QCumber documentation

Gimelbrant lab

v 0.3.2

Mendelevich Asia: a.mendelevich@skoltech.ru

February 18, 2020

QCumber is a set of R tools for quantification of allele-specific expression. It relies on two or more technical replicate RNA-seq libraries to calculate *Quality Correction Constant (QCC)* and use it to correct for allelic imbalance overdipserion.
QCumber analysis starts with a table of allelic counts per gene, calculated from RNA-seq data using any analysis pipeline such as ASEReadCounter* (https://github.com/gimelbrantlab/ASEReadCounter_star).

Paper

Mendelevich A., Vinogradova S.*, Gupta S., Mironov A., Sunyaev S., Gimelbrant A. "Unexpected variability of allelic imbalance estimates from RNA sequencing"*

Source GitHub repository: https://github.com/gimelbrantlab/QCumber

Installation To install current version of this package in R:

```
devtools::install_github("gimelbrantlab/QCumber")
```

Contents

Functions

BuildDesign

Creates a design matrix for the experiment

Usage

```
BuildDesign(experimentNames, techReps, corrConst = NA)
```

Arguments

`experimentNames`	Vector with names of the experiments
`techReps`	Vector with number of technical replicates in each experiment
`corrConst`	Optional, a vector with correction constants for each experiment

Value

Dataframe with experiments numbered and numbers of columns

Examples

```
BuildDesign(c("clone1", "clone2", "clone3"), c(2, 2, 3), c(1.7, 1.8, mean(1.55, 1.6, 1.57)))
```

ComputeAICIs

Calculates Binomial and QCC-corrected binomial CIs for a given vector of AI estimates, and calculates test statistics for comparison with a point or vector of points, for given QCC.

Usage

```
ComputeAICIs(
  inDF,
  vectReps,
  vectRepsCombsCC,
  pt = 0.5,
  Q = 0.95,
  BF = T,
  thr = NA,
  thrUP = NA,
  thrType = "each"
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector of replicate numbers for which the analysis should be applied
`vectRepsCombsCC`	A vector of pairwise-computed correction constants for given replicates (QCC=1 is no correction)
`pt`	Optional (default=0.5), a value or a vector of values to compare with (if second, should be compatible with the order and size of genes vector in table of allelic counts)
`Q`	Optional (default=0.95), confidence level, quantile
`BF`	Optional (default=True), Bonferroni correction for multiple testing, set False ONLY IF Q is alredy corrected
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis

`thrUP` Optional (default=NA), threshold for max gene coverage (default = NA)

`thrType` Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

A table with IDs, AI estimates, coverage, test p-value, and Confidence Intervals

ComputeCorrConstantFor2Reps

Computes QCC for one pair of replicates.

Usage

```
ComputeCorrConstantFor2Reps(
  inDF,
  reps,
  binNObs = 40,
  fitCovThr = 50,
  EPS = 1.05,
  thr = NA,
  thrUP = NA,
  thrType = "each"
)
```

Arguments

`inDF` Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)

`reps` A vector of 2 replicate numbers for which the analysis should be applied

`binNObs` Optional (default=40), threshold on number of observations per bin

`fitCovThr` Optional (default=50), threshold on coverage for genes that will be included in Beta-Bin fitting

`EPS` Optional (default=1.05), base of exponent for the coverage binning, setting greater base (1.1 or 1.2 or 1.3) would result in fewer number of coverage bins in fitting process, thus will increase the computational speed, but may potentially reduce accuracy

`thr` Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis

`thrUP` Optional (default=NA), threshold for max gene coverage (default = NA)

`thrType` Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

List with (1) fitted QCC ($fittedCC) and (2) a table with proportions of observed to expected quantiles per coverage bin ($QObsExpPropsTable).

ComputeCorrConstantsForAllPairsReps

Computes QCC for all possible pairs of given replicates.

Usage

