Package 'IUTA'

May 28, 2014

Type Package

Title Test differential isoform usage between two groups

Version 1.0

Date 2013-08-20

Author Liang Niu

Maintainer Liang Niu <niul@niehs.nih.gov>

Description This package takes Bam files and gene annotation files as input and test differential isoform usage for genes

License GPL (>=2)

Depends Rsamtools, parallel, BiocGenerics, R (>= 2.14.0)

R topics documented:

`IUTA-package` *Isoform Usage Two-step Analysis*

Description

Package `IUTA-package` performs a two-step analysis to detect genes with differential isoform usages (set of relative abundances of isoforms) between two samples. In addition to the main function `IUTA`, it also provide a function `GetGeneGtf` to generate a GTF file suitable for `IUTA` and two functions i.e., `bar_compare` and `pie_compare`, to present a gene's isoform usages between two groups.

Details

Package:	IUTA
Type:	Package
Version:	1.0
Date:	2013-08-20
License:	GPL (>=2)

`pie_compare` provides a comparison of isoform usages of a gene between two groups by two pie plots.

`bar_compare` provides a comparison of isoform usages of a gene between two groups of samples by a bar plot.

`IUTA` takes BAM files from two groups of samples and a GTF file of the related species and tests for differential isoform usages (set of isoform relative abundances) for the inquired genes.

`GetGeneGtf` creates a new GTF file (with correct gene id) from a input GTF file and a gene-transcript reference file.

Author(s)

Liang Niu

Maintainer: Liang Niu <niul@niehs.nih.gov>

References

Liang Niu, Weichun Huang, David M. Umbach and Leping Li (2013). IUTA: a tool for effectively detecting differential isoform usage from RNA-Seq data, in preparation.

See `http://mblab.wustl.edu/GTF22.html` for the details of Gene transfer format (GTF) and `http://samtools.sourceforge.net/SAMv1.pdf` for the details of Sequence Alignment/Map (SAM) format. The BAM format is the compressed binary version of SAM format.

`bar_compare` *Comparison of estimated isoform usages of a gene between two groups of samples by a bar plot*

Description

`bar_compare` creates a pdf file with a bar plot in which vertical bars that represent the same isoform are juxtaposed in the same column. Each column represents an isoform of the gene and each bar in the column represents the estimated relative abundance of the corresponding isoform in a sample. Bars for different sample groups are differently colored. See "Details" for the details of the output bar plot.

Usage

```
bar_compare(gene.name, n1, estimates.file = "estimates.txt",
            output.file =
            paste("Barplot_", gene.name, ".pdf", sep = ""),
            legend.pos = "topright", group.name = c("1","2"),
            output.screen = FALSE)
```

Arguments

gene.name	Name of the gene whose estimated isoform usages are used to create the bar plot.
n1	Number of samples in the first group.
estimates.file	The path (either relative or full) of the tab-delimited text file (with header) which contains the estimated isoform usages for gene gene.name in the samples, e.g., the output file "estimates.txt" for isoform usages from IUTA function. See "Details" for the format of the tab-delimited text file.
output.file	The path (either relative or full) of the output pdf file. See "Details" for the details of the output pdf file.
legend.pos	The location of the legend of the bar plot. It is a keyword from the list ``bottomright'', ``bottom'', ``bottomleft'', ``left'', ``topleft'', ``top'', ``topright'', ``right'' and ``center''. The defult is ``topright''.
group.name	A character vector of the names of the two groups. The first (second) element is the name of the first (second) group. The default names are "1" and "2".
output.screen	Whether to print the bar plot on screen. The default is FALSE (not to print on screen).

Details

The tab-delimied text file with path estimates.file (with header) should contain $2 + n_1 + n_2$ columns: the first two columns are the gene name (column 1) and the isoform (column 2); the next n_1 columns are the estimates of the relative isoform abundance of the isoform from samples in group one; the last n_2 columns are the estimates of the relative isoform abundance of the isoform from samples in group two. Such a file can be obtained by IUTA.

bar_compare first checks the tab-delimited text file with path estimates.file to see if there is any records for the gene gene.name. If there is no record or the estimated relative abundances are all NAs for all the isoforms of the gene in all the samples, bar_compare stops with an error message "NO data for the input gene!"; otherwise bar_compare plots a bar plot using all the valid (not NA) estimated isoform usages and saves the plot in a output pdf file with path output.file. In the bar plot, there are K columns that represent the K isoforms of the gene. In each column, there are $a + b$ juxposed vertical bars, where a (b) is the number of valid estimated isoform usages in group one (two); each bar represents the relative abundance of the corresponding isoform in a sample; the a bars for samples in group one are colored in red and the b bars for samples in group two are colored in green. The title of the plot include the gene name gene.name and the information of a and b. Note that the order of the samples represented by the $a + b$ bars in each column is identical for all the columns.

