Package 'PeakSegDP'

August 15, 2017

Maintainer Toby Dylan Hocking <toby.hocking@r-project.org>

Author Toby Dylan Hocking, Guillem Rigaill

Version 2017.08.15

License GPL-3

Title Dynamic Programming Algorithm for Peak Detection in ChIP-Seq Data

Description A quadratic time dynamic programming algorithm can be used to compute an approximate solution to the problem of finding the most likely changepoints with respect to the Poisson likelihood, subject to a constraint on the number of segments, and the changes which must alternate: up, down, up, down, etc. For more info read <http://proceedings.mlr.press/v37/hocking15.html> ``PeakSeg: constrained optimal segmentation and supervised penalty learning for peak detection in count data'' by TD Hocking et al, proceedings of ICML2015.

Suggests ggplot2 (>= 2.0), testthat, penaltyLearning

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-08-15 21:39:42 UTC

R topics documented:

`calc.grad.list` *calc grad list*

Description

List of calc.grad functions: x, features, limits -> gradient.

Usage

```
"calc.grad.list"
```

`calc.loss.from.lp.list` *calc loss from lp list*

Description

if we have already calculated the linear predictor using fit$predict, this function can be useful.

Usage

```
"calc.loss.from.lp.list"
```

`calc.loss.list` *calc loss list*

Description

List of interval regression loss functions: x, feat, lim => numeric.

Usage

```
"calc.loss.list"
```

cDPA *cDPA*

Description

A constrained dynamic programming algorithm (cDPA) can be used to compute the best segmentation with respect to the Poisson likelihood, subject to a constraint on the number of segments, and the changes which must alternate: up, down, up, down, ...

Usage

```
cDPA(count, weight = rep(1, length(count)), maxSegments)
```

Arguments

count	Integer vector of count data to segment.
weight	Data weights (normally this is the number of base pairs).
maxSegments	Maximum number of segments to consider.

Author(s)

Toby Dylan Hocking, Guillem Rigaill

Examples

```
fit <- cDPA(c(0, 10, 11, 1), maxSegments=3)
stopifnot(fit$ends[3,4] == 3)
stopifnot(fit$ends[2,3] == 1)
```

chr11ChIPseq *ChIP-seq aligned read coverage for 4 samples on a subset of chr11*

Description

A ChIP-seq experiment was performed to locate the genomic positions of a histone (H3K4me3) in 2 B cell samples (McGill0091, McGill0322) and 2 T cell samples (McGill0002, McGill0004). The short sequence reads (about 100 base pairs each) were aligned to the hg19 reference genome, and the "coverage" in this data set contains the total count of aligned reads at each base pair. It also contains annotated regions determined by an expert who examined scatterplots of the coverage profiles.

Usage

```
data("chr11ChIPseq")
```

Format

A named list of 2 data.frames: regions contains annotations about which regions contain or do not contain peaks, and coverage contains the noisy signal.

Source

H3K4me3_TDH_immune chunk 5 in http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/ which in turn comes from http://epigenomesportal.ca/

Examples

```
data(chr11ChIPseq)
library(ggplot2)
ann.colors <-
  c(noPeaks="#f6f4bf",
    peakStart="#ffafaf",
    peakEnd="#ff4c4c",
    peaks="#a445ee")

if(interactive() && require(ggplot2)){

ggplot()+
  scale_fill_manual("annotation", values=ann.colors,
                    breaks=names(ann.colors))+
  penaltyLearning::geom_tallrect(aes(xmin=chromStart/1e3, xmax=chromEnd/1e3,
                    fill=annotation),
                data=chr11ChIPseq$regions, alpha=1/2)+
  theme_bw()+
  theme(panel.margin=grid::unit(0, "cm"))+
  facet_grid(sample.id ~ ., scales="free")+
  geom_step(aes(chromStart/1e3, count), data=chr11ChIPseq$coverage)+
  xlab("position on chr11 (kilo base pairs)")

}
```

---

chr11first *Counts of first base of aligned reads*

Description

For 4 samples on chr11 (hg19), this data set counts the first base pair of aligned reads at each genomic position. In contrast, chr11ChIPseq counts every base pair in each read (and each read is about 100bp, so that means there is some auto-correlation in chr11ChIPseq, but not in chr11first).

Usage

```
data("chr11first")
```

Format

A data frame with 23252 observations on the following 4 variables.