```
ComputeCorrConstantsForAllPairsReps(
  inDF,
  vectReps,
  binNObs = 40,
  fitCovThr = 50,
  EPS = 1.05,
  thr = NA,
  thrUP = NA,
  thrType = "each"
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector of ≥ 2 replicate numbers for which the analysis should be applied
`binNObs`	Optional (default=40), threshold on number of observations per bin
`fitCovThr`	Optional (default=50), threshold on coverage for genes that will be included in Beta-Bin fitting
`EPS`	Optional (default=1.05), base of exponent for the coverage binning, setting greater base (1.1 or 1.2 or 1.3) would result in fewer number of coverage bins in fitting process, thus will increase the computational speed, but may potentially reduce accuracy
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

List of fitting outputs of ComputeCorrConstantFor2Reps() for each combinatorial pair of replicates (in order 1-2,1-3,..,1-N,2-3,..2-N,..,(N-1)-N): list with (1) fitted QCC ($fittedCC) and (2) a table with proportions of observed to expected quantiles per coverage bin ($QObsExpPropsTable).

CountsToAI

Calculates allelic imbalances from merged counts over given replicates (ai(sum_reps(gene))).

Usage

```
CountsToAI(
  df,
  reps = NA,
  meth = "mergedToProportion",
  thr = NA,
  thrUP = NA,
  thrType = "each"
)
```

Arguments

`df`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`reps`	Optional (default=NA, all replicates), a vector of replicate numbers for which the analysis should be applied

`meth` Optional (default="mergedToProportion", also can be "meanOfProportions"), method to use, either sum(m)/sum(p) (default), or sum(m/p)

`thr` Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis

`thrUP` Optional (default=NA), threshold for max gene coverage (default = NA)

`thrType` Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

A table with IDs and calculated AI estimate for given set of replicates

Examples

```
CountsToAI(allelicCountsTable, reps=c(1,2), thr=10, thrUP=1000)
```

GetGatkPipelineTabs

Loads the working parts of tables ("ID", allele count 2x columns, "contig" if specified); concatenates (uniting merge) tables from all provided files.

Usage

```
GetGatkPipelineTabs(inFiles, nReps, contigs = vector())
```

Arguments

`inFiles` A vector of full pathes to files with alelleic counts tables; necessary columns: "ID", pairs of ref and alt allelic counts; optionally, for filtering, "contig"

`nReps` A vector of numbers, either: (1) each entry is a number of replicates in the corresponding file; (2) for one file only, each entry is a number of replicates corresponding to particular experiment

`contigs` Optional (default=vector()), parameter defining if the resulting table should be filtered by contig column (preserving only rows corresponding to a given vector), default set to empty vector() and no filtering applied

Value

A concatenated table with allele counts for all replicates, each row corresponds to a feature ("ID")

MeanCoverage

Calculates mean allelic coverage (mat+pat) among given replicates.

Usage

```
MeanCoverage(df, reps = NA, thr = NA, thrUP = NA, thrType = "each")
```

Arguments

`df` Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)

`reps` Optional (default=NA, all replicates), a vector of replicate numbers for which the analysis should be applied

`thr` Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis

`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

A table with IDs and calculated mean allelic coverage for given set of replicates

Examples

```
MeanCoverage(allelicCountsTable, reps=c(3,4), thr=8)
```

MergeSumCounts

Creates a table of sums of maternal and paternal alellic counts for given replicates.

Usage

```
MergeSumCounts(df, reps = NA, thr = NA, thrUP = NA, thrType = "each")
```

Arguments

`df`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`reps`	Optional (default=NA, all replicates), a vector of replicate numbers for which the analysis should be applied
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

A table with IDs and calculated allelic counts for each of the given replicates

Examples

```
MeanCoverage(allelicCountsTable, reps=c(1,2))
```

MixBetaBinomialFit

Fitting beta-binomial mixture distribution of AI in particular bin.

Usage

```
MixBetaBinomialFit(initials, coverage, observations)
```

Arguments

`initials`	Initials for EM algm: initials = c(w1, alpha1, alpha2), weight of first component and alphas for both beta-binomial distributions in a mixture
`coverage`	A number, that represents the coverage bin
`observations`	A vector of "maternal counts" in the bin

Value

Fitted weight of first component and alphas for both beta-binomial distributions in a mixture, plus number of steps.

MixBetaBinomialFitStep

One step of fitting beta-binomial mixture distribution of AI in particular bin.