Value

No value is returned by bar_compare.

Author(s)

Liang Niu

See Also

IUTA

Examples

```
## read the sample tab-delimied file with
## the estimated isoform usage information
estimates<-system.file("sampleoutput","estimates.txt",package="IUTA")

## plot the bar plots to comapre the isoform usages
## of gene "Frmd5" between two groups
bar_compare("Mrpl15",n1=3,estimates,legend.pos="topleft",output.screen=TRUE)

## now the "Barplot_Frmd5.pdf" is
## in the current directory

## remember to delete the pdf file
file.remove("Barplot_Mrpl15.pdf")
```

`GetGeneGtf` *Generation of GTF (Gene transfer format) File with Gene Symbols*

Description

`GetGeneGtf` creates a new gene annotation GTF file from an input gene annotation GTF file and a gene-transcript reference file.

Usage

```
GetGeneGtf(gene.file.name, transcript.gtf.name,
           out.gtf.name = "modified.gtf")
```

Arguments

`gene.file.name`	The path (either relative or full) of the input gene-transcript reference file. See "Details" for the requirements of the gene-transcript reference file
`transcript.gtf.name`	The path (either relative or full) of the input gene annotation GTF file. See "Details" for the requirements of the GTF file.
`out.gtf.name`	The path (either relative or full) of the output gene annotation GTF file.

Details

`GetGeneGtf` is useful for those GTF files with the "gene_id" information in the last (9-th) column is missing or inaccurate (like the GTF file downloaded from UCSC genome browser).\ The `gene.file.name` is the path of the input gene-transcript reference file, which should be a tab-delimited text file without header. The first two columns of the file should be transcript name (column 1) and gene symbol (column 2). It may contain other columns, but are not used by `GetGeneGtf`.

The `transcript.gtf.name` is the path of the gene annotation input GTF file. The last (9-th) column should contain a mandatory "transcript_id" attribute for a GTF file.

The `out.gtf.name` is the path of the output gene annotation GTF file. The new GTF file is the same as the input GTF file, except the last (9-th) column. The last column of the new GTF file has the form "gene_id XXX; transcript_id YYY;", where "XXX" is the gene symbol (inferred from `gene.file.name`) and "YYY" is the transcript name. Such a GTF file is needed for `IUTA`.

Value

No value is returned by `GetGeneGtf`.

Note

If the gene-transcript reference file with path `gene.file.name` does not provide a valid gene symbol for a transcript in the input gene annotation GTF file with path `transcript.gtf.name`, `GetGeneGtf` exclude all records of the transcript from the output GTF file. At the end of `GetGeneGtf`, one or both of the following warningss are then reported: if there is at least one transcript belong to different gene symbol, the warning message is "Found transcript(s) belong to different genes in reference! Such transcript(s) are removed from the gene annotation!"; if there is at least one transcript in the input GTF file but not in the gene-transcript reference file, the warning message is "found transcript(s) in annotation but not in reference! such transcript(s) are removed from the gene annotation!".

Author(s)

Liang Niu

References

See http://mblab.wustl.edu/GTF22.html for the details of GTF format.

Examples

```
## get the paths of sample GTF file and sample reference file
transcript.gtf<-system.file("gtf","mm10_kg_sample.gtf",package="IUTA")
gene.transcript.ref<-system.file("gtf","gene_id.txt",package="IUTA")

## check the last (9-th) column of the first line of transcript.gtf
## notice they are the same
print(read.delim(transcript.gtf,header=FALSE)[1,])

## run GetGeneGtf
GetGeneGtf(gene.transcript.ref,transcript.gtf,"modified.gtf")

## read in the new GTF file and check the gene_id attribute
print(read.delim("modified.gtf",header=FALSE)[1,])

## remove "modified.gtf"
file.remove("modified.gtf")
```

`IUTA` *Isoform Usage Two-step Analysis*

Description

`IUTA` takes RNA-Seq alignment files (in BAM format) from two groups of samples, together with a gene annotation file (in GTF format) for the related species, to test for differential isoform usage (set of relative abundances of isoforms) for each of the inquired genes. It outputs two tab-delimited files (with header): “estimates.txt” and “p_values.txt”. See “Details” for the details of `IUTA` and the details of the two output files.

