details

This function computes a regular Z-score for every chromosome of every sample in a NIPTControlGroup object. It returns a named list with diagnostics information.

The function returns a named list with 3 fields:

- **Z_scores** A matrix containing Z-scores for every sample and every chromosome
- **abberant_scores** Dataframe with samplename and chromosome of Z-scores outside -3 3 range
- **control_group_statistics** Matrix with mean, standard deviation and P value of Shapiro-Wilk test

Value

named list

Examples

```
## Not run:
diagnose_control_group(nipt_control_group = control_group)

## End(Not run)
```

`gc_correct` *Perform a GC bias correction on nipt sample*

Description

LOESS based GC bias correction algorithm described by Chen et al (2011)

Usage

```
gc_correct(nipt_object, method = "LOESS", include_XY = F, span = 0.75,
  ref_genome = "hg37")
```

Arguments

`nipt_object`	The object that will be corrected. This can either be a 'NIPTSample' or a 'NIPTControlGroup' object
`method`	To select the LOESS based method use *"LOESS"*, to select the bin weights based method use *"bin"*.
`include_XY`	Also apply correction to X and Y chromosomes?

span	The span for the LOESS fit. Only applicable when LOESS method is used.
ref_genome	The reference genome used. Either *"hg37"* or *"hg38"* default = *"hg37"*

Details

GC content bias is the correlation between the number of reads mapped to a specific genomic region and the GC content of this region. In NIPTeR, two GC bias correction algorithms have been implemented, the LOESS based method introduced by Chen et al. (2011) and the bin weight based method described by Fan and Quake (2010).

Value

Depending on the input object either a NIPTSample or a NIPTControlGroup object

Examples

```
## Not run:
##Correct NIPTSample object using LOESS method
loess_corrected_sample <- gc_correct(nipt_object = sample_of_interest, method = "LOESS",
                                     include_XY = F, span = 0.75)
##Correct NIPTControlGroup object using bin method
gc_bin_corrected_control_group <- gc_correct(nipt_object = control_group, method = "bin",
                                             include_XY = T)

## End(Not run)
```

---

getcontrolchromosomes *Get control chromosomes names*

Description

Get control chromosomes names

Usage

```
getcontrolchromosomes(nipt_sample, control_chromosomes = control_chromosomes)
```

Arguments

nipt_sample	A sample to check wether combined or separated strands are used
control_chromosomes	Vector with control chromosomes

`getfractionscontrolgroup` *Get all chromosomal fractions of a control group*

Description

Get all chromosomal fractions of a control group

Usage

```
getfractionscontrolgroup(nipt_control_group)
```

Arguments

`nipt_control_group`	The NIPTControlGroup to retrieve the chromosomal fraction of every autosome for

`getreadscontrolgroup` *Get reads per chromosome per control group sample*

Description

Get reads per chromosome per control group sample

Usage

```
getreadscontrolgroup(nipt_control_group)
```

Arguments

`nipt_control_group`	The control group to retrieve reads for

`match_control_group` *Best matching control group by least sum of squares*

Description

The matchcontrolgroup function determines how well an NIPTSample fits within the NIPTControlGroup

Usage

```
match_control_group(nipt_sample, nipt_control_group, mode, n_of_samples,
  include_chromosomes = NULL, exclude_chromosomes = NULL)
```

Arguments

`nipt_sample`	The NIPTSample object that is the focus of the analysis
`nipt_control_group`	The NIPTControlGroup object used in the analysis
`mode`	The function mode. This can either be *"subset"* or *"report"*. Mode *"subset"* means the return value will be a new 'NIPTControlGroup' object containing *n* samples. When mode *"report"* is used the output is a matrix containing the sum of squares score of the differences between the chromosomal fractions of the sample and the control for every control sample, sorted in increasing score.
`n_of_samples`	The length of the resulting NIPTControlGroup. Only applicable if mode *"subset"* is used.
`include_chromosomes`	integer. Include potential trisomic chromosomes into the comparison? Default = NULL, meaning chromosomes 13, 18 and 21 are not included
`exclude_chromosomes`	integer.Exclude other autosomal chromosomes besides chromosomes 13, 18 and 21? Default = NULL

Details

The 'matchcontrolgroup' function determines how well an NIPTSample fits within the NIPTControlGroup and, if needed, makes a subset 'NIPTControlGroup' of length *n*.

Value

The output for mode *subset* is a new 'NIPTControlGroup' composed of _n_ samples. The output for mode *report* is a matrix with a single column containing the sum of squares in ascending order.