Usage

```
MixBetaBinomialFitStep(initials_old, coverage, observations)
```

Arguments

`initials_old`	Initials for EM step: initials = c(w1, alpha1, alpha2), weight of first component and alphas for both beta-binomial distributions in a mixture
`coverage`	A number, that represents the coverage bin
`observations`	A vector of "maternal counts" in the bin

Value

Re-fitted initials for next EM step.

NameColumns

Helper function to quickly rename columns in geneCountTab dataframe

---

Usage

```
NameColumns(exp_n, rep_n)
```

Arguments

`exp_n`	Experiment number
`rep_n`	Number of replicates for the experiment

Value

Vector with names

Examples

```
colnames(allelicCountsTable)[2:9] <- c(NameColumns(1,2), NameColumns(2,2))
```

PerformBinTestAIAnalysisForConditionNPoint_knownCC

Performs Binomial and QCC-corrected binomial tests (with Bonferroni correction) with a given point estimate, for given QCC.

Usage

```
PerformBinTestAIAnalysisForConditionNPoint_knownCC(
  inDF,
  vectReps,
  vectRepsCombsCC,
  pt = 0.5,
  Q = 0.95,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector ($\geq$2) of replicate numbers for which the analysis should be applied
`vectRepsCombsCC`	A vector of pairwise-computed correction constants for given replicates (QCC=1 is no correction)
`pt`	Optional (default=0.5), a value to compare with
`Q`	Optional (default=0.95), confidence level, quantile
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type
`minDifference`	Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

A table of gene names, AIs + CIs, p-values for both non-corrected and (BT..) and QCC corrected (BT_CC..) tests, classification into genes demonstrating signifficant difference (TRUE) from point estimate AI and those that don't (FALSE).

PerformBinTestAIAnalysisForConditionNPoint

Calculates QCC. Performs Binomial and QCC-corrected binomial tests (with Bonferroni correction) with a given point estimate.

Usage

```
PerformBinTestAIAnalysisForConditionNPoint(
  inDF,
  vectReps,
  pt = 0.5,
  binNObs = 40,
  Q = 0.95,
  fitCovThr = 50,
  EPS = 1.05,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector ($\geq$2) of replicate numbers for which the analysis should be applied
`pt`	Optional (default=0.5), a value to compare with
`binNObs`	Optional (default=40), threshold on number of observations per bin
`Q`	Optional (default=0.95), confidence level, quantile
`fitCovThr`	Optional (default=50), threshold on coverage for genes that will be included in Beta-Bin fitting
`EPS`	Optional (default=1.05), base of exponent for the coverage binning, setting greater base (1.1 or 1.2 or 1.3) would result in fewer number of coverage bins in fitting process, thus will increase the computational speed, but may potentially reduce accuracy
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type
`minDifference`	Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

List of (1) fitted QCC for all combanatorial pairs of replicates ($CC), (2) ComputeCorrConstantsForAllPairsReps() output ($FitDATA), and (3) PerformBinTestAIAnalysisForConditionNPoint_knownCC() output ($Output).

PerformBinTestAIAnalysisForConditionNPointVect_knownCC

Performs Binomial and QCC-corrected binomial tests (with Bonferroni correction) with a given vector of point estimates, for given QCC.

Usage

```
PerformBinTestAIAnalysisForConditionNPointVect_knownCC(
  inDF,
  vectReps,
  vectRepsCombsCC,
  ptVect,
  Q = 0.95,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector ($\geq$2) of replicate numbers for which the analysis should be applied
`vectRepsCombsCC`	A vector of pairwise-computed correction constants for given replicates (QCC=1 is no correction)
`ptVect`	A vector of values to compare with, should be compatible with the order and size of genes vector in table of allelic counts

`Q`	Optional (default=0.95), confidence level, quantile
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type
`minDifference`	Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

A table of gene names, AIs + CIs, p-values for both non-corrected and (BT..) and QCC corrected (BT_CC..) tests, classification into genes demonstrating signifficant difference (TRUE) from corresponding point estimate AI and those that don't (FALSE).

PerformBinTestAIAnalysisForConditionNPointVect

Calculates QCC. Performs Binomial and QCC-corrected binomial tests (with Bonferroni correction) with a given vector of point estimates.

Usage

