Package 'MiClip'

December 16, 2012

Type Package

Title A Model-based Approach to Identify Binding Sites in CLIP-Seq Data

Version 1.0

Date 2012-11-15

Author Tao Wang

Maintainer Tao Wang <tao.wang@utsouthwestern.edu>

Depends R (>= 1.13.0), moments, VGAM

Description Cross-linking immunoprecipitation coupled with high-throughput sequencing (CLIP-Seq) has made it possible to identify targeting sites of RNA-binding proteins in various cell culture systems and tissue types on a genome-wide scale. Here we present MiClip,a novel model-based approach to identify high-confidence protein-RNA binding sites in CLIP-Seq datasets. This approach assigns confidence value to each binding site on a probabilistic basis. The MiClip package can be flexibly applied to analyze both HITS-CLIP data and PAR-CLIP data.

License GPL-2

R topics documented:

MiClip — *A Model-based Approach to Identify Binding Sites in CLIP-Seq Data*

Description

Construct a "MiClip" class object for following analysis.

Usage

```
MiClip(file="",mut.type="T->C",step=5,max.hmm=100,paired=F,suffix=NULL,
  empirical="auto",model.cut=0.2,max.iterats=20,conver.cut=0.01)
```

Arguments

`file` The file name (may include path name) of the mapped tag file. `file` can be only in SAM format and basespace. The package can work on both single-end and paired-end datsets.

`mut.type` The marker mutation for the CLIP-Seq experiment, separated by ",", e.g. "T->C", "T->C,T->A" or "T->C,Ins,Del". "T->C" denotes T-to-C substitution, "Ins" denotes insertion of any length and "Del" denotes deletion of any length. The default is "T->C". If `mut.type` is set to "all", all kinds of mutations are included as marker mutation.

`paired` Whether the sequencing data is paired-end. Default is FALSE.

`suffix` The suffix of the paired-end read data. This is a vector which contains the suffix of the names of forward reads and backward reads. For example, if the mate pairs in the SAM file are named as "1_2_100708_26_788_F3", "1_2_100708_26_788_F5-RNA", etc, `suffix` can either be `c("F3","F5-RNA")` or `c("_F3","_F5-RNA")`. Default is NULL and will be set automatically to `c("1","2")` if `paired` is TRUE but `suffix` is not set.

`step` In the first HMM, all clusters will be divided into bins of the same length of `step` bp and HMM will work to distinguish enriched bins from non-enriched ones.

`max.hmm` The maximum number of reads in a bin or on a base. This is used to keep calculation within the dynamic range of R. If this number is too large, probability values which are very small will become zero.

`empirical` A parameter used in model fitting in the first HMM. Default is "auto" which lets the algorithm decides its value. It can be set to the estimated minimal number of overlapping tags for a reliable CLIP cluster if default does not work. A higher value will lead to more conservative estimation.

`model.cut` The cutoff for fitting the mixture model in the second HMM. It can be set to the estimated minimal proportion of mutation tags vs. total tags for a binding site to be reliable. Larger values will lead to more conservative predictions. It should be between 0 and 1.

`max.iterats` The maximum number of iterations allowed for both HMM iterations.

`conver.cut` The cutoff for reaching convergence

Details

The function `MiClip` takes all the necessary parameters for calculation and constructs the initial `MiClip` class object.

Value

An object of class `MiClip` is returned.

`file` The file name (may include path name) of the alignment file.

`mut.type` The type of mutation wanted.

`paired` Whether the sequencing data is paired-end.

`suffix` The suffix of the paired-end read data.

`step` Bin length.

`max.hmm` The maximum number of reads in a bin or on a base.

`empirical` A parameter used in model fitting in the first HMM

`model.cut` The cutoff for fitting the mixture model in the second HMM.

`max.iterats` The maximum number of iterations allowed for both HMM iterations.