Usage

```
IUTA(bam.list.1, bam.list.2, transcript.info,
     rep.info.1 = rep(1, length(bam.list.1)),
     rep.info.2 = rep(1, length(bam.list.2)),
     output.dir = paste(getwd(), "/IUTA", sep = ""),
     output.na = FALSE,
     genes.interested = "all",
     strand.specific = rep("1.5",
                   length(rep.info.1)+length(rep.info.2)),
     gene.filter.chr = c("_", "M", "Un"),
     mapq.cutoff = NA, alignment.per.kb.cutoff = 10,
     IU.for.NA.estimate = "even",
     sample.FLD = FALSE, FLD = "empirical",
     mean.FL.normal = NA, sd.FL.normal = NA,
     number.samples.EFLD = 1e+06,
     isoform.weight.cutoff = 1e-4,
     adjust.weight = 1e-4, epsilon = 1e-05,
     test.type = "SKK", log.p = FALSE, fwer = 1e-2,
     mc.cores.user = NA)
```

Arguments

`bam.list.1` A character vector of paths (either relative or full) of the BAM files for the replicates of samples in group one. It has $r_1 + r_2 + \cdots + r_{n_1}$ elements, where r_i is the number of replicates of sample i ($i = 1, 2, \cdots, n_1$) in group one and n_1 is the number of samples in group one. The paths for the replicates of the same sample should be placed together, i.e., the first r_1 elements of `bam.list.1` should be the paths of the r_1 replicates of sample 1, the next r_2 elements of `bam.list.1` should be the paths of the r_2 replicates of sample 2, etc. See "References" for a reference for BAM format.

`bam.list.2` A character vector of paths (either relative or full) of the BAM files for the samples in group two. The format of `bam.list.2` is the same as the format of `bam.list.1`. See "References" for a reference for BAM format.

`transcript.info` The path (either relative or full) of the GTF file for the related species. See "References" for a reference of GTF format. See "Details" for the requirement of the input GTF file.

`rep.info.1` The technical replicate information for the samples in group one. It is a n_1-dimensional vector with the i-th entry be the number of technical replicates in the i-th sample in group one, where n_1 is the number of samples in group one. The default is set in such a way that each sample has only one technical replicate.

`rep.info.2` The technical replicate information for the samples in group two. It is a n_2-dimensional vector with the i-th entry be the number of technical replicates in the i-th sample in group two, where n_2 is the number of samples in group two. The default is set in such a way that each sample has only one technical replicate.

`output.dir` The path (either relative or full) of the directory in which the two output files are stored. If the directory does not exist, `IUTA` will create it. The default is the subdirectory "IUTA" under the working directory.

`output.na` Whether to include genes with `NA` results in the two output text files or not. If it is `TRUE`, all inquired genes are included in the two output text files. If it is `FALSE` (default), genes with `NA` as the estimated isoform usages in ALL

samples of the two groups are excluded from "estimates.txt" and genes with NA as the p-values of ALL tests (including all user-specified tests in test.type) are excluded from "p-values.txt".

genes.interested

A character vector of the inquired gene names. The default is *"all"*, i.e., all genes with more than two isoforms in the filtered gene annotation GTF file. See "Details" for the details of filtering process for the gene annotation GTF file.

strand.specific

A character vector of length $n_1 + n_2$, where n_1 is the number of samples in group one and n_2 is the number of samples in group two. The i-th element (either "1", or "2", or "1.5") of strand.specific indicates that which read (in a read pair) has the same orientation as the mRNA molecule from which the read pair was sequenced from the replicates of sample i. Specifically, if all replicates of sample i were sequenced by a strand-specific protocol such that the first read of each pair has the same orientation as the mRNA molecule from which the read pair was sequenced, then the i-th element of strand.specific is set as "1"; if all replicates of sample i were sequenced by a strand-specific protocol such that the second read of each pair has the same orientation as the mRNA molecule from which the read pair was sequenced, then the i-th element of strand.specific is set as "2"; if all replicates of sample i were sequenced by a non-strand-specific protocol (such as the standard Illumina), then the i-th element of strand.specific is set as "1.5".