Examples

```
## Not run:
##Mode report
scores_control_group <- matchcontrolgroup(nipt_sample = sample_of_interest,
                                          nipt_control_group = control_group,
                                        mode = "report", include_chromosomes = c(13,18))

##Mode subset
subset_control_group <- matchcontrolgroup(nipt_sample = sample_of_interest,
                                          nipt_control_group = control_group,
                                          mode = "subset", n_of_samples = 50)

## End(Not run)
```

`perform_regression` *Regression based Z score*

Description

Make multiple models using linear regression and calculate Z-score

Usage

```
perform_regression(nipt_sample, nipt_control_group, chromo_focus,
  n_models = 4, n_predictors = 4, exclude_chromosomes = NULL,
  include_chromosomes = NULL, use_test_train_set = T,
  size_of_train_set = 0.6, overdispersion_rate = 1.15,
  force_practical_cv = F)
```

Arguments

`nipt_sample`	The NIPTSample object that is the focus of the analysis
`nipt_control_group`	The NIPTControlGroup object used in the analysis
`chromo_focus`	The chromosome of interest. Most commonly chromosome 13, 18 or 21. However, every autosomal chromosome can be predicted
`n_models`	Integer Number of linear models to be made. Default setting is 4 models
`n_predictors`	Integer The number of predictors each model contains. Default is 4
`exclude_chromosomes`	integer. Exclude which autosomal chromosomes as potential predictors? Default potential trisomic chromosomes 13, 18 and 21 are exluded.
`include_chromosomes`	integer. Include potential trisomic chromosomes? Options are: chromosomes 13, 18 and 21

`use_test_train_set`
: Use a test and train set to build the models? Default is TRUE

`size_of_train_set`
: The size of the train set expressed in a decimal. Default is 0.6 (60 of the control samples)

`overdispersion_rate`
: The standard error of the mean is multiplied by this factor

`force_practical_cv`
: Boolean, Ignore the theoretical CV and always use the practical CV?

Details

The regression based Z-score builds *n* models with *m* predictors using stepwise regression with forward selection. The models are used to predict the chromosomal fraction of interest, for the sample and for the control group. The observed fractions are then divided by the expected fraction, and Z-scores are calculated over these ratios. The Z-score is calculated by subtracting one from the ratio of the sample and dividing this result by the coefficient of variation. The coefficient of variation (CV) can either be the Practical or Theoretical CV. The Theoretical CV is the standard error multiplied by the overdispersion. Theoretically, the CV cannot be lower than the standard error of the mean. If it is case the CV is lower than Theoretical CV, then the Theoretical CV is used.

The output of this function is an object of type RegressionResult, a named list containing:

- **prediction_statistics** A dataframe with 7 rows and a column for every model. The rows are:
  - **Z_score_sample** The regression based Z score for the model
  - **CV** The coefficient of varation for the model
  - **cv_types** The CV type used to calculate the regression based Z score for the model. Either *Practical_CV* or *Theoretical_CV*
  - **P_value_shapiro** The P value of the Shaipro-Wilk test for normality of the control group regression based Z scores for the model
  - **Predictor_chromosomes** The predictor chromosomes used in the model
  - **Mean_test_set** The mean of the test set. Note that for calculating the regression based Z scores the mean is replaced by one. The mean, however, can be seen as a quality metric for the model
  - **CV_train_set** The CV of the train set. The difference between this CV and the CV of the test can be used as a measure to quantify overfit
- **control_group_Zscores** A matrix containing the regression based Z-scores for the control sample
- **focus_chromosome** he chromosome of interest. Most commonly chromosome 13, 18 or 21. However, every autosomal chromosome can be predicted
- **correction_status** The correction status of the control group autosomes
- **control_group_sample_names** The sample names of the test set group
- **models** List of the summary.lm output for every model
- **potential_predictors** The total pool of chromosomes where the predictors are selected from
- **all_control_group_Z_scores** Z-scores for every sample using theoretical and practical VCs
- **additional_statistics** Statistics for both the practical and theoretical CVs for every prediction set

Value

RegressionResult object

Examples

```
## Not run:
regression_score_21 <- perform_regression(nipt_sample = sample_of_interest,
                       nipt_control_group = control_group, chromo_focus = 21)

## End(Not run)
```

`prepare_ncv` *Prepare NCV calculation*

Description

Determine the best NCV chromosomes, calculate NCV scores and asses normal distribution control group using Shapiro-Wilk test

Usage

```
prepare_ncv(nipt_control_group, chr_focus, max_elements,
  exclude_chromosomes = NULL, include_chromosomes = NULL,
  use_test_train_set = T, size_of_train_set = 0.6)
```

Arguments

`nipt_control_group`
: The NIPTControlGroup object used in the analysis

`chr_focus`
: Integer.The chromosome of interest. Most commonly chromosome 13, 18 or 21. However, every autosomal chromosome can be predicted

`max_elements`
: Integer, The maximum number of denominator chromosomes.