`conver.cut` The cutoff for reaching convergence

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `summary.MiClip`

Examples

```
library("MiClip")

test=MiClip(file="MiClip/doc/test.sam")

# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))

test=MiClip.read(test) # read raw data
test=MiClip.enriched(test,quiet=FALSE) # identify enriched regions
test=MiClip.binding(test,quiet=TRUE) # identify binding sites

enriched=test$enriched # test is a list of 3 components
sites=test$sites
clusters=test$clusters

summary(test) # print summary
```

`MiClip.adaptor` *Trim 3' adaptor*

Description

This helper function will remove 3' adaptors from raw reads in the sequence file.

Usage

```
## S3 method for class 'adaptor'
MiClip(file="",format="fastq",adaptor="",min=15,mismatch=0.4)
```

Arguments

`file` The filename (including full path name) of the sequencing file.

`format` The format of the sequencing file. It can be either "fastq" or "fasta". Also the raw sequencing file must be in basespace.

`adaptor` The adaptor sequence, for example "TCGTATGCCGTCTTCTGCTTG". "N" is allowed, and it is case insensitive. So "TCGTNNGCCGTCttcnncttg" is also ok.

`min` After trimming, if a sequence is shorter than `min`, it will be tossed away.

`mismatch` The maximum proportion of mismatches allowed when aligning adaptor sequence to the 3' end.

Details

This function is a wrapper function of a perl script. It trimms a full or partial 3' adaptor from each sequencing read and generates a new file in the same folder of the original sequencing file. For example, if `adaptor` is "TCGTATGCCGTCTTCTGCTTG", `min` is 15 and `mismatch` is 0.25, "NNTGGAGGCCGGACGCTTCCNAAANNNGTATGTCGT" will be trimmed down to "NNTGGAGGCCGGACGCTTCCNAAAN". There is only one mismatch in the partial adaptor sequence "NNGTATGTCGT" and 1/11<0.25, so this part will be trimmed from the short read. The adaptor at the 5' end usually won't be sequenced. Even if part of the 5' end adaptor is sequenced, such cases are usually rare. So 5' end adaptor is not considered in this function. If the user would like to remove 5' end adaptor too, please refer to other specialized adaptor removing algorithm.

Examples

```
library("MiClip")

MiClip.adaptor(file="MiClip/doc/test.fastq",
  adaptor="TGGAATTCTCGGGTGCCAAGGAACTCCAGTCAC")
```

`MiClip.binding` *Identify binding sites*

Description

This function implements the second HMM and tries to identify binding sites within enriched bins.

Usage

```
## S3 method for class 'binding'
MiClip(mic,quiet=FALSE)
```

Arguments

`mic` `mic` is an ojbect of class "MiClip" returned by `MiClip.enriched`.

`quiet` Whether the intermediate messages should be printed.

Details

The function `MiClip.binding` will first expand all adjacent enriched bins into single base pairs and then concatenate neighboring sites. So one cluster may contain multiple enriched segments, although this is rare. Then `MiClip.binding` employs HMM algorithm and Viterbi algorithm to infer true binding sites. The output is stored in `sites`.

Value

An object of class `MiClip` is returned.

`enriched` The output of the first HMM as a data frame. `region_id` is the id number generated for each cluster. `chr`, `strand`, `start` and `end` specify the genomic location of each bin. `tag` is the rounded average tag count in each bin. `enriched` and `probability` are the inference results.

`sites` The output of the second HMM as a data frame. `region_id` is the id number generated for the cluster where each base resides in. `sub_region_id` is the id number of the concatenated segment within enriched clusters. `chr`, `strand` and `pos` specify the genomic location of each base. `tag` is the read count on each base and `mutant` is the mutant count on each base. `sites` and `probability` are the inference results.

`clusters` The summary of results for all CLIP clusters. `clusters` contains information of chromosome, strand, start position, end position, whether or not contains enriched bins and whether or not contains binding sites.

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `summary.MiClip`

Examples

```
library("MiClip")

test=MiClip(file="MiClip/doc/test.sam")

# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))

test=MiClip.read(test) # read raw data
test=MiClip.enriched(test,quiet=FALSE) # identify enriched regions
test=MiClip.binding(test,quiet=TRUE) # identify binding sites

enriched=test$enriched # test is a list of 3 components
sites=test$sites
clusters=test$clusters

summary(test) # print summary
```

`MiClip.enriched` *Identify enriched bins*

Description

This function implements the firstHMM and tries to identify enriched bins within CLIP clusters.

