

Package ‘IVAS’

October 9, 2015

Type Package

Title Identification of genetic Variants affecting Alternative Splicing

Version 1.0.0

Author Seonggyun Han, Sangsoo Kim

Maintainer Seonggyun Han <hangost@ssu.ac.kr>

Description Identification of genetic variants affecting alternative splicing.

License GPL-2

Depends R (> 3.0.0), GenomicFeatures

Imports doParallel, lme4, Matrix, BiocGenerics, GenomicRanges, IRanges, foreach, AnnotationDbi, S4Vectors, GenomeInfoDb

Suggests BiocStyle

biocViews AlternativeSplicing, DifferentialExpression, DifferentialSplicing, GeneExpression, GeneRegulation, Regression, RNASeq, Sequencing, SNP, Software, Transcription

NeedsCompilation no

R topics documented:

IVAS-package	2
calSignificant	2
chrseparate	3
findAlternative	4
findOversnp	5
MsqtlFinder	6
samplexp	7
samplesnp	8
samplesnplocus	9
saveBplot	9
sqtlfinder	10

Index	12
--------------	-----------

IVAS-package	<i>IVAS : Identification of genomic variants affecting Alternative Splicing</i>
--------------	---

Description

The tool is to detect genomic variants affecting the alternative splicing using genotypic and gene expression data(RNA-seq).

calSignificant	<i>Calculates P.values by using two statistical models.</i>
----------------	---

Description

calculates P.values for association between expressions and genotypes by using the linear regression model and/or generalized linear mixed model.

Usage

```
calSignificant(tx.gene=NULL, total.locus=NULL, exon.locus=NULL, intron.locus=NULL,
info.strand=NULL, overapvalue=NULL, chrnum=NULL, expdata=NULL, snpdata=NULL, method=NULL)
```

Arguments

tx.gene	The matrix of transcripts including transcript IDs, Ensembl gene names, Ensembl transcript names, transcript start sites, and transcript end sites.
total.locus	Ranges including alternative exons and flanking introns in a gene.
exon.locus	All exon locus of a single gene.
intron.locus	All intron locus of a single gene.
info.strand	The strand information of a single gene (forward strand = "+", reverse strand = "-").
overapvalue	Snps located in the alternative exons and the flanking introns
chrnum	The chromosome number of a single gene.
expdata	Dataframe of expression data.
snpdata	Dataframe of genotype data.
method	The option for statistical models and boxplot.("lm" : analysis using linear regression model, "glm" : analysis using generalized linear mixed model, "both" : "lm" and "glm", and "boxplot" : for writing boxplot).

Value

The lm or glm method returns matrix including; SNP marker IDs, Chromosome numbers, alternative exons ranges, Intron ranges, alternative types, P values, information of differential median values of expression ratio among genotypes ("sig" if differential median > 0.1 and "not sig" otherwise), gene names, and methods ("lm" or "glm"). The boxplot method returns matrix with relative ratio values and genotypes of samples.

Author(s)

Seonggyun Han, Sangsoo Kim

References

Chambers, J. M. (1992) Linear models. Chapter 4 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole. Breslow, N.E. Clayton, D.G. (1993). Approximate Inference in Generalized Linear Mixed Models. Journal of the American Statistical Association 88.

See Also

[findOverlaps](#), [lm](#), [glmer](#)

chrseparate

Separate a TranscriptDb object based on a chromosome.

Description

With the `isActiveSeq` method in `GenomicFeatures` package, this function filters the `transcriptDb` object in the `GenomicFeatures` package based on a single chromosome.

Usage

```
chrseparate(transdb = NULL, chrname = NULL)
```

Arguments

<code>transdb</code>	The <code>transcriptDb</code> object in the <code>GnomicFeatures</code> package.
<code>chrname</code>	The chromosome number you would like to select from <code>TnascriptDb</code>

Value

This function returns the `TnascriptDb` limited to the chromosome number that you want.

Author(s)

Seonggyun Han, Sangsoo Kim

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan M, and Carey V. Software for Computing and Annotating Genomic Ranges. PLoS Computational Biology, 9, e1003118. 2013.

