

GSOC Tests(2019)

Anuraag Srivastava(as4378@nau.edu)

February 22, 2019

Test 1

Selecting one profile id and one chromosome to continue with the test.

```
selected_profile_id = "1"  
selected_chromosome = "1"
```

Creating a data table for selected profile:

```
selected.profiles = data.table(filter(neuroblastoma$profiles,  
                                     profile.id == selected_profile_id,  
                                     chromosome == selected_chromosome))
```

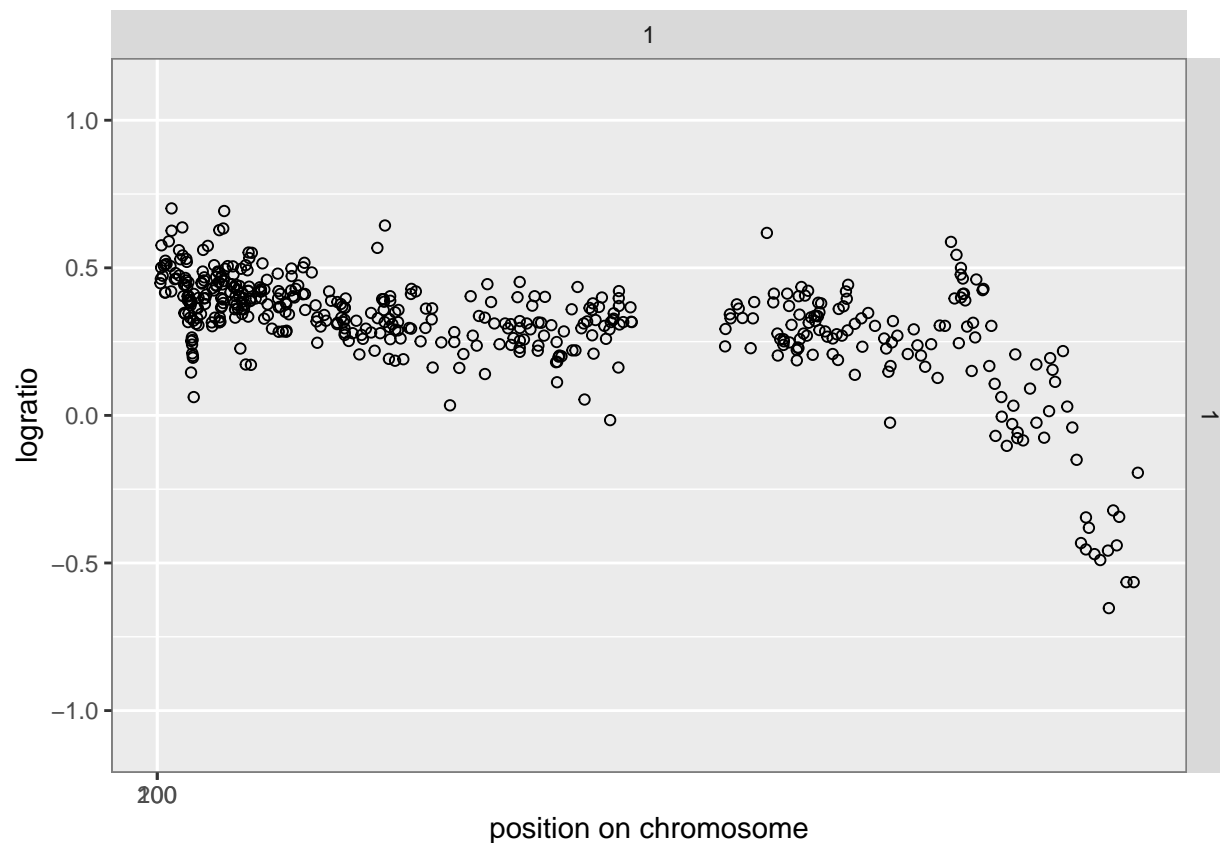
```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

Plotting this problem in a grid we get:

