Package 'chimera'

October 9, 2013

Type Package

Title A package for secondary analysis of fusion products

Version 1.2.6

Date 2013-06-05

Author Raffaele A Calogero, Matteo Carrara, Marco Beccuti, Francesca Cordero

Maintainer Raffaele A Calogero <raffaele.calogero@unito.it>

Depends Biobase, GenomicRanges, Rsamtools, methods, org.Hs.eg.db,org.Mm.eg.db, AnnotationDbi, BSgenome.Hsapiens.UCSC.hg19,TxDb.Hsapiens.UCSC.hg19.knownGene

Suggests BSgenome.Mmusculus.UCSC.mm9,TxDb.Mmusculus.UCSC.mm9.knownGene

Enhances Rsubread

Description This package facilitates the characterisation of fusion products events. It allows to import fusion data results from the following fusion finders: bellerophontes, deFuse, FusionFinder, FusionHunter, mapSplice, tophat-fusion, FusionMap

biocViews Infrastructure

SystemRequirements TopHat, bowtie and samtools are required for some functionalities

License Artistic-2.0

R topics documented:

`chimera-package` *A package for secondary analysis of fusion products*

Description

The package imports fusion results from tophat-fusion, mapSplice, deFuse, fusionmap, bellerophontes, fusionfinder, fusionhunter. The package is made to facilitate the characterisation of fusion products events.

Details

Package:	chimera
Type:	Package
Version:	1.0
Date:	2012-10-16
License:	Artistic-2.0

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Raffaele A Calogero Maintainer: Raffaele A Calogero <raffaele.calogero@unito.it>

References

~~ Literature or other references for background information ~~

`chimeraSeqs` *A function to generate the nucleotide sequences of a fusion event*

Description

A function generating the nucleotide sequences of a chimera.

Usage

```
chimeraSeqs(fset, extend=1000, type="transcripts")
```

Arguments

fset	A fSet object.
extend	number of nucleotides used to extend a genomic region that is not an annotated gene. Default is 1000 nts
type	Chimera can be build at transcript level, i.e. transcript level will generate multiple fusion transcripts depending of the number of transcripts associated to each of the two genes involved in the fusion.

Value

A DNAStringSet encompassing the fusions generated using all the isoforms for each gene involved in the fusion.The name of each element of the DNAStringSet has the following format: gene1-lengthOfGeneFragment:gene2-lengthOfGeneFragment. In case the fusion junction is located in an intronic sequence, a warning is provided. The presence of a partial intron in the fusion is an indication that the fusion does not generate an active chimeric peptide.

Author(s)

Raffaele A Calogero

See Also

fusionName

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
fusion.names <- fusionName(tmp)
fusion.names
myset <- tmp[[13]]
tmp.seq <- chimeraSeqs(myset, type="transcripts")
#write.XStringSet(tmp.seq, paste(sub(":","_",fusion.names[[13]]),".fa",sep=""), format="fasta")
```

---

filterList *A function to filter a list of fSet objects*

Description

A function filtering a list of fSet objects on the basis of supporting reads or fusion names.

Usage

```
filterList(x,type=c("supporting.reads","fusion.names", "intronic"),query)
```

Arguments

x	a list of fSet object
type	filtering can be performed on the basis of a minimal number of supporting reads or on the basis of fusion names
query	query is number in case supporting.reads is selected or a vector of fusion names if the case fusion.names is selected. In case type is intronic query is NULL. In the latter case fusion having one of the elements located in an intronic region are discarded

Author(s)

Raffaele A Calogero

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
fusion.names <- fusionName(tmp)
tmp1 <- filterList(tmp, type="fusion.names", fusion.names[c(1,3,7)])
tmp2 <- filterList(tmp, type="supporting.reads", 2)
```

---

fSet *Class fSet, a class represent fusion data, and methods for processing it*

Description

This is class representation for a fusion event.

Slots

fusionInfo A list

fusionLoc A GRangesList encompassing fusion locations for gene 1 and 2

fusionRNA A DNAStringSet encompassing fusion transcripts that can be generated combining all transcripts for gene 1 and 2 the sequences are generated by the chimeraSeqs and they are needed to generate a bam file with tophatRun, which maps all the reads on the transcripts involved in the fusion. The bam file can be loaded in the slot fusionsLoc of the fSetSummary object as GappedAlignments. These data can be used to plot the reads coverage on the fused transcripts.

fusionGA A GappedAlignments object encompassing positions for all reads mapping on the DNAStringSet located in fusionRNA slot

Methods

Standard generic methods:

fusionData (fSet) An accessor function used to retrieve information for a fusion

fusionGRL (fSet) An accessor function used to retrieve GRangesList encompassing fusion locations for gene 1 and 2

fusionRNA (fSet) An accessor function used to extract the DNAStringSet

addRNA(fSet, rna) An accessor function used to add the DNAStringSet to the fset object

fusionGA(fSet) An accessor function used to extract the GappedAlignments object

addGA(fSet, bam) An accessor function used to add the GappedAlignments object to the fSet object

Author(s)