`sample.id` a factor with levels for each of 4 samples

`chromStart` integer vector: base before, on chr11

`chromEnd` integer vector: last base on chr11

`count` integer: aligned first base read counts

Source

H3K4me3_TDH_immune chunk 5 in http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/ which in turn comes from http://epigenomesportal.ca/

Examples

```
data(chr11ChIPseq)
data(chr11first)
library(ggplot2)
ann.colors <-
  c(noPeaks="#f6f4bf",
    peakStart="#ffafaf",
    peakEnd="#ff4c4c",
    peaks="#a445ee")
both <- list(coverage=chr11ChIPseq$coverage, first=chr11first)
representations <- NULL
one.sample <- "McGill0322"
for(data.type in names(both)){
  one <- subset(both[[data.type]], sample.id==one.sample)
  representations <- rbind(representations, data.frame(data.type, one))
}
one.sample.regions <- subset(
  chr11ChIPseq$regions, sample.id==one.sample)

if(interactive() && require(ggplot2)){

ggplot()+
  scale_fill_manual("annotation", values=ann.colors,
                    breaks=names(ann.colors))+
  penaltyLearning::geom_tallrect(aes(xmin=chromStart/1e3, xmax=chromEnd/1e3,
                    fill=annotation),
                data=one.sample.regions, alpha=1/2)+
  theme_bw()+
  theme(panel.margin=grid::unit(0, "cm"))+
  facet_grid(data.type ~ ., scales="free")+
  geom_step(aes(chromStart/1e3, count), data=representations)+
  xlab("position on chr11 (kilo base pairs)")

}
```

derivs *derivs*

Description

List of functions, each a derivative of a phi loss.

Usage

```
"derivs"
```

GeomTallRect *GeomTallRect*

Description

ggproto object for geom_tallrect

Usage

```
"GeomTallRect"
```

getPath *getPath*

Description

Extract endpoint matrix from cDPA result.

Usage

```
getPath(A)
```

Arguments

A

Author(s)

Toby Dylan Hocking, Guillem Rigaill

H3K36me3.AM.immune.19 *Several ChIP-seq profiles, some of which have few data points*

Description

These data are used to test the PeakSegDP algorithm, to make sure it gives sensible results, even when there are few data.

Usage

```
data("H3K36me3.AM.immune.19")
```

Format

Named list of 21 data.frames, each with columns chromStart, chromEnd, count.

Source

http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/ data set H3K36me3_AM_immune, chunk id 19

H3K36me3.TDH.other.chunk3.cluster4 *8 profiles of H3K36me3 data*

Description

these data caused a bug in multiSampleSegHeuristic.

Usage

```
data("H3K36me3.TDH.other.chunk3.cluster4")
```

Format

A data frame with 36914 observations on the following 4 variables.

`sample.id` a factor with 8 levels

`chromStart` integer vector

`chromEnd` integer vector

`count` integer vector

Source

http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db/ data set H3K36me3_TDH_other chunk 3.

H3K4me3.TDH.immune.chunk12.cluster4 *Histone ChIP-seq data, 26 samples, chr1 subset*

Description

26 samples, each with the same overlapping peak(s).

Usage

```
data("H3K4me3.TDH.immune.chunk12.cluster4")
```

Format

A data frame.

Source

http://cbio.ensmp.fr/~thocking/chip-seq-chunk-db H3K4me3_TDH_immune data set, chunk.id=12.

PeakSegDP *PeakSegDP*

Description

Compute the PeakSeg model on a data.frame of compressed sequence reads.

Usage

```
PeakSegDP(compressed, maxPeaks)
```

Arguments

compressed	data.frame with columns chromStart, chromEnd, count.
maxPeaks	maximum number of peaks to consider.

Author(s)

Toby Dylan Hocking, Guillem Rigaill

Examples

```
library(PeakSegDP)
data(chr11ChIPseq, envir=environment())
one <- subset(chr11ChIPseq$coverage, sample.id=="McGill0002")[10000:12000,]
fit <- PeakSegDP(one, 3L)

if(interactive() && require(ggplot2)){

  ggplot()+
    geom_step(aes(chromStart/1e3, count), data=one)+
    geom_segment(aes(chromStart/1e3, mean,
                     xend=chromEnd/1e3, yend=mean),
                 data=fit$segments, color="green")+
    geom_segment(aes(chromStart/1e3, 0,
                     xend=chromEnd/1e3, yend=0),
                 data=subset(fit$segments, status=="peak"),
                 size=3, color="deepskyblue")+
    theme_bw()+
    theme(panel.margin=grid::unit(0, "cm"))+
    facet_grid(peaks ~ ., scales="free", labeller=function(df){
      s <- ifelse(df$peaks==1, "", "s")
      df$peaks <- paste0(df$peaks, " peak", s)
      df
    })

}
```

`phi.list` *phi list*

Description

List of functions, each a phi loss.

Usage

```
"phi.list"
```

`PoissonLoss` *PoissonLoss*

Description

Compute the weighted Poisson loss function, which is seg.mean - count * log(seg.mean). The edge case is when the mean is zero, in which case the probability mass function takes a value of 1 when the data is 0 (and 0 otherwise). Thus the log-likelihood of a maximum likelihood segment with mean zero must be zero.

Usage

```
PoissonLoss(count, seg.mean, weight = 1)
```

Arguments

```
count
seg.mean
weight
```

Author(s)

Toby Dylan Hocking, Guillem Rigaill

Examples

```
PoissonLoss(1, 1)
PoissonLoss(0, 0)
PoissonLoss(1, 0)
PoissonLoss(0, 1)
```

---

regression.funs *regression funs*

Description

List of regression functions: features, limits -> list.

Usage

```
"regression.funs"
```

Index