```
gg.unsupervised = ggplot()+  
  theme(  
    panel.margin=grid::unit(0, "lines"),  
    panel.border=element_rect(fill=NA, color="grey50")  
  )+  
  facet_grid(profile.id ~ chromosome, scales="free", space="free_x")+  
  geom_point(aes(position, logratio),  
            data=selected.profiles,  
            shape=1)+  
  scale_x_continuous(  
    "position on chromosome",  
    breaks=c(100, 200))+  
  scale_y_continuous(  
    "logratio",  
    limits=c(-1,1)*1.1)
```

```
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property  
## instead
```

```
print(gg.unsupervised)
```



Now, fitting the unsupervised change point model using the `cpt.mean` function and using “SIC0” as the penalty parameter we get:

```
pen.name = "SIC0"
(models = selected.profiles[, {
  fit.pelt = cpt.mean(
    logratio, penalty=pen.name, method="PELT")
  end = fit.pelt@cpts
  before.change = end[-length(end)]
  after.change = before.change+1L
  data.table(
    pen.name,
    pen.value=fit.pelt@pen.value,
    changes=list(as.integer((position[before.change]+position[after.change])/2)),
    before_mean=mean(logratio[1:before.change]),
    after_mean=mean(logratio[after.change:length(logratio)]),
    end_pos=position[length(position)]
  )
}], by=list(profile.id, chromosome)])
```

##	profile.id	chromosome	pen.name	pen.value	changes	before_mean
## 1:	1	1	SIC0	6.161207	212809180	0.3518651
##			after_mean	end_pos		

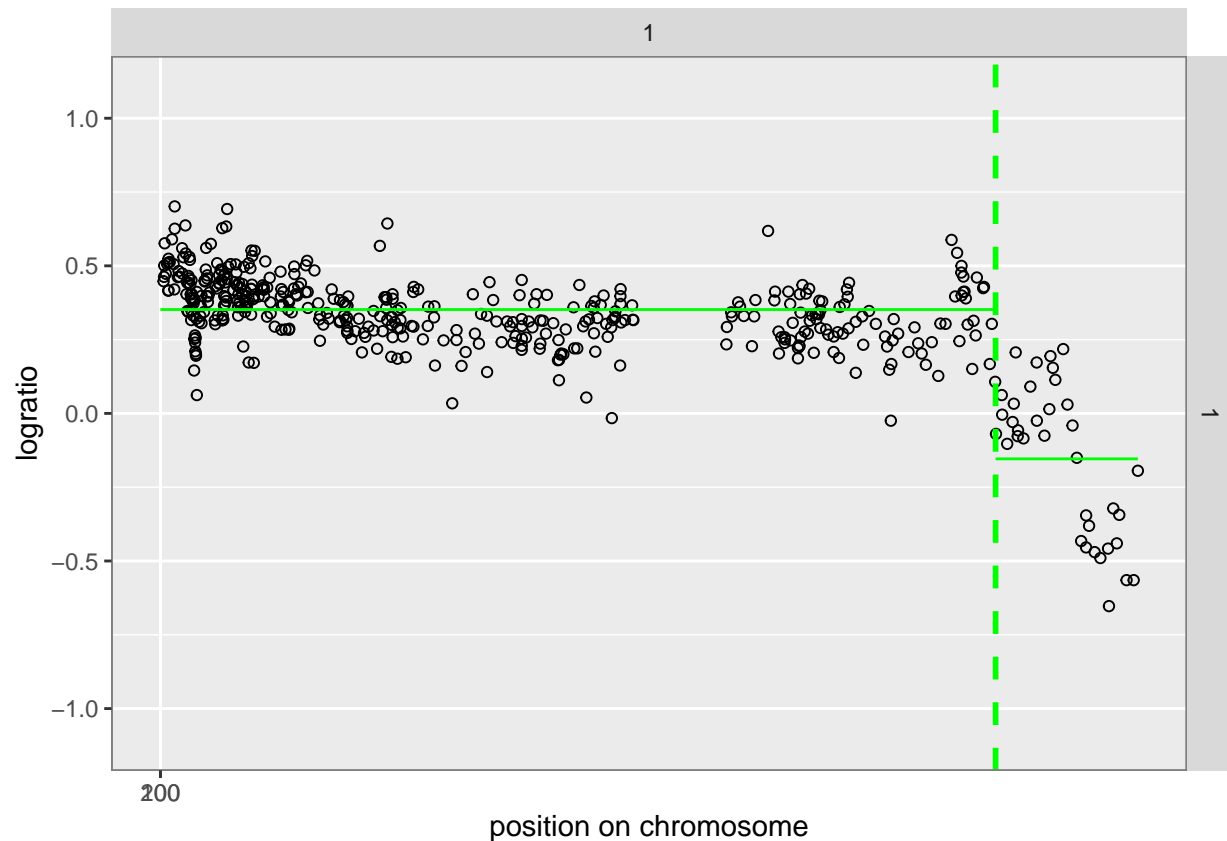
```
## 1: -0.1539234 249063592
(changes = models[, data.table(
  change=changes[[1]],before_mean = before_mean, after_mean = after_mean, end_pos=end_pos
), by=list(profile.id, chromosome, pen.name)])