```
PerformBinTestAIAnalysisForConditionNPointVect(
  inDF,
  vectReps,
  ptVect,
  binNObs = 40,
  Q = 0.95,
  fitCovThr = 50,
  EPS = 1.05,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vectReps`	A vector ($\geq$2) of replicate numbers for which the analysis should be applied
`ptVect`	A vector of values to compare with, should be compatible with the order and size of genes vector in table of allelic counts
`binNObs`	Optional (default=40), threshold on number of observations per bin
`Q`	Optional (default=0.95), confidence level, quantile
`fitCovThr`	Optional (default=50), threshold on coverage for genes that will be included in Beta-Bin fitting
`EPS`	Optional (default=1.05), base of exponent for the coverage binning, setting greater base (1.1 or 1.2 or 1.3) would result in fewer number of coverage bins in fitting process, thus will increase the computational speed, but may potentially reduce accuracy
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)

`thrType` Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

`minDifference` Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

List of (1) fitted QCC for all combanatorial pairs of replicates ($CC), (2) ComputeCorrConstantsForAllPairsReps() output ($FitDATA), and (3) PerformBinTestAIAnalysisForConditionNPointVect_knownCC() output ($Output).

PerformBinTestAIAnalysisForTwoConditions_knownCC

Performs differential tests (with Bonferroni correction) for AI values for two conditions, for given QCC.

Usage

```
PerformBinTestAIAnalysisForTwoConditions_knownCC(
  inDF,
  vect1CondReps,
  vect2CondReps,
  vect1CondRepsCombsCC,
  vect2CondRepsCombsCC,
  Q = 0.95,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF` Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)

`vect1CondReps` A vector ($\geq$2) of replicate numbers that should be considered as first condition's tech reps

`vect2CondReps` A vector ($\geq$2) of replicate numbers that should be considered as second condition's tech reps

`vect1CondRepsCombsCC` A vector of pairwise-computed correction constants for first condition's tech reps (QCC=1 is no correction)

`vect2CondRepsCombsCC` A vector of pairwise-computed correction constants for second condition's tech reps (QCC=1 is no correction)

`Q` Optional (default=0.95), confidence level, quantile

`thr` Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis

`thrUP` Optional (default=NA), threshold for max gene coverage (default = NA)

`thrType` Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

`minDifference` Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

A table of gene names, AIs + CIs for both conditions, p-values for both non-corrected (BT..) and QCC corrected (BT_CC..) differential tests, classification into genes demonstrating signifficant difference (TRUE) of AI estimates in two conditions, and those that don't (FALSE).

PerformBinTestAIAnalysisForTwoConditions

Calculates QCC. Performs differential tests (with Bonferroni correction) for AI values for two conditions.

Usage

```
PerformBinTestAIAnalysisForTwoConditions(
  inDF,
  vect1CondReps,
  vect2CondReps,
  binNObs = 40,
  fitCovThr = 50,
  Q = 0.95,
  EPS = 1.05,
  thr = NA,
  thrUP = NA,
  thrType = "each",
  minDifference = NA
)
```

Arguments

`inDF`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`vect1CondReps`	A vector ($\geq$2) of replicate numbers that should be considered as first condition's tech reps
`vect2CondReps`	A vector ($\geq$2) of replicate numbers that should be considered as second condition's tech reps
`binNObs`	Optional (default=40), threshold on number of observations per bin
`fitCovThr`	Optional (default=50), threshold on coverage for genes that will be included in Beta-Bin fitting
`Q`	Optional (default=0.95), confidence level, quantile
`EPS`	Optional (default=1.05), base of exponent for the coverage binning, setting greater base (1.1 or 1.2 or 1.3) would result in fewer number of coverage bins in fitting process, thus will increase the computational speed, but may potentially reduce accuracy
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type
`minDifference`	Optional (default=NA), if specified, one additional column is added to the output (T/F depending if the gene changed AI expression more than minDifference in addition to passing the test)

Value

List of (1) fitted QCC for all combanatorial pairs of replicates for both conditions ($CC), (2) ComputeCorrConstantsForAllPairsReps() output for both conditions ($FitDATA), and (3) PerformBinTestAIAnalysisForTwoConditions_knownCC() output ($Output).

ThresholdingCounts

Takes allelic counts table and returns table, where all genes that don't pass a given coverage threshold have NA coverage. Can be restricted to particular replicates.

Usage