gene.filter.chr

A character vector of symbols that are used to filter the genes on "irregular" chromosomes. Specifically, all genes with at least one transcript on chromosomes with these symbols are filtered from the GTF file with path transcript.info and are not considered in the IUTA analysis. The default is c("_","M","Un"), which correspond to "chrN_random" (N is a chromosome number), "chrM" and "chrUn". If the user wants to keep all the "irregular" chromosomes in consideration for IUTA analysis, set gene.filter.chr be NA.

mapq.cutoff The mapping quality cut-off that is used to filter the RNA-Seq read pairs for IUTA. If it is NA (default), ALL read pairs will be used for IUTA. Otherwise, only reads pairs with both mapping qualities bigger than mapq.cutoff are used for IUTA analysis.

alignment.per.kb.cutoff

The unit (per kilobases) cut-off number of "valid" alignments (read pairs) that are needed for IUTA to estimate the isoform usage of a gene. That is, IUTA estimates the isoform usage of a gene in a sample only when the number of "valid" read pairs is bigger than alignment.per.kb.cutoff times the length of the union of exons (in unit of kilobases). The default of alignment.per.kb.cutoff is 10. See "Details" for the definition of a "valid" read pair.

IU.for.NA.estimate

The way that the isoform usage of a gene is estimated for a sample when it cannot be estimated from the data. The "artificial" estimates obtained in this way are only used for (differential isoform usage) testing purpose. IU.for.NA.estimate is only valid when both groups have at least two samples for which the isoform usages of the gene can be estimated from the data (otherwise no test can be performed). IU.for.NA.estimate can be either "even" (default), or "average", or "none". See "Details" for more details.

sample.FLD Whether the fragment length distribution (FLD) for each sample is sample-specific or group-specific. The default is *FALSE*, i.e., use the group-specific

FLD for each sample in the group. The group-specific FLD is the FLD determined for the first sample of the group.

`FLD` Whether to use empirical (*"empirical"*) FLD (EFLD) or normal (*"normal"*) FLD. If it is *"empirical"*, the EFLD is used and it is estimated from the data. If it is *"normal"*, a discrete normal distribution is used as FLD. In the latter case, user can specify the mean and the standard deviation (sd) via `mean.FL.normal` and `sd.FL.normal`; if user does not specify the mean or/and the standard deviation of the normal FLD, the corresponding estimate(s) from the raw EFLD (i.e., before smoothing) will be used.

`mean.FL.normal`
The mean of the normal FLD. Only valid if FLD="normal". The default `NA` is set so that the mean of the raw EFLD (i.e., before smoothing) is used.

`sd.FL.normal` The standard deviation of the normal FLD. Only valid if FLD="normal". The default `NA` is set so that the sd of raw EFLD (i.e., before smoothing) is used.

`number.samples.EFLD`
The maximum number of sample fragments used to estimate EFLD. The default is 10^6.

`isoform.weight.cutoff`
A small non-negative value (less than 1) that is used to determine whether to keep an isoform in consideration when testing for differential isoform usage. Specifically, those isoforms whose estimated relative abundances are no more than `isoform.weight.cutoff` in all samples (excluding those in which isoform usage cannot be estimated from the data) are not considered when testing for differential isoform usage. The default is 10^{-4}. See "Details" for more details.

`adjust.weight`
A small positive value that is used to adjust the estimated isoform usage for testing purpose. Specifically, for the isoforms that are considered in the test(s) of differential isoform usage, all estimated relative abundances that are smaller than `adjust.weight` (including those zeros) are replaced by `adjust.weight` and the tests of differential isoform usage are based on the adjusted estimates. The main purpose of such adjustment is to make the isometric logratio transformation (ilr) applicable for the estimated isoform usages, since ilr is not applicable when an isoform usage has zero entries. The default is 10^{-4}.

`epsilon` A small positive value used in the stop criterion of the EM algorithm for estimating isoform usages. The EM stops when the (Euclidean) distance between two consecutive estimations is smaller than `epsilon`. The default is 10^{-5}.

`test.type` A character vector consists of the test types that the user wants to use for testing differential isoform usage in `IUTA`. Three types of test are available: "SKK" (default), "CQ" and "KY". The character vector is composed using the three test types, e.g., c("SKK","CQ"), or c("CQ","SKK","KY"). See "Details" and "References".