##   profile.id chromosome pen.name   change before_mean after_mean
## 1:         1          1     SICO 212809180   0.3518651 -0.1539234
##      end_pos
## 1: 249063592

segments.mean = NULL
segments.mean = rbind(segments.mean, c("start" = 0, "end" = changes$change,
                                         "mean" = changes$before_mean))
segments.mean = rbind(segments.mean, c("start" = changes$change,
                                         "end" = changes$end_pos,
                                         "mean" = changes$after_mean))
```

Plotting the data along with optimal segment means (green line segments) we get:

```
gg.unsupervised+
  theme(legend.box="horizontal")+
  geom_vline(aes(
    xintercept=change),
    color="green",
    size=1,
    linetype="dashed",
    data=changes)+
  geom_segment(aes(
    x = start,
    y = mean,
    xend = end,
    yend = mean,
    col = I("green")),
    data = as.data.frame(segments.mean))
```



Test 2

Selecting 2 data sets.

```
profile1 = "1"
profile2 = "4"
chromosome1 = "1"
chromosome2 = "1"
```

Creating data table for selected profiles:

```
selected.profiles1 = filter(neuroblastoma$profiles, profile.id == profile1,
                           chromosome == chromosome1)
selected.profiles2 = filter(neuroblastoma$profiles, profile.id == profile2,
                           chromosome == chromosome2)