See Also

[isActiveSeq](#), [seqinfo](#)

Examples

```
sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
filtered.txdb <- chrseparate(sample.Txdb,19)
```

findAlternative	<i>Find alternative exons of a gene.</i>
-----------------	--

Description

Search alternative exons among transcript isoforms from a single gene.

Usage

```
findAlternative(geneid = NULL, txTable = NULL, totalExrange = NULL,
totalInrange = NULL, one.chr = NULL)
```

Arguments

geneid	Ensembl gene name.
txTable	The matrix of transcripts including transcript IDs, Ensembl gene names, Ensembl transcript names, transcript start sites, and transcript end sites.
totalExrange	A list of GRnages objects including total exon ranges in each transcript resulted from the exonsBy function in GenomicFeatures.
totalInrange	A list of GRnages objects including total intron ranges in each transcript resulted from the intronsByTranscript function in GenomicFeatures.
one.chr	The chromosome number that you want.

Value

alterIntron	A GRanges object with flanking introns of alternative exons
tableBygene	An information table of transcripts including transcript IDs, Ensembl gene names, Ensembl transcript names, transcript start sites, and transcript end sites.
exonRange	All exons locus of a gene
intronRange	All intron locus of a gene

Author(s)

Seonggyun Han, Sangsoo Kim

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan M, and Carey V. Software for Computing and Annotating Genomic Ranges. PLoS Computational Biology, 9, e1003118. 2013.

See Also

[GRanges](#), [IRanges](#)

Examples

```
sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
filtered.txdb <- chrseparate(sample.Txdb,19)
trans.exon.range <- exonsBy(filtered.txdb,by="tx")
trans.intron.range <- intronsByTranscript(filtered.txdb)
txTable <- select(filtered.txdb, keys=names(trans.exon.range),
  columns=c("TXID","TXNAME","GENEID","TXSTART","TXEND"), keytype="TXID")
Altvalue <- findAlternative("ENSG00000170889",txTable,trans.exon.range,trans.intron.range,19)
```

findOversnp	<i>Find SNPs which belongs to alternative exons and flanking introns of them.</i>
-------------	---

Description

Find SNPs which belong to alternative exons and flanking introns of them.

Usage

```
findOversnp(altInvalue = NULL, snprange = NULL)
```

Arguments

altInvalue	A list data set from the findAlternative function.
snprange	A matrix of SNP ranges.

Value

This function returns a matrix with SNPs in alternative exons and flanking introns and ranges of those SNPs.

Author(s)

Seonggyun Han, Sangsoo Kim

See Also

[findOverlaps](#)

Examples

```

sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
data(samplesnplocus)
data(samplesnp)
filtered.txdb <- chrseparate(sample.Txdb,19)
trans.exon.range <- exonsBy(filtered.txdb,by="tx")
trans.intron.range <- intronsByTranscript(filtered.txdb)
txTable <- select(filtered.txdb, keys=names(trans.exon.range),
columns=c("TXID", "TXNAME", "GENEID", "TXSTART", "TXEND"), keytype="TXID")
ch.snp.locus <- as.matrix(samplesnplocus[samplesnplocus[,2] == 19,])
ch.snps <- matrix(ch.snp.locus[is.element(ch.snp.locus[,1],rownames(samplesnp)),],ncol=3,byrow=FALSE)
ch.snps.range <- GRanges(seqnames=Rle(19),ranges=IRanges(start=as.integer(ch.snps[,3]),
end=as.integer(ch.snps[,3])),metadata=ch.snps[,1])
Altvalue <- findAlternative("ENSG00000170889", txTable,trans.exon.range,trans.intron.range,19)
overlapsnp <- findOversnp(Altvalue,ch.snps.range)

```

MsqtlFinder

Find SQTls in multiple genes.

Description

This function enables one to analyze multiple genes using multi-thread version of the foreach function and joins output results from sqtlfinder function. Moreover, it calculates the FDR using P-values of the matrix result data.

