Package 'flipflop'

October 9, 2015

Title Fast lasso-based isoform prediction as a flow problem

Version 1.6.1

Date 2013-09-26

Author Elsa Bernard, Laurent Jacob, Julien Mairal and Jean-Philippe Vert

Maintainer Elsa Bernard <elsa.bernard@mines-paristech.fr>

Description Flipflop discovers which isoforms of a gene are expressed in a given sample together with their abundances, based on RNA-Seq read data. It takes an alignment file in SAM format as input. It can also discover transcripts from several samples simultaneously, increasing statistical power.

License GPL-3

LazyLoad yes

Imports methods, Matrix, IRanges, GenomicRanges, parallel

Suggests GenomicFeatures

SystemRequirements GNU make

Depends R (>= 2.10.0)

NeedsCompilation yes

BuildVignettes true

URL http://cbio.ensmp.fr/flipflop

biocViews RNASeq, RNASeqData, AlternativeSplicing, Regression

R topics documented:

flipflop *Estimate isoform compositions and abundances*

Description

This function takes count data (RNA-seq alignment in SAM format) for a given gene as input and estimates which isoforms of the gene are most likely to have generated this set of counts. It is based on a Poisson likelihood penalized by an l1 norm as explained in Bernard et al., 2013.

Usage

```
flipflop(data.file, out.file="FlipFlop_output.gtf", output.type="gtf", annot.file="",
  samples.file="", mergerefit=FALSE, paired=FALSE, frag=400, std=20,
  OnlyPreprocess=FALSE, preprocess.instance="", minReadNum=40, minFragNum=20,
  minCvgCut=0.05, minJuncCount=1, sliceCount=1, mc.cores=1, NN="", verbose=0,
  verbosepath=0, cutoff=1, BICcst=50, delta=1e-07, use_TSSPAS=0, max_isoforms=10)
```

Arguments

data.file	[input] Input alignment file in SAM format. The SAM file must be sorted according to chromosome name and starting position. If you use a multi-sample strategy, please give a single SAM file that results from the "samtools merge" command with the "-r" option (i.e attach RG tag infered from file name).
out.file	[output] Output gtf file storing the structure of the transcripts which are found to be expressed together with their abundances (in FPKM and expected count).
output.type	[output] Type of output when using several samples simultaneously. When equal to "gtf" the output corresponds to a gtf file per sample with specific abundances. When equal to "table" the output corresponds to a single gtf file storing the structure of the transcripts, and an associated table with the abundances of all samples (transcripts in rows and samples in columns) Default "gtf".
annot.file	[input optional] Optional annotation file in BED12 format. If given, exon boundaries will be taken from the annotations. The BED file should be sorted according to chromosome name and starting position of transcripts.
samples.file	[multi-samples] Optional samples file (one line per sample name). The names should be the one present in the RG tag of the provided SAM file.
mergerefit	[multi-samples] If TRUE use a simple refit strategy instead of the group-lasso. Default FALSE.
paired	[paired] Boolean for paired-end reads. If FALSE your reads will be considered as single-end reads. Default FALSE.
frag	[paired] Mean fragment size. Only used if paired is set to TRUE. Default 400.
std	[paired] Standard deviation of fragment size. Only used if paired is set to TRUE. Default 20.
OnlyPreprocess	[pre-processing] Boolean for performing only the pre-processing step. Output is two files: one file '.instance' and one other file '.totalnumread'. Default FALSE.

`preprocess.instance`
: [pre-processing] Give directly the pre-processed '.instance' input file created when using the OnlyPreprocess option. If non empty, the data.file and annot.file fields are ignored.

`minReadNum`
: [pre-processing] The minimum number of clustered reads to output. Default 40. If you give an annotation file it will be the minimum number of mapped reads to process a gene.

`minFragNum`
: [pre-processing] The minimum number of mapped read pairs to process a gene. Only used if paired is TRUE. Default 20.

`minCvgCut`
: [pre-processing] The fraction for coverage cutoff, should be between 0-1. A higher value will be more sensitive to coverage discrepancies in one gene. Default 0.05.

`minJuncCount`
: [pre-processing] The minimum number of reads to consider a junction as valid. Default 1.

`NN`
: [pre-processing] Total number of mapped fragments. Optional. If given the number of mapped fragments is not read from the '.totalnumread' file.

`sliceCount`
: [parallelization] Number of slices in which the pre-processing '.instance' file is cut. It creates several instance files with the extension '_jj.instance' where jj is the number of the slice. If you set OnlyPreprocess to TRUE, it will create those slices and you can run FlipFlop independently afterwards on each one of the slice. Default 1.

`mc.cores`
: [parallelization] Number of cores. If you give sliceCount>1 with OnlyPreprocess set to FALSE, it will distribute the slices on several cores. If you give a preprocess.instance file as input (which might be a slice of an original instance file), it will use several cores when you are using a multi-samples strategy. Default 1.

`verbose`
: [verbosity] Verbosity. Default 0 (little verbosity). Put 1 for more verbosity.

`verbosepath`
: [verbosity] Verbosity of the optimization part. Default 0 (little verbosity). Put 1 for more verbosity. Mainly used for de-bugging.

`cutoff`
: [parameter] Do not report isoforms whose expression level is less than cutoff percent of the most expressed transcripts. Default 1.

`BICcst`
: [parameter] Constant used for model selection with the BIC criterion. Default 50.

`delta`
: [parameter] Loss parameter, Poisson offset. Default 1e-7.

`use_TSSPAS`
: Do we restrict the candidate TSS and PAS sites. 1 is yes and 0 is no. Default 0 i.e each exon can possibly starts or ends an isoform.

`max_isoforms`
: Maximum number of isoforms given during regularization path. Default 10.

Value

A `list` with the following elements:

`transcripts`
: A list storing the structure of the expressed isoforms. The list is a GenomicRangesList object from the GenomicRanges package. Rows correspond to

exons. On the left hand side each exon is described by the gene name, the chromosome, its genomic position on the chromosome and the strand. Transcripts are described on the right hand side. Every transcript is a binary vector where an exon is labelled by 1 if it is included in the transcript.

`abundancesFPKM` A list storing the abundances of the expressed isoforms in FPKM unit. Each element of the list is a vector whose length is the number of expressed transcripts listed in the above 'transcripts' object.

`expected.counts`
A similar list as 'abundancesFPKM' but storing the expected fragment counts for each expressed isoforms.

`timer` A vector with the computation time in seconds for each gene.

Author(s)

Elsa Bernard, Laurent Jacob, Julien Mairal, Jean-Philippe Vert

Examples

```
## Load the library
library(flipflop)

## Alignment data file in SAM format
data.file <- system.file('extdata/vignette-sam.txt', package='flipflop')

## Run flipflop
ff.res <- flipflop(data.file=data.file,
                   out.file='FlipFlop_output_example.gtf')

## Names of the result list returned by flipflop
names(ff.res)

## Structure of the expressed isoforms for the first gene
## Rows correspond to exons, with chromosome, genomic position and strand information for each exon
## The metadata columns correspond to the expressed transcripts
transcripts <- ff.res$transcripts[[1]]
print(transcripts)

## Abundances in FPKM of the expressed isoforms for the first gene
## The length of the vector corresponds to the number of transcripts listed in the 'transcripts' object
## Each element of the vector is the estimated abundance of the corresponding transcript
abundancesFPKM <- ff.res$abundancesFPKM[[1]]
print(abundancesFPKM)

## Expected 'raw' counts of each expressed isoforms for the first gene
expected.counts <- ff.res$expected.counts[[1]]
print(expected.counts)
```

Index