Package 'MiClip'

July 2, 2014

Type Package

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

Version 1.2

Date 2013-11-15

Author Tao Wang

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

Depends R (>= 2.15.0), 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

SystemRequirements Perl

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-16 08:44:12

R topics documented:

`Chi` *The demo dataset for* `MiClip` *package*

Description

This demo dataset is a small portion of the single-end HITS-CLIP data provided in the Chi study.

Usage

```
data(Chi)
```

Format

3 lists that are generated by different `MiClip` functions

References

Chi, S.W., Zang, J.B., Mele, A. and Darnell, R.B. (2009) Argonaute HITS-CLIP decodes microRNA-mRNA interaction maps. Nature, 460, 479-486.

`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="T2C",step=5,max.hmm=100,paired=F,suffix=NULL,
  empirical="auto",model.cut=0.2,max.iterats=20,conver.cut=0.01,background=NULL)
```

Arguments

file	The file name (may include path name) of the mapped tag file. file must be in SAM format and basespace but either single-end or paired-end mode.
mut.type	The marker mutation for the CLIP-Seq experiment, separated by ",", e.g. "T2C", "T2C,T2A" or "T2C,Ins,Del". "T2C" denotes T-to-C substitution, "Ins" denotes insertion of any length and "Del" denotes deletion of any length. The default is "T2C". 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. Default is 5 and for larger dataset(>20M mapped reads) it is better to set step to a value between 10-15.
empirical	Used to help 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 enriched 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 and the default is 0.2.
max.hmm	The maximum number of tag counts 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. Default is 100.
max.iterats	The maximum number of iterations allowed for both HMM iterations. Default is 20.
conver.cut	The cutoff for reaching convergence. Default is 0.01.
background	An optional data frame with preprocessed count data in the control experiment for normalization purpose. It must be of the same format as the raw data frame in the output of MiClip.read. The first 4 columns must be the same as raw and the corresponding columns for total tag count should be replaced with tag count in the control condition. We suggest running MiClip without the background experiment first, constructing a data frame according to the raw data frame, and then running MiClip again with that data frame.

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
`background`	background data frame

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `MiClip.sum`

Examples

```
test1=MiClip(file=system.file("extdata/test.sam",package="MiClip"),mut.type="Del")
# for paired-end data
# test=MiClip(file="test.sam",paired=TRUE,suffix=c("F3","F5-RNA"))
```

`MiClip.adaptor` *Trim 3' adaptor*

Description

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

Usage

```
MiClip.adaptor(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 "TCGTNNGCCGTCttcnnctttg" 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. It can only work on single-end reads now. 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=system.file("extdata/test.fastq",package="MiClip"),
  adaptor="TGGAATTCTCGGGTGCCAAGGAACTCCAGTCAC")
```

`MiClip.binding` *Identify binding sites*

Description

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

Usage

```
MiClip.binding(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. 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, MiClip.sum

Examples

```
data(Chi,package="MiClip")
test4=MiClip.binding(test3,quiet=TRUE) # identify binding sites
```

MiClip.enriched *Identify enriched bins*

Description

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

Usage

```
MiClip.enriched(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`, `MiClip.sum`

Examples

```
data(Chi,package="MiClip")
test3=MiClip.enriched(test2,quiet=FALSE) # identify enriched regions
```

MiClip.galaxy *Wrapper function for running MiClip on Galaxy*

Description

A wrapper function for running MiClip on Galaxy mirror. This function is not to be used by outside users.

Usage

```
MiClip.galaxy(file="",control=NULL,mut.type="T2C",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.
`control`	The file name of the control experiment (if available).

mut.type	The marker mutation for the CLIP-Seq experiment, separated by ",", e.g. "T2C", "T2C,T2A" or "T2C,Ins,Del". "T2C" denotes T-to-C substitution, "Ins" denotes insertion of any length and "Del" denotes deletion of any length. The default is "T2C". 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

MiClip.read *Read raw sequencing data*

Description

Read the sequencing data and form CLIP clusters by overlapping.

Usage

```
MiClip.read(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
`background`	The background data frame if available

See Also

`MiClip.read`, `MiClip.enriched`, `MiClip.binding`, `MiClip.sum`

Examples

```
test1=MiClip(file=system.file("extdata/test.sam",package="MiClip"),mut.type="Del")
test2=MiClip.read(test1) # read raw data
```

MiClip.snp *Distinguish possible SNPs from high confidence CLIP binding sites*

Description

Read the alignment data of the control experiments (e.g. RNA with no cross-linking) and mark those high confidence CLIP bindings sites which actually might be SNPs.

Usage

```
MiClip.snp(mic,file="",mut.type="T2C",paired=F,suffix=NULL)
```

Arguments

mic	mic is an ojbect of class "MiClip" returned by MiClip.binding
file	file is the alignment file of the control experiments.
mut.type	mut.type must be the same as the one used for MiClip when running the crosslinked sample.
paired	paired must be the same as the one used for MiClip when running the crosslinked sample.
suffix	suffix must be the same as the one used for MiClip when running the crosslinked sample.

Details

The function MiClip.snp is devised to distinguish possible SNPs from inferred high-confidence crosslinking sites. Optionally, users can add additional quality control steps before alignment of the control data to the reference genome. MiClip.snp takes the alignment file of the control condition and looks for the same mutations as in the treatment sample. A null hypethesis is tested on each mutant site by MiClip.snp in order to extract possible SNPs. Then the binding sites inferred by MiClip.binding are screened for these possible SNPs (mutant sites that are not inferred as binding sites are ignored). A column will be added to sites in the final output specifying whether a binding site could actually be a SNP. And another column will be added to clusters in the final output specifying those clusters, at least one of whose binding sites could be a SNP.

Value

An object of class MiClip is returned.

enriched	The same as the one returned by MiClip.binding.
sites	The same as the one returned by MiClip.binding.
clusters	The same as the one returned by MiClip.binding.
snps	This data frame is added by MiClip.snp. It contains information of all the possible SNP sites extracted from the control experiment file.

See Also

MiClip.read, MiClip.enriched, MiClip.binding, MiClip.sum

Examples

```
data(Chi,package="MiClip")
test5=MiClip.snp(test4,file=system.file("extdata/snp.sam",package="MiClip"),
mut.type="Del") # mark possible SNPs
```

MiClip.sum *Summary of MiClip Inference Results*

Description

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

Usage

```
MiClip.sum(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, MiClip.sum

Examples

```
data(Chi,package="MiClip")
MiClip.sum(test4) # print summary
```

Index