Raffaele A Calogero

See Also

chimeraSeqs, tophatRun, plotCoverage

Examples

```
#creating a fusion report from output of fusionMap
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
#extracting the fusions names
fusion.names <- fusionName(tmp)
fusion.names
#extracting the fSet object ofr one of the fusions
myset <- tmp[[13]]
#constructing the fused sequence(s)
trs <- chimeraSeqs(myset, type="transcripts")
#adding the sequences to the fSet object
myset <- addRNA(myset , trs)
#extracting sequences from an fSet object
tmp.seq <- fusionRNA(myset)
#adding reads mapped on the fusion generated using tophatRun function
myset <- addGA(myset, paste(path.package(package="chimera"),"/examples/mcf7_trs_accepted_hits.bam",sep=""))
#extracting the GappedAlignments from an fSet object
ga <- fusionGA(myset)
```

---

fusionName *A function to extract fusion names for a list of fSet object*

Description

A function allowing extract fusion names from a list of fSet objects.

Usage

```
fusionName(list)
```

Arguments

list	a list of fSet object

Author(s)

Raffaele A Calogero

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
fusion.names <- fusionName(tmp)
fusion.names
```

---

fusionPeptides *A function associating to a fusion the peptides involved.*

Description

A function associate the donor and the acceptor peptides involved in the fusion.

Usage

```
fusionPeptides(fset, which.isoform=1, donor.up=200, acceptor.down=200)
```

Arguments

fset	An fSet object, encompassing a fusion including two annotated genes
which.isoform	Since each gene can be represented by multiple transcripts, one of the available transcripts need to be selected
donor.up	The number of nucleotides upstream to the donor end, to be used to extract the DNA fragment for PCR validation
acceptor.down	The number of nucleotides downstream to the acceptor start, to be used to extract the DNA fragment for PCR validation

Value

A list encompassing:

selected.alignment	the alignment used to define the position of the donor end on the transcript
transcrip1	the nucleotide sequence of the donor fusion transcript
pep1	the peptide sequence encoded by the donor transcript

frame.pep1	The frame of the encoded peptide
transcrip2	the nucleotide sequence of the acceptor fusion transcript
selected.alignment2	the alignment used to define the position of the acceptor start on the transcript
pep2	the peptide sequence encoded by the acceptor transcript
frame.pep2	The frame of the encoded peptide
fusion.status	if peptides are in frame on the basis of the fused mRNAs. IMPORTANT: this information os only based on fusion junction information, if indels are present results might be different. sequencign of the validation.seq, see below are mandatory to confirm the fusion event.
validation.seq	The DNA fragment encompassing the fusion to be used to design PCR primers for sanger sequencing
junction.ga	The GappedAlignment object describing the reads encompassing the fusion junction. If no genomic regions are indicated the reads spanning over the junction are missed.

Author(s)

Raffaele A Calogero

Examples

```
# #creating a fusion report from output of fusionMap
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
#extracting the fusions names
fusion.names <- fusionName(tmp)
fusion.names
#extracting the fSet object ofr one of the fusions
myset <- tmp[[13]]
#constructing the fused sequence(s)
trs <- chimeraSeqs(myset, type="transcripts")
#adding the sequences to the fSet object
myset <- addRNA(myset , trs)
#extracting sequences from an fSet object
tmp.seq <- fusionRNA(myset)
#adding reads mapped on the fusion generated using tophatRun function
myset <- addGA(myset, paste(path.package(package="chimera"),"/examples/mcf7_trs_accepted_hits.bam",sep=""))

  mypeps <- fusionPeptides(fset=myset, which.isoform=1, donor.up=200, acceptor.down=200)
```

---

importFusionData *A function to import fusion data detected by different fusion finders*

Description

A function to import in a list fusions data detected by bellerophontes, defuse, fusionfinder, fusionhunter, mapsplice, tophat-fusion, fusionmap, chimerascan.

Usage

```
importFusionData(format, filename, ...)
```

Arguments

format	bellerophontes, defuse, fusionfinder, fusionhunter, mapsplice, tophat-fusion, fusionmap, chimerascan. Format allows to select the data structure to be imported
filename	The file generated by one of the fusion finders defined in format
...	Additional parameters

Author(s)

Raffaele A Calogero

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", s
#min.support allow to retrieve only the subset of fusions supported by a user defined minimal number of junction s
 #tmp <- importFusionData("chimerascan", "edgren_cs10.bedpe", min.support=10, org="hs")
```

---

plotCoverage *A function to plot the coverage of a fusion gene*

Description

A function to plot the coverage of a fusion gene.

