Package 'FLOPART'

January 20, 2025

Type Package

Title Functional Labeled Optimal Partitioning

Version 2024.6.19

Description Provides an efficient 'C++' code for computing an
optimal segmentation model
with Poisson loss,
up-down constraints,
and label constraints,
as described by Kaufman et al. (2024) <doi:10.1080/10618600.2023.2293216>.

License GPL-3

RoxygenNote 7.3.1

Suggests testthat, PeakError, knitr, markdown, ggplot2

Depends R (>= 2.10)

LinkingTo Rcpp

Imports Rcpp, data.table

VignetteBuilder knitr

NeedsCompilation yes

Author Toby Dylan Hocking [aut, cre]

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

Repository CRAN

Date/Publication 2024-06-20 21:30:10 UTC

Contents

`FLOPART` *Functional Labeled Optimal Partitioning*

Description

Main function for computing optimal segmentation model with Poisson loss, up-down constraints, and label constraints.

Usage

```
FLOPART(coverage, label, penalty)
```

Arguments

`coverage`	data frame of coverage
`label`	data frame of labels
`penalty`	non-negative penalty constant

Value

list with named elements: coverage_dt is a data table with columns chromStart, chromEnd, count, weight; label_dt is a data table with columns chromStart, chromEnd, annotation, type, firstRow, lastRow; cost_mat is a Nx2 numeric matrix of optimal penalized Poisson loss values up to each data point and in each state; intervals_mat is a Nx2 integer matrix of counts of intervals used to store the optimal cost function, useful for analyzing time/space complexity; segments_dt is a data table with columns chromStart, chromEnd, status, mean.

Author(s)

Toby Dylan Hocking

Examples

```
library(data.table)
data("Mono27ac.simple", package="FLOPART")
Mono27ac.simple
label.pen <- 1400
fit <- with(Mono27ac.simple, FLOPART::FLOPART(coverage, label, label.pen))
lapply(fit, head)

## Plot data and model.
ann.colors <- c(
  noPeaks="orange",
  peakStart="#efafaf",
  peakEnd="#ff4c4c")
model.color <- "blue"
(peaks.dt <- fit[["segments_dt"]][status=="peak"][, peak.y := -2][])
if(require("ggplot2")){
  ggplot()+
    ggtitle("Model with label constraints (FLOPART)")+
    scale_fill_manual("label", values=ann.colors)+
    geom_rect(aes(
      xmin=chromStart, xmax=chromEnd,
      ymin=-Inf, ymax=Inf,
      fill=annotation),
      alpha=0.5,
      color="grey",
      data=Mono27ac.simple[["label"]])+
    geom_step(aes(
      chromStart, count),
      data=Mono27ac.simple[["coverage"]],
      color="grey50")+
    geom_step(aes(
      chromStart, mean),
      data=fit[["segments_dt"]],
      color=model.color)+
    geom_segment(aes(
      chromStart, peak.y,
      xend=chromEnd, yend=peak.y),
      color=model.color,
      size=1,
      data=peaks.dt)+
    geom_point(aes(
      chromEnd, peak.y),
      color=model.color,
      shape=21,
      fill="white",
      data=peaks.dt)+
    theme_bw()+
    theme(panel.spacing=grid::unit(0, "lines"))
}

## To analyze computational complexity, plot number of intervals
## stored in cost function, versus data point, for each cost status.
imat <- fit[["intervals_mat"]]
interval.dt <- data.table(
  intervals=as.integer(imat),
  status=c("peak", "background")[as.integer(col(imat))],
  data.i=as.integer(row(imat)))
if(require("ggplot2")){
  ggplot()+
    scale_fill_manual("label", values=ann.colors)+
    geom_rect(aes(
      xmin=firstRow-0.5, xmax=lastRow+0.5,
      ymin=-Inf, ymax=Inf,
      fill=annotation),
      alpha=0.5,
      color="grey",
      data=fit[["label_dt"]])+
    geom_line(aes(
      data.i, intervals, color=status),
      size=1,
      data=interval.dt)
}
```

`FLOPART_data` *Convert data for input to FLOPART*

Description

FLOPART needs at most one label per coverage data row, which may not be the case for arbitrary coverage/labels.

Usage

```
FLOPART_data(coverage, label)
```

Arguments

`coverage`	data frame of coverage with columns chromStart, chromEnd, count
`label`	data frame of labels with with columns chromStart, chromEnd, annotation

Value

named list: coverage_dt is data table representing a run-length encoding of the input coverage data, with additional rows if there are label chromStart/chromEnd values not in the set of coverage positions; label_dt is a data table with one row per label, and additional columns firstRow/lastRow which refer to row numbers of coverage_dt, 0-based for passing to C++ code.

Author(s)

Toby Dylan Hocking

Examples

```
library(data.table)
d <- function(chromStart, chromEnd, count){
  data.table(chromStart, chromEnd, count)
}
(cov.dt <- rbind(
  d(0, 10, 53),
  d(10, 20, 124)))
l <- function(chromStart, chromEnd, annotation){
  data.table(chromStart, chromEnd, annotation)
}
lab.dt <- rbind(
  l(2, 7, "noPeaks"),
  l(8, 15, "peakStart"))
FLOPART::FLOPART_data(cov.dt)
FLOPART::FLOPART_data(cov.dt, lab.dt)
```

```
data("Mono27ac", package="FLOPART")
sapply(Mono27ac, dim)
converted <- with(Mono27ac, FLOPART::FLOPART_data(coverage, labels))
sapply(converted, dim)
```

`FLOPART_interface` *Interface to FLOPART C++ code*

Description

Interface to FLOPART C++ code

Usage

```
FLOPART_interface(
  data_vec,
  weight_vec,
  penalty,
  label_type_vec,
  label_start_vec,
  label_end_vec
)
```

Arguments

`data_vec`	Integer vector of non-negative count data
`weight_vec`	Numeric vector of positive weights (same size as data_vec)
`penalty`	non-negative real-valued penalty (larger for fewer peaks)
`label_type_vec`	Integer vector of label types
`label_start_vec`	Integer vector of label starts
`label_end_vec`	Integer vector of label ends

Value

List with named elements: cost_mat and intervals_mat (one row for each data point, first column up, second down), segments_df (one row for each segment in the optimal model)

`get_label_code` *Lookup the integer values used to represent different label types*

Description

Lookup the integer values used to represent different label types

Usage

```
get_label_code()
```

Value

Integer vector with names corresponding to supported label types

`Mono27ac` *H3K27ac ChIP-seq data from one Monocyte sample*

Description

Raw coverage data and labels are provided in order to test the FLOPART algo.

Usage

```
data("Mono27ac")
```

Format

List of 2 data.tables, coverage and labels.

Source

https://github.com/tdhock/feature-learning-benchmark

`Mono27ac.simple` *Smaller H3K27ac ChIP-seq data from one Monocyte sample*

Description

Raw coverage data and labels are provided in order to test the FLOPART algo.

Usage

```
data("Mono27ac.simple")
```

Format

List of two data tables, coverage and label.

Source

https://github.com/tdhock/feature-learning-benchmark

Index