`log.p` Whether to output logarithm of p-values or p-values. The default is `FALSE`, i.e., to output p_values.

`fwer` The family-wise error rate (FWER) that the user wants to control for the the main test (the first test in `test.type`). In `IUTA`, the FWER is controlled by Bonferroni correction. Specifically, all genes with p-values less than $\texttt{fwer}/n_t$ are claimed as genes with differential isoform usage, where n_t is the total number of valid tests. The default is `0.01`.

mc.cores.user

The number of cores to use, i.e. at most how many child processes will be run simultaneously. The default (NA) is set to use all cores that R detects on the machine. Note that in windows mc.cores.user has to be set to be 1, since the function mclapply used in IUTA is not applicable when mc.cores.user bigger than 1.

Details

IUTA first checks the input gene annotation GTF file with path transcript.info to remove records for the following three types of genes: those with isoforms on "irregular" chromosomes (according to gene.filter.chr, e.g., when gene.filter.chr=c("_","M","Un") (default), the "irregular" chromosomes are chrN_random (N is a chromosome number), chrM and chrUn), those with isoforms on different chromosomes and those with isoforms on different strands. The new GTF file consists of the remaining records is used for the further analysis. If genes.interested is "all", IUTA then estimates the isoform usage and tests for differential isoform usage for all the genes with at least two isoforms in the new GTF file; otherwise, IUTA removes the genes that are not in the new GTF file from gene.interested and performs further analysis on the remaining genes in gene.interested, the number of removed genes is reported.

After the genes for the further analysis are selected, IUTA combines the BAM files of the technical replicates into a single BAM file for each sample, and then iterates such BAM files one by one to estimate the isoform usage for each selected genes in each sample using the fragment length distribution (FLD) for the sample. The FLD can be either an empirical fragment length distribution (EFLD) or a discrete normal distribution, depending on FLD, and it can be either identical across the samples within a group or sample-specific, depending on sample.FLD.

If FLD="empirical" and sample.FLD="true", the FLD for the sample is set to be a sample-specific EFLD that is obtained from the (possibly combined) BAM file for the sample. To obtain the EFLD, IUTA makes use of those "stand-alone" exons in the (filtered) GTF file, i.e., exons that do not overlap with any exons of any gene but themselves and proceeds in iterations. Specifically, In each iteration, IUTA selects 1000 "stand-alone" exons in the decreasing order of exon length, and reads the (possibly combined) BAM file to select paired-end reads that satisfy all the following three requirements: both reads in the pair fall into any of the "stand-alone" exons selected for the iteration; both reads in the pair has mapping quality bigger the mapq.cutoff (when mapq.cutoff is not NA, otherwise, this requirement is ignored); both reads in the pair have flags consistent with the strand.specific and the direction of the exon they fall in. For each such read pair, a fragment is inferred and the fragment length is recorded. The iteration stops when either the number of inferred fragments exceeds number.samples.EFLD or the "stand-alone" exons are all used. The raw EFLD is then the relative frequency distribution of recorded lengths. The mean and standard deviation (sd) of the raw ELFD, together with the number of recorded fragments, are reported. By smoothing the raw EFLD by a smoothing window of length 11, i.e., the function value of length l is the average of relative frequencies of fragments with length between $l-5$ and $l+5$, and then standardizing the resulted function, the EFLD is obtained.

If FLD="empirical" and sample.FLD="false", then the FLD for the sample is set to be the EFLD that is obtained by the above procedure from the (possibly combined) BAM file for the first sample in the group, thus the FLD is group-specific.

If FLD="normal", a discrete normal distribution is used as the FLD for the sample. A warning "Please consider using EFLD estimates for Fragment Length Distribution if they are much different from the user specified ones!" is printed, either for each sample (when sample.FLD="true") or for the first sample of each group (when sample.FLD="false"). If sample.FLD="true", the normal FLD is sample-specific; otherwise the normal FLD is group-specific. In fact, sample.FLD takes no effect when the user specifies both the mean (via mean.FL.normal) and the sd (via sd.Fl.normal), since the same mean and sd are used for all samples. However, sample.FLD

takes effect when the mean and/or the sd are not specified. Specifically, when the mean and/or the sd are not specified, IUTA sets the mean and/or the sd as the mean and/or the sd of the raw EFLD of the sample when sample.FLD="true"; and sets the mean and/or the sd as the mean and/or the sd of the raw EFLD of the first sample in the group when sample.FLD="false".