selected.profiles = data.table(rbind(selected.profiles1, selected.profiles2))
```

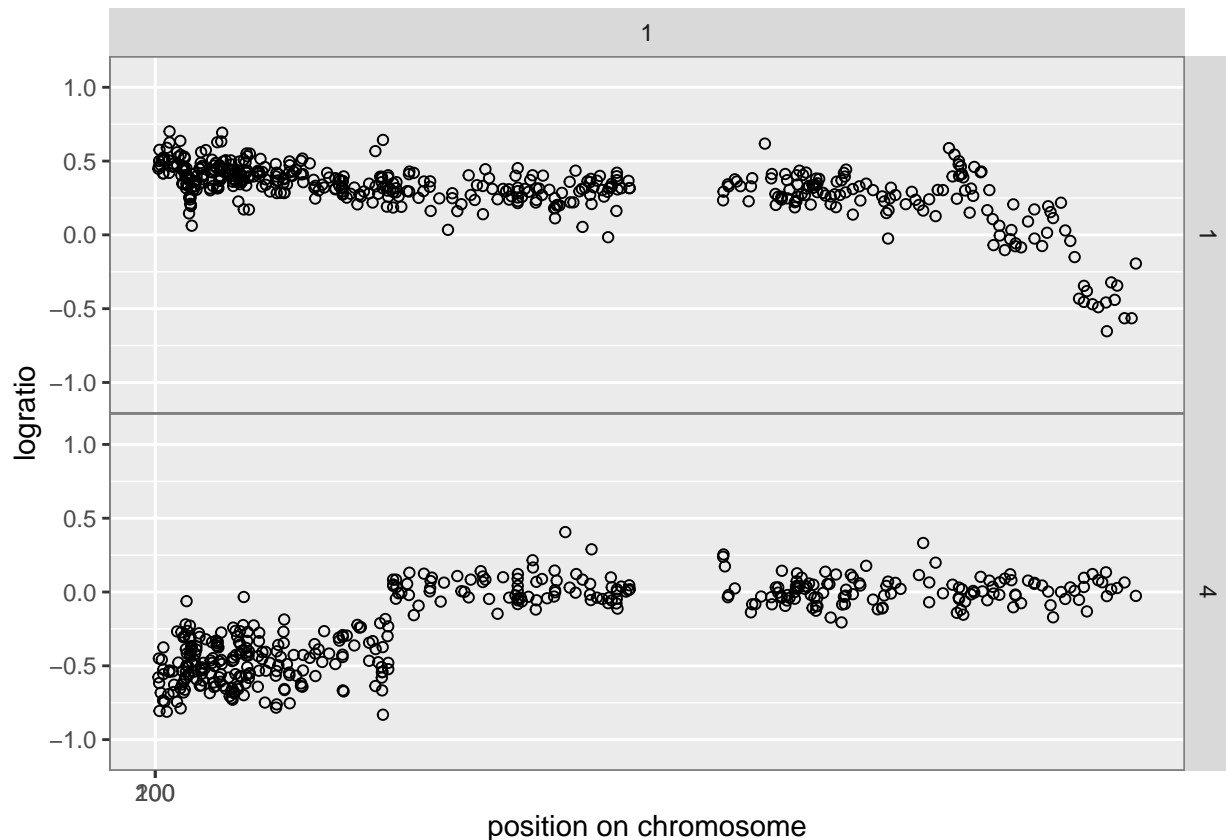
Printing the problems:

```
gg.unsupervised = ggplot()+
  theme(
    panel.margin=grid::unit(0, "lines"),
    panel.border=element_rect(fill=NA, color="grey50")
```

```
)+
facet_grid(profile.id ~ chromosome, scales="free", space="free_x")+
geom_point(aes(position, logratio),
           data=selected.profiles,
           shape=1)+
scale_x_continuous(
  "position on chromosome",
  breaks=c(100, 200))+
scale_y_continuous(
  "logratio",
  limits=c(-1,1)*1.1)

## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead

print(gg.unsupervised)
```



Now, fitting the unsupervised change point model using the `cpt.mean` function and using 3 penalty paramters "SIC0, BIC0 and AIC0" we get:

```
pen.name = "SIC0"
(models = selected.profiles[, {
  fit.pelt = changepoint::cpt.mean(
    logratio, penalty=pen.name, method="PELT")
  end = fit.pelt@cpts
  before.change = end[-length(end)]
```

```

after.change = before.change+1L
data.table(
  pen.name,
  pen.value=fit.pelt@pen.value,
  changes=list(
    as.integer((position[before.change]+position[after.change])/2)
  ),
  before_mean=mean(logratio[1:before.change]),
  after_mean=mean(logratio[after.change:length(logratio)]),
  end_pos=position[length(position)])
}, by=list(profile.id, chromosome))

```

```

##   profile.id chromosome pen.name pen.value  changes before_mean
## 1:          1          1    SICO  6.161207 212809180  0.3518651
## 2:          4          1    SICO  6.059123  59792500 -0.4800527
##   after_mean  end_pos
## 1: -0.15392338 249063592
## 2:  0.01468708 249063592

```