Usage

```

MsqtlFinder(expdata = NULL, snpdata = NULL, snplocus = NULL, GTFdata = NULL,
met = NULL, Ncor = 1, bplotout = NULL, cutFDR = 0.01)

```

Arguments

expdata	Dataframe of expression data.
snpdata	Dataframe of genotype data.
snplocus	Locus of SNP markers in the snpdata.
GTFdata	The transcriptDb object in the GnomiFeatures package.
met	The option for statistical models.("lm" : analysis using linear regression model, "glm" : analysis using generalized linear mixed model,and "both" : "lm" and "glm").
Ncor	The number of cores for multi-threads.
bplotout	A directory saving boxplots
cutFDR	The false discovery rate value you would like to set threshold.

Value

This function returns the result matrix including SNP markers ID, chromosome number, alternative exons range, intron ranges, alternative type, P value, information of differential median values of expression ratio among genotypes ("sig" if differential median > 0.1 and "not sig" otherwise), gene names, methods ("lm" or "glm").

Author(s)

Seonggyun Han, Sangsoo Kim

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan M, and Carey V. Software for Computing and Annotating Genomic Ranges. PLoS Computational Biology, 9, e1003118. 2013.

Benjamini, Yoav, Hochberg, and Yosef. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B 57, 289-300. 1995.

See Also

[foreach](#), [GRanges](#), [p.adjust](#)

Examples

```
sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
data(sampleexp)
data(samplesnp)
data(samplesnplocus)
#final.result <- MsqtlFinder(sampleexp,samplesnp,samplesnplocus,sample.Txdb,"lm",1)
```

sampleexp

CEU expression data

Description

CEU expression data including 78 individuals

Usage

```
data("sampleexp")
```

Format

A data frame with 64 transcript expressions on the 78 individuals

Value

A data frame with 64 transcript expressions on the 78 individuals

Source

The data was generated by GEUVADIS (Genetic European Variation in Health and Disease, A European Medical Sequencing Consortium) RNA sequencing project for 1000 Genomes samples (<http://www.geuvadis.org/web/geuvadis/RNAseq-project>).

References

Tuuli Lappalainen, et al. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* 501, 506-511.

Examples

```
data(sampleexp)
```

samplesnp

CEU genotype data

Description

CEU genotype data including 78 individuals

Usage

```
data("samplesnp")
```

Format

A data frame with 11 SNPs on the 78 individuals

Value

A data frame with 11 SNPs on the 78 individuals

Source

The data has 1000 genomes Phages 1 dataset and was imputed by GEUVADIS (Genetic European Variation in Health and Disease, A European Medical Sequencing Consortium) RNA sequencing project for 1000 Genomes samples (<http://www.geuvadis.org/web/geuvadis/RNAseq-project>).

References

Tuuli Lappalainen, et al. (2013). Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* 501, 506-511.

Examples

```
data(samplesnp)
```

samplesnplocus	<i>snplocus</i>
----------------	-----------------

Description

snplocus

Usage

```
data("samplesnplocus")
```

Format

A data frame with 11 SNPs and locus of them

Value

A data frame with 11 SNPs and locus of them

Examples

```
data(samplesnplocus)
```

saveBplot	<i>Save boxplots</i>
-----------	----------------------

Description

Save boxplots

Usage

```
saveBplot(sig.sqtl = NULL, expdata = NULL, snpdata = NULL,
snplocus = NULL, GTFdata = NULL, outdir = NULL)
```

Arguments

sig.sqtl	A matrix of significant SQTls from the sqtlfinder function
expdata	Dataframe of expression data.
snpdata	Dataframe of genotype data.
snplocus	Locus of SNP markers in the snpdata.
GTFdata	The transcriptDb object in the GnomiceFeatures package.
outdir	A directory saving boxplots

Value

This function draws the boxplot

Author(s)

Seonggyun Han, Sangsoo Kim

See Also

[boxplot](#)

