Hard Test

Garrett Tetrault

3/5/2019

```
library(dplyr)
library(ggplot2)
library(reshape2)
library(microbenchmark)
library(changepoint)
library(fpop)
data(neuroblastoma, package = "neuroblastoma")
```

Get relevant data

```
# Set chromosome we are testing on and number of
# tests we are going to run.
chr <- "1"
num_tests <- 200
profile <- filter(neuroblastoma$profiles,
                  chromosome == chr)

id <- unique(neuroblastoma$profiles$profile.id)[1:num_tests]
```

Benchmark changepoint algorithms

```
# Preallocate data vectors.
prof_length <- vector(length=num_tests)
cpt_time <- vector(length=num_tests)
fpop_time <- vector(length=num_tests)

for(i in 1:num_tests) {
  prof_id <- filter(profile, profile.id == id[i])
  prof_length[i] <- length(prof_id$logratio)

  # Benchmark changepoint functions and record
  # mean time taken.
  cpt_bench <-
    microbenchmark(
      cpt.mean(prof_id$logratio, method="PELT")
    )
  cpt_time[i] <- mean(cpt_bench$time)

  fpop_bench <- microbenchmark(Fpop(prof_id$logratio, 1))
  fpop_time[i] <- mean(fpop_bench$time)
}

# Make a data frame from benchmarks.
bench <-
```

```
  cbind(prof_length, cpt_time, fpop_time) %>%
  as.data.frame()
```

Plot results

```
molten_bench <- melt(bench,
                     measure.vars=c("cpt_time", "fpop_time"),
                     value.name="time", variable.name="method")

# We filter out a data point of a much larger
# size to see in more detail the data for the
# majority of benchmarks.
ggplot(data=molten_bench[molten_bench$prof_length < 1000, ],
       mapping=aes(x=prof_length, y=time, col=method)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```