Package 'Romulus'

April 27, 2016

Title Robust Multi-State Identification of Transcription Factor Binding Sites

Version 1.0.1

Description Romulus is a computational method to accurately identify individual transcription factor binding sites from genome sequence information and cell-type--specific experimental data, such as DNase-seq. It combines the strengths of its predecessors, CENTIPEDE and Wellington, while keeping the number of free parameters in the model robustly low. The method is unique in allowing for multiple binding states for a single transcription factor, differing in their cut profile and overall number of DNase I cuts.

URL `http://github.com/ajank/Romulus`

Depends R (>= 2.10)

License GPL-3

LazyData true

Suggests knitr

VignetteBuilder knitr

Author Aleksander Jankowski [aut, cre]

Maintainer Aleksander Jankowski <ajank@mimuw.edu.pl>

R topics documented:

`fitRomulus` *Fit the Romulus model.*

Description

Fit the Romulus model.

Usage

```
fitRomulus(cuts1, cuts2, anno, priors, bins1, bins2, nbound = NA,
  PriorLik = NULL, addIntercept = T, maxIter = 100,
  maxPostProbDiff = 0.001)
```

Arguments

`cuts1`	integer matrix of forward strand DNase I cuts, with a row for each candidate binding site. The columns should correspond to the genomic locations upstream and within the candidate binding site.
`cuts2`	integer matrix of reverse strand DNase I cuts, with a row for each candidate binding site. The columns should correspond to the genomic locations within the candidate binding site and downstream.
`anno`	data frame with annotations of the candidate binding sites. The numeric columns to be used are specified in `priors`.
`priors`	character vector or list of character vectors specifying the names of columns from `anno` to be considered in the logistic regression. If a list, each item specifies the column names for each bound state separately, otherwise the same column names will be used for all the bound states.
`bins1`	integer vector or matrix specifying how the columns of `cuts1` are grouped into bins. If a matrix, each row specifies the grouping for each bound state separately, otherwise the same grouping will be used for all the bound states. The numbers specifying the grouping must form the set of the first N natural numbers (1, . . ., N) for some N.
`bins2`	integer vector or matrix specifying how the columns of `cuts2` are grouped into bins, as above.
`nbound`	optional integer, specifying the number of bound states. If not provided, the value will be guessed from the length of `anno` or the number of rows in `bins1` and `bins2`.
`PriorLik`	optional numeric matrix, with a row for each candidate binding site and a column for each bound state, containing the initial prior likelihoods. If not provided, the default initialization procedure will be used.
`addIntercept`	logical. Should an additional intercept term be included in the model?
`maxIter`	integer. Maximal number of Expectation-Maximization iterations to perform.
`maxPostProbDiff`	numeric. The Expectation-Maximization procedure will be terminated when the absolute differences in posterior probabilities between the iterations will become smaller than this value.

Details

Fit the Romulus model using an Expectation-Maximization approach.

Value

A list with the elements

`nstates`	number of states (including the unbound state).
`nbound`	number of bound states.
`bins1`	matrix specifying how the columns of `cuts1` are grouped into bins, with a row for each state.
`bins2`	matrix specifying how the columns of `cuts2` are grouped into bins, with a row for each state.
`binsizes1`	list specifying, for each state, how many columns of `cuts1` fall into each bin.
`binsizes2`	list specifying, for each state, how many columns of `cuts2` fall into each bin.
`Beta`	list of estimated logistic regression coefficients $\beta_j^{(k)}$ for each state.
`NegBinom`	matrix of estimated negative binomial parameters for each state, with columns `"NegBinomR1"` `"NegBinomR2"`, `"NegBinomP1"` and `"NegBinomP2"`, representing $r^{+(k)}$, $r^{-(k)}$, $p^{+(k)}$ and $p^{-(k)}$, respectively.
`Lambda1`	list of estimated multinomial parameters $\lambda_b^{+(k)}$ for each state.
`Lambda2`	list of estimated multinomial parameters $\lambda_b^{-(k)}$ for each state.
`LambdaReg1`	matrix of multinomial parameter estimates for forward strand DNase I cuts, smoothed by replacing the fixed-value bins by a piecewise linear function, with a row for each state.
`LambdaReg2`	matrix of multinomial parameter estimates for reverse strand DNase I cuts, as above.
`PriorProb`	matrix of prior probabilities calculated from the logistic regression component, with a row for each candidate binding site and a column for each state.
`LogLikelihood`	matrix of log-likelihoods calculated from the negative binomial and multinomial components, with a row for each candidate binding site and a column for each state.
`PostProb`	matrix of posterior probabilities, calculated from the complete model, with a row for each candidate binding site and a column for each state.