Usage

```
## S3 method for class 'enriched'
MiClip(mic,quiet=FALSE)
```

Arguments

mic	mic is an ojbect of class "MiClip" returned by MiClip.read.
quiet	Whether the intermediate messages should be printed.

Details

The function MiClip.enriched will first divide each cluster into bins of length of step bp and then calculate the average tag coverage in each bin. Then it employs HMM algorithm and Viterbi algorithm to infer enriched bins. The output is stored in enriched.

Value

An object of class MiClip is returned.

raw	The raw data matrix including chromosomes, strands, positions, total read counts and mutant read counts
max.hmm	The maximum number of reads in a bin or on a base.
model.cut	The cutoff for fitting the mixture model in the second HMM.
max.iterats	The maximum number of iterations allowed for both HMM iterations.
conver.cut	The cutoff for reaching convergence
enriched	The output of the first HMM as a data frame. region_id is the id number generated for each cluster. chr, strand, start and end specify the genomic location of each bin. tag is the rounded average tag count in each bin. enriched and probability are the inference results.

See Also

MiClip.read, MiClip.enriched, MiClip.binding, summary.MiClip

Examples

```
library("MiClip")

test=MiClip(file="MiClip/doc/test.sam")

# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))

test=MiClip.read(test) # read raw data
test=MiClip.enriched(test,quiet=FALSE) # identify enriched regions
test=MiClip.binding(test,quiet=TRUE) # identify binding sites

enriched=test$enriched # test is a list of 3 components
sites=test$sites
clusters=test$clusters

summary(test) # print summary
```

`MiClip.read` *Read raw sequencing data*

Description

Read the sequencing data and form CLIP clusters by overlapping.

Usage

```
## S3 method for class 'read'
MiClip(mic)
```

Arguments

`mic`	`mic` is an ojbect of class "MiClip" returned by `MiClip`

Details

The function `MiClip.read` calls embeded perl scripts to read SAM format file and extract mutation information. Then CLIP clusters are formed from reads that can overlap by at least 1 bp. Reads that cannot be overlapped with any other reads are discarded.

Value

An object of class `MiClip` is returned.

`raw`	The raw data matrix including chromosomes, strands, positions, total read counts and mutant read counts
`max.hmm`	The maximum number of reads in a bin or on a base.
`empirical`	A parameter used in model fitting in the first HMM
`model.cut`	The cutoff for fitting the mixture model in the second HMM.
`max.iterats`	The maximum number of iterations allowed for both HMM iterations.
`conver.cut`	The cutoff for reaching convergence

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `summary.MiClip`

Examples

```
library("MiClip")

test=MiClip(file="MiClip/doc/test.sam")

# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))

test=MiClip.read(test) # read raw data
test=MiClip.enriched(test,quiet=FALSE) # identify enriched regions
test=MiClip.binding(test,quiet=TRUE) # identify binding sites

enriched=test$enriched # test is a list of 3 components
```

```
sites=test$sites
clusters=test$clusters

summary(test) # print summary
```

---

summary.MiClip *Summary of MiClip Inference Results*

Description

This summary function computes simple statistics for the results produced by `MiClip.binding`.

Usage

```
## S3 method for class 'MiClip'
summary(mic,...)
```

Arguments

`mic`	`mic` is an ojbect of class "`MiClip`" returned by `MiClip.enriched` or `MiClip.binding`.
`...`	further arguments passed to or from other methods.

Details

This function will compute summary statistics only if `mic` is generated from `MiClip.binding`.

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `summary.MiClip`, `summary`

Examples

```
library("MiClip")

test=MiClip(file="MiClip/doc/test.sam")

# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))

test=MiClip.read(test) # read raw data
test=MiClip.enriched(test,quiet=FALSE) # identify enriched regions
test=MiClip.binding(test,quiet=TRUE) # identify binding sites

enriched=test$enriched # test is a list of 3 components
sites=test$sites
clusters=test$clusters

summary(test) # print summary
```

Index