Examples

```
sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
data(samplesnplocus)
data(sampleexp)
data(samplesnp)
filtered.txdb <- chrseparate(sample.Txdb,19)
trans.exon.range <- exonsBy(filtered.txdb,by="tx")
trans.intron.range <- intronsByTranscript(filtered.txdb)
txTable <- select(filtered.txdb, keys=names(trans.exon.range),
  columns=c("TXID","TXNAME","GENEID","TXSTART","TXEND"), keytype="TXID")
ch.snp.locus <- as.matrix(samplesnplocus[samplesnplocus[,2] == 19,])
ch.snps <- matrix(ch.snp.locus[is.element(ch.snp.locus[,1],rownames(samplesnp)),],ncol=3,byrow=FALSE)
ch.snps.range <- GRanges(seqnames=Rle(19),ranges=IRanges(start=as.integer(ch.snps[,3]),
  end=as.integer(ch.snps[,3])),metadata=ch.snps[,1])
Altvalue <- findAlternative("ENSG00000170889",txTable,trans.exon.range,trans.intron.range,19)
overlapsnp <- findOversnp(Altvalue,ch.snps.range)
sqt1.result <- sqtlfinder(Altvalue,overlapsnp,sampleexp,samplesnp,"lm")
saveBplot(sqt1.result,sampleexp,samplesnp,samplesnplocus,filtered.txdb,"./result")
```

sqtlfinder

Find SQTs.

Description

Find significant SNPs using the calSignificant function.

Usage

```
sqtlfinder(altInvalue = NULL, overapvalue = NULL, expdata = NULL, snpdata = NULL, method = NULL)
```

Arguments

altInvalue	A list data set from the findAlternative function.
overapvalue	A matrix data with SNPs in the flanking introns of alternative exons and ranges of those SNPs from findOversnp function.
expdata	Expression data of samples.
snpdata	Genotype data of samples.
method	The option for statistical models and boxplot("lm" : analysis using linear regression model, "glm" : analysis using generalized linear mixed model, "both" : "lm" and "glm",and "boxplot" : for writing boxplot).

Value

The lm or glm method returns matrix data including SNP markers ID, chromosome number, alternative exons range, intron ranges, alternative type, P value, information of differential median values of expression ratio among genotypes ("sig" if differential median > 0.1 and "not sig" otherwise), a gene name, methods ("lm" or "glm"),and strand information of the gene. The boxplot method returns matrix data with relative ratio values and genotypes of samples.

Author(s)

Seonggyun Han, Sangsoo Kim

Examples

```
sampleDB <- system.file("extdata", "sampleDB", package="IVAS")
sample.Txdb <- loadDb(sampleDB)
data(samplesnplocus)
data(sampleexp)
data(samplesnp)
filtered.txdb <- chrseparate(sample.Txdb,19)
trans.exon.range <- exonsBy(filtered.txdb,by="tx")
trans.intron.range <- intronsByTranscript(filtered.txdb)
txTable <- select(filtered.txdb, keys=names(trans.exon.range),
columns=c("TXID","TXNAME","GENEID","TXSTART","TXEND"), keytype="TXID")
ch.snp.locus <- as.matrix(samplesnplocus[samplesnplocus[,2] == 19,])
ch.snps <- matrix(ch.snp.locus[is.element(ch.snp.locus[,1],rownames(samplesnp))],,ncol=3,byrow=FALSE)
ch.snps.range <- GRanges(seqnames=Rle(19),ranges=IRanges(start=as.integer(ch.snps[,3]),
end=as.integer(ch.snps[,3])),metadata=ch.snps[,1])
Altvalue <- findAlternative("ENSG00000170889",txTable,trans.exon.range,trans.intron.range,19)
overlapsnp <- findOversnp(Altvalue,ch.snps.range)
sqt1.result <- sqtlfinder(Altvalue,overlapsnp,sampleexp,samplesnp,"lm")
```

Index

*Topic **datasets**

sampleexp, [7](#)

samplesnp, [8](#)

samplesnplocus, [9](#)

*Topic **package**

IVAS-package, [2](#)

boxplot, [10](#)

calSignificant, [2](#)

chrseparate, [3](#)

findAlternative, [4](#)

findOverlaps, [3](#), [5](#)

findOversnp, [5](#)

foreach, [7](#)

glmer, [3](#)

GRanges, [5](#), [7](#)

IRanges, [5](#)

isActiveSeq, [3](#)

IVAS (IVAS-package), [2](#)

IVAS-package, [2](#)

lm, [3](#)

MsqtlFinder, [6](#)

p.adjust, [7](#)

sampleexp, [7](#)

samplesnp, [8](#)

samplesnplocus, [9](#)

saveBplot, [9](#)

seqinfo, [3](#)

sqtlfinder, [10](#)