Once FLD is achieved, IUTA starts to estimate the isoform usages gene by gene. For each gene of interest, IUTA reads the (possibly combined) BAM file to get reads that fall into the gene region (including both exons and introns) and selects paired-end reads satisfying the following three requirements: both reads in a pair are consistent at least one isoform of the gene (i.e., can be from a fragment of the isoform); both reads in a pair has mapping quality bigger than mapq.cutoff (when mapq.cutoff is not NA, otherwise, this requirement is ignored); both reads in a pair have flags consistent with the read.direction and the direction of the gene. Then for each such pair, IUTA calculates the length of its corresponding fragment on each compatible isoform and calculates the probabilities of lengths based on FLD; if all such probabilities are zero, the pair is then discarded. All the remaining pairs are called "valid" pairs. If there are enough "valid" pairs, i.e., more than the product of alignment.per.kb.cutoff and the length of the union of exons (in unit of kilobases), IUTA then performs an EM algorithm to find the MLE of isoform usage based on the IUTA model (see "References" for IUTA model) using the length information as observed data; otherwise, IUTA records NA as the estimated isoform usage for the gene in the sample. The estimated isoform usage is written into the tab-delimited text file "estimates.txt" at the end of IUTA.

After a sample is processed, IUTA reports a summary of the analysis for the sample. In the summary, IUTA reports the number of genes with no reads after filtering, the number of genes with no data fits annotation, the number of genes with no enough data fits FLD and the number of genes with isoform usages estimated.

After IUTA processed all the samples and gets the estimated isoform usages for the genes of interest in all samples, IUTA then tests for differential isoform usage for each gene using the estimated gene isoform usages.

For each gene, IUTA requires that there are at least two valid estimates, i.e., not NA, in both groups, otherwise, IUTA cannot perform any test and records NA as the p-value. IUTA also assumes zero relative abundance (in both groups) for the isoforms with small (less than isoform.weight.cutoff) estimated relative abundances across all samples, and performs tests based on the isoform usage formed by the relative abundances of the other, say K, isoforms. If $K = 0$, then IUTA cannot perform any tests and records NA as the p-value; if $K = 1$, then IUTA assumes that there is only one isoform are produced in all samples and records the p-value as 1 (or 0 when log.p=TRUE); if $K > 1$, then IUTA replaces the small (less than adjust.weight, can be zero) entries of the estimated isoform usages (K-dimensional) by adjust.weight, such replacement has two advantages: first, it makes the isometric logratio transformation (ilr, see "References") be applicable to the estimated isoform usages, as ilr is not applicable to an estimated isoform usage with zero entries; second, it makes the tests less sensitive to the (isoform usage) estimation error caused by the noise in the alignment data, as such error can affect the test result dramatically. Note that the number of the valid estimates in each group is recorded and later output as the "test_sample_size" for the gene in the text file "p_values.txt", for all genes of interest.

In addition to the above data preprocessing procedures, IUTA also checks the argument IU.for.NA.estimate to decide whether and how the extra "artificial" estimated isoform usage should be created to perform the test(s) of differential isoform usage. Specifically, if IU.for.NA.estimate is ``even'' (default), IUTA assumes that the estimated isoform usage is a $K$-dimensional vector with all entries equal to $\frac{1}{K}$ for each sample with no valid estimated isoform usage, if IU.for.NA.estimate is ``average'', codeIUTA assumes that the estimated isoform usage is the average (in Aitchison geometry) of the valid estimated isoform usages of the corresponding group for each sample with no valid estimated isoform usage, if IU.for.NA.estimate is ``none'', codeIUTA does not create "artificial" estimated isoform usages for the samples with no valid estimated isoform usage. In the first two cases ( IU.for.NA.estimate is ``even'' or ``average''), both "arti-

ficial" estimates and valid (data-based) estimates are used to perform the tests. In general, setting `IU.for.NA.estimate` as ``even'' makes the test results more conservative, that is, the test results have low type I error rates; and setting `IU.for.NA.estimate` as ``average'' makes the test results more powerful, that is, the test results have higher power.