```

pen.name = "BICO"
(models1 = selected.profiles[, {
  fit.pelt = changepoint::cpt.mean(
    logratio, penalty=pen.name, method="PELT")
  end = fit.pelt@cpts
  before.change = end[-length(end)]
  after.change = before.change+1L
  data.table(
    pen.name,
    pen.value=fit.pelt@pen.value,
    changes=list(
      as.integer((position[before.change]+position[after.change])/2)
    ),
    before_mean=mean(logratio[1:before.change]),
    after_mean=mean(logratio[after.change:length(logratio)]),
    end_pos=position[length(position)])
}, by=list(profile.id, chromosome))

```

```

##   profile.id chromosome pen.name pen.value  changes before_mean
## 1:          1          1    BICO  6.161207 212809180  0.3518651
## 2:          4          1    BICO  6.059123  59792500 -0.4800527
##   after_mean  end_pos
## 1: -0.15392338 249063592
## 2:  0.01468708 249063592

```

```
models = rbind(models, models1)
```

```

pen.name = "AICO"
(models2 = selected.profiles[, {
  fit.pelt = changepoint::cpt.mean(
    logratio, penalty=pen.name, method="PELT")
  end = fit.pelt@cpts
  before.change = end[-length(end)]
  after.change = before.change+1L
  data.table(

```

```

    pen.name,
    pen.value=fit.pelt@pen.value,
    changes=list(
      as.integer((position[before.change]+position[after.change])/2)
    ),
    before_mean=mean(logratio[1:before.change]),
    after_mean=mean(logratio[after.change:length(logratio)]),
    end_pos=position[length(position)])
}, by=list(profile.id, chromosome)])

##      profile.id chromosome pen.name pen.value  changes before_mean
## 1:           1           1    AICO         2 212809180  0.3518651
## 2:           4           1    AICO         2  59792500 -0.4800527
##      after_mean  end_pos
## 1: -0.15392338 249063592
## 2:  0.01468708 249063592

models = rbind(models, models2)

changes = models[, data.table(
  change=changes[[1]], before_mean = before_mean,
  after_mean = after_mean, end_pos=end_pos,
  start = 0, end = changes[[1]], mean = before_mean
), by=list(profile.id, chromosome, pen.name)]

segments = changes[, data.table(
  change=change, before_mean = before_mean,
  after_mean = after_mean, end_pos=end_pos,
  start = change, end = end_pos, mean = after_mean
), by=list(profile.id, chromosome, pen.name)]

segments = rbind(changes, segments)