`exclude_chromosomes`
: Integer. Exclude which autosomal chromosomes as potential predictors? Default potential trisomic chromosomes 13, 18 and 21 are exluded.

`include_chromosomes`
: Integer. Which potential trisomic chromosomes (13,18 and 21) to include?

`use_test_train_set`
: Boolean. Use a test and train set?

`size_of_train_set`
: Double The size of the train set expressed in a decimal. Default is 0.6 (60% of the control group samples)

Details

chromosomes to calculate the chromosomal fractions. The 'best' subset is the set which yields the lowest coefficient of variation for the chromosomal fractions of the chromosome of interest in the control group. Because a brute force approach is used to determine the best subset, which can be computationally intensive,this method is divided into two functions, prepare_ncv and calculate_ncv. prepare_ncv returns a template object (NCVTemplate) for a given chromosome of interest and the control group used. This template can be used for any number of analyses. If the control group or chromosome of interest changes, a new template must be made.

The ncv_template object is a list containing:

- Character **denominators** The set of denominator chromosomes
- Character **focus_chromosome**The chromosome of interest used for this 'NCVTemplate' object
- Character **nipt_sample_names** The sample names of the test set samples
- Character **correction_status** The correction status(es) of the control group samples
- Data.frame **control_group_Z_scores** The NCV scores for the test set samples
- Character **potential_denominators** The total pool of denominators the best denominators are selected from
- Numeric **control_group_statistics** Named num of length 3, the first field being the mean (name mean), the second field is the standard deviation (name SD) and the third field is the P value of the Shapiro-Wilk test (name Shapiro_P_value)

If a Test and Train set is used the ncv_template object also includes:

- Character **sample_names_train_set** The sample name where the model is trained on
- Numeric **train_set_statistics** Mean, SD and Shapiro-Wilk test P value of the Z scores of the train set
- Data.frame **train_set_Zscores** The Z scores of the train set

Value

ncv template object

References

Sehnert et al.

Examples

```
## Not run:
##Create NCVTemplates for chromosome 13 with max 9 denominators and default settings, so:
##All autosomals chromosomes are potential predictors,
##except the potential trisomic chromosomes 13, 18 and 21
new_ncv_template_13 <- prepare_ncv(nipt_control_group = control_group,
                                   chr_focus = 13, max_elements = 9)

## End(Not run)
```

`remove_duplicates_controlgroup` *Remove duplicate samples from control group*

Description

Removes all duplicate samples in control group by samplename.

Usage

```
remove_duplicates_controlgroup(nipt_control_group)
```

Arguments

`nipt_control_group`	NIPTControlGroup object

Details

This functions removes duplicate samples from the control group based on name. It returns a new NIPTControlGroup object.

Value

NIPTControlGroup object

Examples

```
## Not run:
new_control_group <- remove_duplicates_controlgroup(nipt_control_group = old_control_group)

## End(Not run)
```

`remove_sample_controlgroup` *Remove a sample by samplename from control group*

Description

Remove a sample by samplename from control group

Usage

```
remove_sample_controlgroup(samplename, nipt_control_group)
```

Arguments

samplename	Regular expression string. All matching samplenames are removed from the control group
nipt_control_group	NIPTControlGroup object to remove samples from

Details

This function removes a sample from the 'NIPTControlGroup' object by name. Note that this function uses a regular expression, and if more sample_names satisfy the regular expression, they will also be removed. It returns a new NIPTControlGroup object.

Value

NIPTControlGroup object

Examples

```
## Not run:
new_control_group <- remove_sample_controlgroup(samplename = unwanted_sample,
                                                nipt_control_group = old_control_group)

## End(Not run)
```

---

retrieve_fractions_of_interest

Retrieve the chromosomal fractions of a chromosome of interest

---

Description

Retrieve the chromosomal fractions of a chromosome of interest

Usage

```
retrieve_fractions_of_interest(nipt_sample, chromo_focus, chromosomal_fracs)
```

Arguments

nipt_sample	NIPTSample to check wether the strands are combined or separated
chromo_focus	The chromosome of interest
chromosomal_fracs	The chromosomal fractions to extract the chromosome of interest from

Index