To do tests, `IUTA` performs ilr to all the estimates, which may include the valid isoform usage estimates (possibly adjusted) and the "artificial" estimates, to transform these K-dimensional vectors to $K-1$-dimensional vectors. `IUTA` assumes the transformed estimates follow group-specific multivariate normal distributions and performs the user-specified test(s) in `test.type` (whenever applicable). Notice that when $K = 2$, the "KY" test becomes Welch's t-test. Since ilr is a isometric transformation between K-dimensional open simplex with Aitchison geometry (See "References") and $(K-1)$-dimensional real space, the test for equal group mean of the transformed estimates is equivalent to the test of equal group mean (in Aitchison geometry) of the original estimates. All p-values (or log of it, if `log.p=TRUE`) are recorded and are written in the tab-delimited text file "p_values.txt".

Finally, `IUTA` outputs two tab-delimited text files with header, "estimates.txt" and "p_values.txt", in the directory with path `output.dir`. The file "p_values.txt" contains a table with $3+1+1+(m-1)+1$ columns, where m is the number of tests in `test.type`. The first three columns are "gene" (gene name), "number_of_isoform" (number of isoforms of the gene), "test_sample_size" (number of samples of each group in which the isoform usage can be estimated, separated by comma). The fourth column is "test", which is the type of test used to calculate the next column "p_value" (either the first test type in `test.type`, or `NA` when the test outputs `NA`). The fifth column is "p_value", which is the output p-value for the gene by the test in column "test". The next $m-1$ columns corresponding to the p-values by the tests in `test.type` except the first type of test in `test.type`. If `log.p=TRUE`, the logarithm of p-values are output instead of p-values; Notice that "KY" test is only applicable for genes with number of samples in each group bigger than $K-1$, otherwise the output p-value for "KY" is `NA`. The last column is "significant", which can be either "yes", or "no", or `NA`. This is determined by the fifth column "p_value" and the family-wise error rate `fwer` that the user wants to control by the Bonferroni correction. Specifically, all genes with p-value (the "p_value" when `log.p`="FALSE"; the exponential of "p_value" when `log.p`="TRUE") less than $\texttt{fwer}/n_t$ are claimed as genes with differential isoform usage, i.e., with "significant" as "yes"; all genes with "p-value" no less than $\texttt{fwer}/n_t$ are claimed as genes with same isoform usage, i.e., with "significant" as "no"; all genes with "p-value" as `NA` has "significant" as `NA`, where n_t is the number of valid tests, i.e., the number of genes with valid "p-value"s (not `NA`). The table is sorted by the column "p_value" in increasing order. The file "estimates.txt" contains a table with $2 + n_1 + n_2$ columns: the first two columns are "gene" (gene name) and "isoform" (isoform of the gene); the next n_1 columns are the estimates of relative isoform abundance of the isoform from samples in group one; the last n_2 columns are the estimates of relative isoform abundance of the isoform from samples in group two. The name of each of the last n_1+n_2 columns is the file name of the BAM file of the first replicate of the corresponding sample, with extension ".bam" omitted. The gene order of the table is same as in the table in "p_values.txt", and the corresponding isoforms of each gene are ordered alphabetically. There are two comment lines on the top of the table, which provide information about the number of genes analyzed, sample sizes and that which ("normal" or "empirical") FLD is used.

Value

No value is returned by `IUTA`.

Note

`mc.cores.user` has to be set to 1 in Windows.

Author(s)

Liang niu

References

Pawlowsky-Glahn, V. and Egozcue, J. J. (2001). Geometric approach to statistical analysis on the simplex. *Stochastic Environmental Research and Risk Assessment*, **15(5)**, 384–398.

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., and Barcel\'o-Vidal,C. (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, **35(3)**, 279–300.

Liang Niu, Weichun Huang, David M. Umbach and Leping Li (2013). IUTA: a tool for effectively detecting differential isoform usage from RNA-Seq data, in preparation.

See http://mblab.wustl.edu/GTF22.html for the details of Gene transfer format (GTF) and http://samtools.sourceforge.net/SAMv1.pdf for the details of Sequence Alignment/Map (SAM) format. The BAM format is the compressed binary version of SAM format.