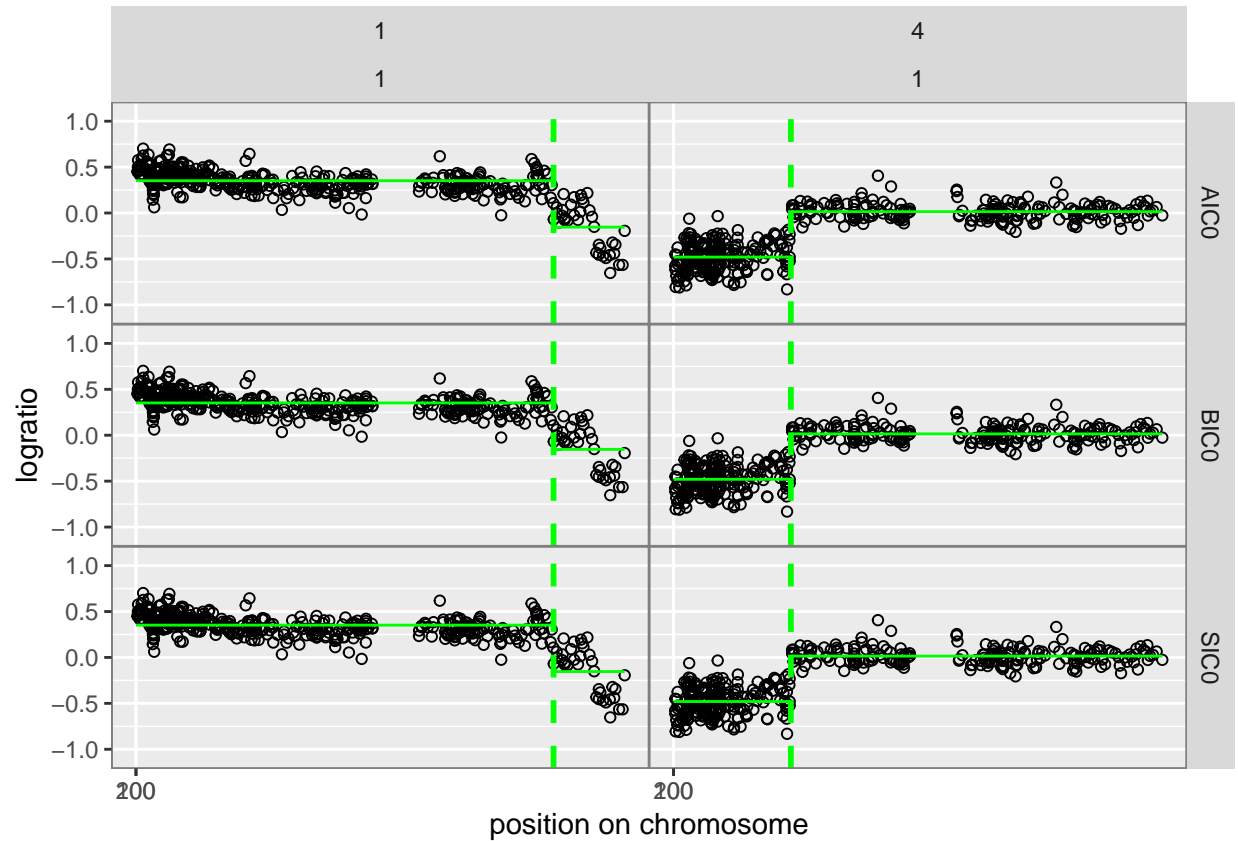
```

Plotting the data along with optimal segment means (green line segments) we get:

```

gg.unsupervised+
  facet_grid(pen.name ~ profile.id + chromosome, scales="free", space="free_x")+
  theme(legend.box="horizontal")+
  geom_vline(aes(
    xintercept=change),
    color="green",
    size=1,
    linetype="dashed",
    data=segments)+
  geom_segment(aes(
    x = start,
    y = mean,
    xend = end,
    yend = mean),
    col = I("green"),
    data = segments
)

```



Test 3

Performing test on all the data sets (each profile.id and chromosome combination).

```
selected_ids = unique(neuroblastoma$profiles$profile.id)
selected_data = neuroblastoma$profiles
num_tests = length(selected_ids)
```

Performing the test:

```
timing_list = list()
n = 1
while(n <= num_tests){
  current_id = selected_ids[n]
  current_data = filter(selected_data, profile.id == current_id)
  length = length(current_data$logratio)
  timing = microbenchmark(
    "cpt_mean"={
      cpt.mean(current_data$logratio, method="PELT", pen.value = 1)
    },
    "fpop"={
      Fpop(current_data$logratio, 1)
    }, times=5)
}
```



```

    timing_list[[paste(n)]] = data.table(length, timing)
    n = n + 1
  }

timing.dt = do.call(rbind, timing_list)

ggplot(data = timing.dt, aes(x = log(timing.dt$length),
                             y = log(timing.dt$time),
                             col = timing.dt$expr)) +
  geom_smooth() +
  labs(x="log(size)", y="log(time(ms))", col="method")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

```