```
ThresholdingCounts(df, reps = NA, thr = NA, thrUP = NA, thrType = "each")
```

Arguments

`df`	Allele counts dataframe: with 2n+1 columns, "ID" and 2n columns with ref & alt counts (rep1_ref, rep1_alt, rep2_ref, rep2_alt, ...)
`reps`	Optional (default=NA, all replicates), a vector of replicate numbers for which the analysis should be applied
`thr`	Optional (default=NA), threshold on the overall number of counts for a gene to be considered in the analysis
`thrUP`	Optional (default=NA), threshold for max gene coverage (default = NA)
`thrType`	Optional (default = "each", also can be "average" for average coverage on replicates), threshold type

Value

Allelic counts table with masked with NA undercovered genes, for selected replicates

Examples

```
ThresholdingCounts(df = allelicCountsTable, reps = c(1,2), thr = 10)
```

Input requirements

Allelic Counts Tables (`GetGatkPipelineTabs` input)

- **Structure:**
  A tab-segmented table with present column names, each line corresponds to a genomic feature (for example, gene). The first column is genomic feature ID, next $2 \times N$ columns stores alellelic counts for reference and alternative alleles for each of N replicates (in order rep1_ref, rep1_alt, rep2_ref, rep2_alt, ... repN_ref, repN_alt). The additional column `contig` should be present in the table at any position $\geq 2N + 1$ if filtering by contig is needed.
- **Column names:**
  If `contig` parameter is set non-empty, a table must have corresponding column named `contig`. The other columns may have any names.

Output formats

AI CI tables

`ID`	Genomic feature ID
`sumCOV`	Total allelic counts
`matCOV`	Maternal/reference allelic counts
`AI`	Allelic Imbalance point estimate
`BT_CIleft`	Left boundary of CI* for non-corrected test
`BT_CIright`	Right boundary of CI* for non-corrected test
`BT_CIleft_CC_1`	Left boundary of CI* for QCC-corrected test (condition 1)
`BT_CIright_CC_1`	Right boundary of CI* for QCC-corrected test (condition 1)
`BT_pval`	Non-corrected test statistics (p-value)
`BT_pval_CC`	QCC-corrected test statistics (p-value)
`BT`	Non-corrected test decision
`BT_CC`	QCC-corrected test decision
`BT_CC_thrDiff`	If `minDifference` was set, QCC-corrected test decision AND passed `minDifference` threshold

Differential AI tables

`ID`	Genomic feature ID
`sumCOV_1`	Total allelic counts (condition 1)
`matCOV_1`	Maternal/reference allelic counts (condition 1)
`AI_1`	Allelic Imbalance point estimate (condition 1)
`BT_CIleft_1`	Left boundary of CI* for non-corrected test (condition 1)
`BT_CIright_1`	Right boundary of CI* for non-corrected test (condition 1)
`BT_CIleft_CC_2`	Left boundary of CI* for QCC-corrected test (condition 1)
`BT_CIright_CC_2`	Right boundary of CI* for QCC-corrected test (condition 1)
`sumCOV_2`	Total allelic counts (condition 2)
`matCOV_2`	Maternal/reference allelic counts (condition 2)
`AI_2`	Allelic Imbalance point estimate (condition 2)
`BT_CIleft_2`	Left boundary of CI* for non-corrected test (condition 2)
`BT_CIright_2`	Right boundary of CI* for non-corrected test (condition 2)
`BT_CIleft_CC_2`	Left boundary of CI* for QCC-corrected test (condition 2)
`BT_CIright_CC_2`	Right boundary of CI* for QCC-corrected test (condition 2)
`BT_pval`	Non-corrected test statistics (p-value)
`BT_pval_CC`	QCC-corrected test statistics (p-value)
`BT`	Non-corrected test decision
`BT_CC`	QCC-corrected test decision
`BT_CC_thrDiff`	If `minDifference` was set, QCC-corrected test decision AND passed `minDifference` threshold

* - for quantile Q corrected on multiple testing using Bonferroni approach ($1 - \frac{1-Q}{\#genes}$), where only considered genes in the analysis were counted.