Usage

```
plotCoverage(fset, plot.type=c("exons","junctions"), junction.spanning=20, fusion.only=FALSE, xlab="r
```

Arguments

fset	A fSet object
plot.type	exons plot exons coverage as junctions plot coverage of junction between exons
junction.spanning	number of nucleotides located upstream and downstream the junction/fusion location
fusion.only	if TRUE only fusion coverage is plotted
xlab	x-axis label
ylab	y-axis label
main	Plot title
col.box1	color of the box describing the first gene
col.box2	color of the box describing the second gene
ybox.lim	y range defining the height of the box representing the exons

Author(s)

Raffaele A Calogero

See Also

fusionName, chimeraSeqs

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
fusion.names <- fusionName(tmp)
fusion.names
myset <- tmp[[13]]
trs <- chimeraSeqs(myset, type="transcripts")
myset <- addRNA(myset , trs)
tmp.seq <- fusionRNA(myset)
myset <- addGA(myset, paste(path.package(package="chimera"),"/examples/mcf7_trs_accepted_hits.bam",sep=""))
ga <- fusionGA(myset)
plotCoverage(myset, plot.type="exons", col.box1="red", col.box2="green", ybox.lim=c(-4,-1))
plotCoverage(myset, plot.type="junctions", col.box1="red", col.box2="yellow", ybox.lim=c(-4,-1))
plotCoverage(myset, fusion.only=TRUE, col.box1="red", col.box2="yellow", ybox.lim=c(-4,-1))
```

---

subreadRun *A function to generate a bam file for fusions coverage evaluation*

Description

A function mapping reads to a chimera sequence set. The bam produced by this remapping on a putative fusion will be used to plot the coverage data for all the fused constructs. The function uses Rsubread aligner.

Usage

```
subreadRun(ebwt,input1, input2, outfile.prefix="accepted_hits", alignment=c("se","pe"),cores=1)
```

Arguments

ebwt	Full path and name of the fasta file of one of the set of fusions of interest, to be used to build the index database. The fusion nucleotide sequences can be generated with the function chimeraSeqs
input1	The R1 fastq of a pair-end
input2	The R2 fastq of a pair-end
outfile.prefix	Prefix of the output bam file. Default is accepted_hits
alignment	se means that both fastq files from the pair-end run are concatenated, pe means that tophat will use fastq files in pair-end mode
cores	Number of cores to be used by the aligner

Value

Standard bam file output. The bam file name by default is accepted_hits.bam.

Author(s)

Raffaele A Calogero

See Also

chimeraSeqs

Examples

```
 if(require(Rsubread)){
subreadRun(ebwt=paste(find.package(package="chimera"),"/examples/SULF2_ARFGEF2.fa",sep=""),
   input1=paste(find.package(package="chimera"),"/examples/mcf7_sample_1.fq",sep=""),
   input2=paste(find.package(package="chimera"),"/examples/mcf7_sample_2.fq",sep=""),
   outfile.prefix="accepted_hits", alignment="se", cores=1)
 }
```

---

supportingReads *A function to extract supporting reads values from a list of fSet object*

Description

A function extracting supporting reads values from a list of fSet objects.

Usage

```
supportingReads(list)
```

Arguments

list	a list of fSet objects

Author(s)

Raffaele A Calogero

Examples

```
tmp <- importFusionData("fusionmap", paste(find.package(package="chimera"),"/examples/mcf7.FMFusionReport", sep=
supporting.reads <- supportingReads(tmp)
supporting.reads
```

tophatInstallation *A function to download tophat, bowtie and samtools*

Description

A function allowing the download and installation of tophat, bowtie and samtools in chimera package folder. The function also creates soft links in the user bin folder to allow the call of the above mentioned programs.

Usage

```
tophatInstallation(binDir, os=c("unix","mac"))
```

Arguments

binDir	The user bin folder
os	The supported operating systems

Author(s)

Raffaele A Calogero

Examples

```
#tophatInstallation(binDir="/somewhere/inyourpc/bin", os="mac")
```

tophatRun *A function to generate a bam file for fusions coverage evaluation*

Description

A function mapping reads to a chimera sequence set. The bam produced by this remapping on a putative fusion will be used to plot the coverage data for all the fused constructs. The function assumes that tophat is installed and located in the path. To run TopHat a softlink to bowtie or bowtie2 need to located in the user bin dir

Usage

```
tophatRun(input1, input2, output,cores=1, bowtie= c("bowtie","bowtie2"), tophat= "tophat",ebwt=paste(
```

Arguments

input1	The R1 fastq of a pair-end
input2	The R2 fastq of a pair-end
output	Folder in which tophat will generate the output
cores	number of cores to be used by tophat with program name, e.g. /somewhere/tophat
bowtie	selecting bowtie or bowtie2 aligner
tophat	full path of tophat
ebwt	full path and name of the fasta file of one of the set of fusions of interest, to be used to build the bowtie database. The fusion nucleotide sequences was generated with the function chimeraSeqs
alignment	se means that both fastq files from the pair-end run are concatenated, pe means that tophat will use fastq files in pair-end mode

Value

TopHat standard output. The bam file of interest is accepted_hits.bam. The bam file will be then loaded in the slot fusionsLoc of the fSetSummary object from which fusions were retrieved.

Author(s)

Raffaele A Calogero

See Also

chimeraSeqs

Examples

```
#tophatRun(input1=paste(find.package(package="chimera"),"/examples/mcf7_sample_1.fq",sep=""), input2=paste(find
```

Index