References

Jankowski, A., Tiuryn, J. and Prabhakar, S. (2016) Romulus: Robust multi-state identification of transcription factor binding sites from DNase-seq data. Bioinformatics. doi: 10.1093/bioinformatics/btw209

Examples

```
# Clip the DNase-seq data for NRSF at 99.9% quantile
thresh <- max(quantile(NRSF.cuts, 0.999), 1L)
cuts <- pmin(NRSF.cuts, thresh)

# Forward strand cuts only upstream and within the candidate binding site
cuts1 <- cuts[, 1:(ncol(cuts) / 2 - NRSF.margin)]
# Reverse strand cuts only within the candidate binding site and downstream
cuts2 <- cuts[, ncol(cuts) / 2 + (NRSF.margin + 1):(ncol(cuts) / 2)]

# Take 20 bp bins outside the candidate binding site and 1 bp bins within it
NRSF.width <- (ncol(cuts) - 4 * NRSF.margin) / 2
bins1 <- c(rep(1:10, each = 20), 10 + 1:NRSF.width)
bins2 <- c(1:NRSF.width, NRSF.width + rep(1:10, each = 20))

# Fit the Romulus model
r.fit <- fitRomulus(cuts1, cuts2, NRSF.anno, list(c("score")), bins1, bins2)


# Benchmarking of Romulus and CENTIPEDE

## Not run:
library(ROCR)
library(CENTIPEDE)
c.fit <- fitCentipede(Xlist = list(DNase = as.matrix(NRSF.cuts)),
  Y = cbind(1, NRSF.anno$score))

r.pred <- prediction(1 - r.fit$PostProb[, r.fit$nstates],
  as.integer(NRSF.anno$signalValue > 0))
r.perf <- performance(r.pred, measure = "tpr", x.measure = "fpr")
r.auc <- performance(r.pred, measure = "auc")

c.pred <- prediction(c.fit$PostPr, as.integer(NRSF.anno$signalValue > 0))
c.perf <- performance(c.pred, measure = "tpr", x.measure = "fpr")
c.auc <- performance(c.pred, measure = "auc")

plot(r.perf, col = "red",
  main = "NRSF binding predictions benchmarked using ChIP-seq data")
lines(c.perf@x.values[[1]], c.perf@y.values[[1]], col = "blue")
legend("bottomright", col = c("red", "blue"), lty = 1,
  legend = c(sprintf("Romulus, AUC = %0.4f", r.auc@y.values[[1]]),
  sprintf("CENTIPEDE, AUC = %0.4f", c.auc@y.values[[1]])))

## End(Not run)
```

`NRSF.anno` *Annotations of 4,828 NRSF (REST) motif instances.*

Description

A dataset containing the annotations of 4,828 NRSF (REST) motif instances in the human genome (hg19 assembly).

Usage

```
NRSF.anno
```

Format

A data frame with 4828 rows and 6 variables:

chrom chromosome

start starting base pair (1-based, inclusive)

end ending base pair

strand strand ("-" or "+")

score Position Weight Matrix score (log-likelihood)

signalValue ChIP-seq signal value in K562 cells

Source

Motif instances are downloaded from http://homer.salk.edu/homer/ (HOMER Known Motifs track).

ChIP-seq signal value was extracted from http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeAwgTfbsUniform/wgEncodeAwgTfbsHaibK562NrsfV0416102UniPk.narrowPeak.gz.

NRSF.cuts *DNase I cuts around 4,828 NRSF (REST) motif instances.*

Description

A dataset containing the exact numbers of DNase I cuts around 4,828 NRSF (REST) motif instances, split into forward and reverse strand cuts.

Usage

```
NRSF.cuts
```

Format

An integer matrix with 4828 rows and 838 columns. Columns 1-419 correspond to forward strand cuts (200 bp upstream + 19 bp motif site + 200 bp downstream), while columns 420-838 correspond to reverse strand cuts at the same positions.

Source

Extracted from `http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeOpenChromDnase/wgEncodeOpenChromDnaseK562AlnRep1V2.bam`.

`NRSF.margin` *Number of base pairs of margin for* `NRSF.cuts`.

Description

Number of base pairs of upstream and downstream margin for `NRSF.cuts`.

Usage

```
NRSF.margin
```

Format

Integer, equal to 200 (base pairs).