Examples

```
## set the paths for the BAM file and GTF file
## notice that the gtf file contains correct gene_id information
bam.list.1<-system.file("bamdata",paste("sample_",1:3,".bam",sep=""),
                        package="IUTA")
bam.list.2<-system.file("bamdata",paste("sample_",4:6,".bam",sep=""),
                        package="IUTA")
transcript.info<-system.file("gtf","mm10_kg_sample_IUTA.gtf",
                             package="IUTA")

## run IUTA in Unix or MacOS (not for Windows!)
IUTA(bam.list.1,bam.list.2,transcript.info,output.dir=getwd(),
     FLD="normal",mean.FL.normal=250,sd.FL.normal=10,
     test.type=c("SKK","CQ","KY"))
## or run IUTA in Windows
IUTA(bam.list.1,bam.list.2,transcript.info,output.dir=getwd(),
     FLD="normal",mean.FL.normal=250,sd.FL.normal=10,
     test.type=c("SKK","CQ","KY"),mc.cores.user=1)


## check the results in file
print(read.delim("estimates.txt",comment.char="#")[1:3,])
print(read.delim("p_values.txt")[1,])

## remove the output text files and BAM index files
file.remove(c("estimates.txt","p_values.txt"))
file.remove(system.file("bamdata",paste("sample_",1:6,".bam.bai",sep=""),
            package="IUTA"))
```

`pie_compare` *Comparison of estimated isoform usages of a gene between two groups by pie plots*

Description

pie_compare creates a pdf file in which two pie plots are presented side by side, separated by a legend. Each of the two pie plots represents the estimated isoform usage of the gene in each group and the estimated isoform usage is calculated as the average (in either Euclidean or Aitchison geometry) of the estimated isoform usages in estimates.file for the samples in the group.

Usage

```
pie_compare(gene.name, n1, estimates.file = "estimates.txt",
            geometry = "Euclidean", adjust.weight = 1e-300,
            output.file =
            paste("Pieplot_", gene.name, ".pdf", sep = ""),
            group.name = c("1", "2"),
            output.screen=FALSE)
```

Arguments

gene.name	Name of the gene.
n1	Number of samples in the first group.
estimates.file	The path (either relative or full) of the tab-delimited text file (with header) which contains the estimated isoform usages of genes in gene.name in both two groups, e.g., the output file "estimates.txt" by function IUTA.
geometry	In which geometry the average of the estimated isoform usages in each group is calculated. It can be either "Eucliden" or "Aitchison". The default is "Euclidean", i.e., the average in the normal sense. See "References" for details of Aitchison geometry.
adjust.weight	A small positive value that is used to replace the zero entries (if any) of the estimated isoform usages (from estimates.file) when geometry="Aitchison". The purpose of such adjustments is to make the isometric logratio transformation (ilr) applicable to the estimated isoform usages, which is an essential step when calculating the average of the estimated isoform usages in Aitchison geometry. The default is 10^{-2}. See "References" for details of ilr transformation.
output.file	The path (either relative or full) of the output pdf file. See "Details" for the details of the output pdf file.
group.name	A character vector of the names of the two groups. The first (second) element is the name of the first (second) group. The default names are "1" and "2".
output.screen	Whether to print the pie plot(s) on screen. The default is FALSE (not to print on screen).

Details

The pie_compare is similar to bar_compare. Like bar_compare, pie_compare takes a tab-delimited text file with path estimates.file, which contains the estimated isoform usages for the gene gene.name in samples from two groups, and compare the isoform usages in the two groups graphically. The difference is that pie_compare creates pie plots of the averages of the estimated isoform usages in the two groups while bar_compare creates bar plots of the estimated isoform usages in all samples of the two groups. See the "Details" of bar_compare for the format of text file with path estimates.file.

Note that when no estimates is recorded in the text file with path `estimates.file` for the gene `gene.name` in group one (two), then the output pdf file with path `output.file` contains only one pie plot for group two (one). Also note that no percentage labels for isoforms with relative abundances less than 0.005 (to avoid overlapped labels).

Value

No value is returned by `pie_compare`.

Author(s)

Liang Niu

References

Pawlowsky-Glahn, V. and Egozcue, J. J. (2001). Geometric approach to statistical analysis on the simplex. *Stochastic Environmental Research and Risk Assessment*, **15(5)**, 384–398.

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., and Barcel\'o-Vidal,C. (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, **35(3)**, 279–300.

See Also

`IUTA`

Examples

```
## read the sample tab-delimied file with
## the estimated isoform usage information
estimates<-system.file("sampleoutput","estimates.txt",package="IUTA")

## plot the pie plots to comapre the isoform usages
## of gene "Frmd5" between two groups
pie_compare("Mrpl15",n1=3,estimates,output.screen=TRUE)

## now the "Pieplot_Frmd5.pdf" is
## in the current directory

## remember to delete the pdf file
file.remove("Pieplot_Mrpl15.pdf")
```

Index