Copy number estimation and CNA detection in tumor samples from WGS HTS data using HMMCopy

Lavinia Carabet

11/19/2020

1. Setup the libraries and input data

Install libraries

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("HMMcopy")
```

Load libraries

```
options(stringsAsFactors = TRUE)
library(HMMcopy)
```

2. Generating Copy Number Profiles

Load normal HTS copy number data

```
rfile <- system.file("extdata", "normal.wig", package = "HMMcopy")
gfile <- system.file("extdata", "gc.wig", package = "HMMcopy")
mfile <- system.file("extdata", "map.wig", package = "HMMcopy")
normal_reads <-  wigsToRangedData(rfile, gfile, mfile)
normal_reads[1000:1010,]
```

```
##     chr    start      end reads     gc      map
##  1:   6  9990001 10000001  7351 0.3932 0.558421
##  2:   6 10000001 10010001 11080 0.3925 0.929000
##  3:   6 10010001 10020001  9369 0.3689 0.975162
##  4:   6 10020001 10030001  9505 0.3634 0.963870
##  5:   6 10030001 10040001 10392 0.3903 0.889620
##  6:   6 10040001 10050001 10785 0.3854 0.952784
##  7:   6 10050001 10060001 11084 0.4264 0.916659
##  8:   6 10060001 10070001  9712 0.3826 0.940946
##  9:   6 10070001 10080001  5320 0.4011 0.422648
```

```
## 10:   6 10080001 10090001 11014 0.3944 0.974912
## 11:   6 10090001 10100001 10318 0.3667 0.971846
```

```
#dim(normal_reads)
```

Correcting normal HTS copy number data

```
normal_copy <-  correctReadcount(normal_reads)
```

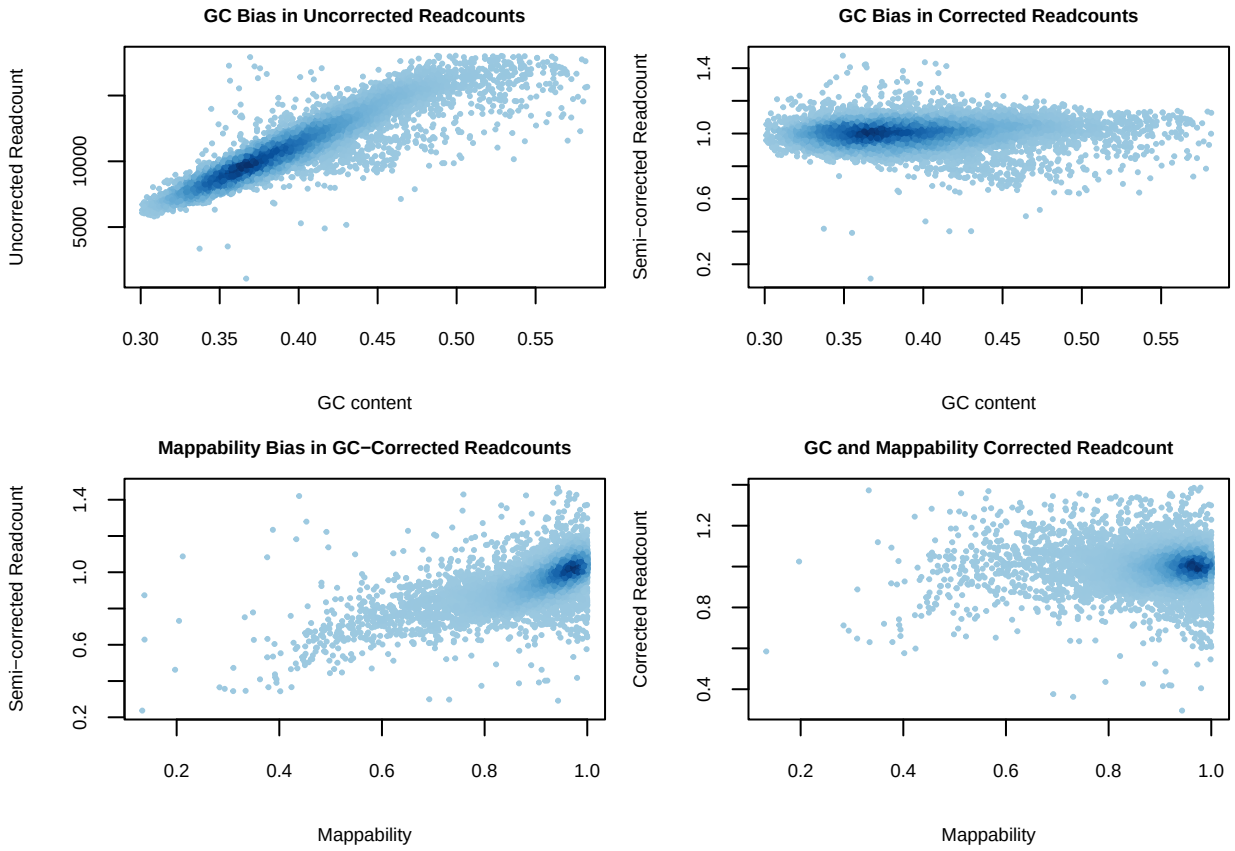
```
## Applying filter on data...

## Correcting for GC bias...

## Correcting for mappability bias...
```

```
normal_copy[1000:1010,]
```

```
##     chr    start      end reads     gc      map valid ideal    cor.gc   cor.map
##  1:   6  9990001 10000001  7351 0.3932 0.558421  TRUE FALSE 0.6667950 0.9462945
##  2:   6 10000001 10010001 11080 0.3925 0.929000  TRUE  TRUE 1.0083988 1.0391931
##  3:   6 10010001 10020001  9369 0.3689 0.975162  TRUE  TRUE 0.9602569 0.9369282
##  4:   6 10020001 10030001  9505 0.3634 0.963870  TRUE  TRUE 1.0055782 0.9947256
##  5:   6 10030001 10040001 10392 0.3903 0.889620  TRUE FALSE 0.9557880 1.0304725
##  6:   6 10040001 10050001 10785 0.3854 0.952784  TRUE  TRUE 1.0154319 1.0179066
##  7:   6 10050001 10060001 11084 0.4264 0.916659  TRUE  TRUE 0.8754294 0.9154566
##  8:   6 10060001 10070001  9712 0.3826 0.940946  TRUE  TRUE 0.9268861 0.9420144
##  9:   6 10070001 10080001  5320 0.4011 0.422648  TRUE FALSE 0.4655253 0.7623204
## 10:   6 10080001 10090001 11014 0.3944 0.974912  TRUE  TRUE 0.9934066 0.9695664
## 11:   6 10090001 10100001 10318 0.3667 0.971846  TRUE  TRUE 1.0709252 1.0491263
##             copy
##  1: -0.079638885
##  2:  0.055463804
##  3: -0.093989665
##  4: -0.007629453
##  5:  0.043305954
##  6:  0.025605244
##  7: -0.127436530
##  8: -0.086178990
##  9: -0.391530635
## 10: -0.044588376
## 11:  0.069188320
```

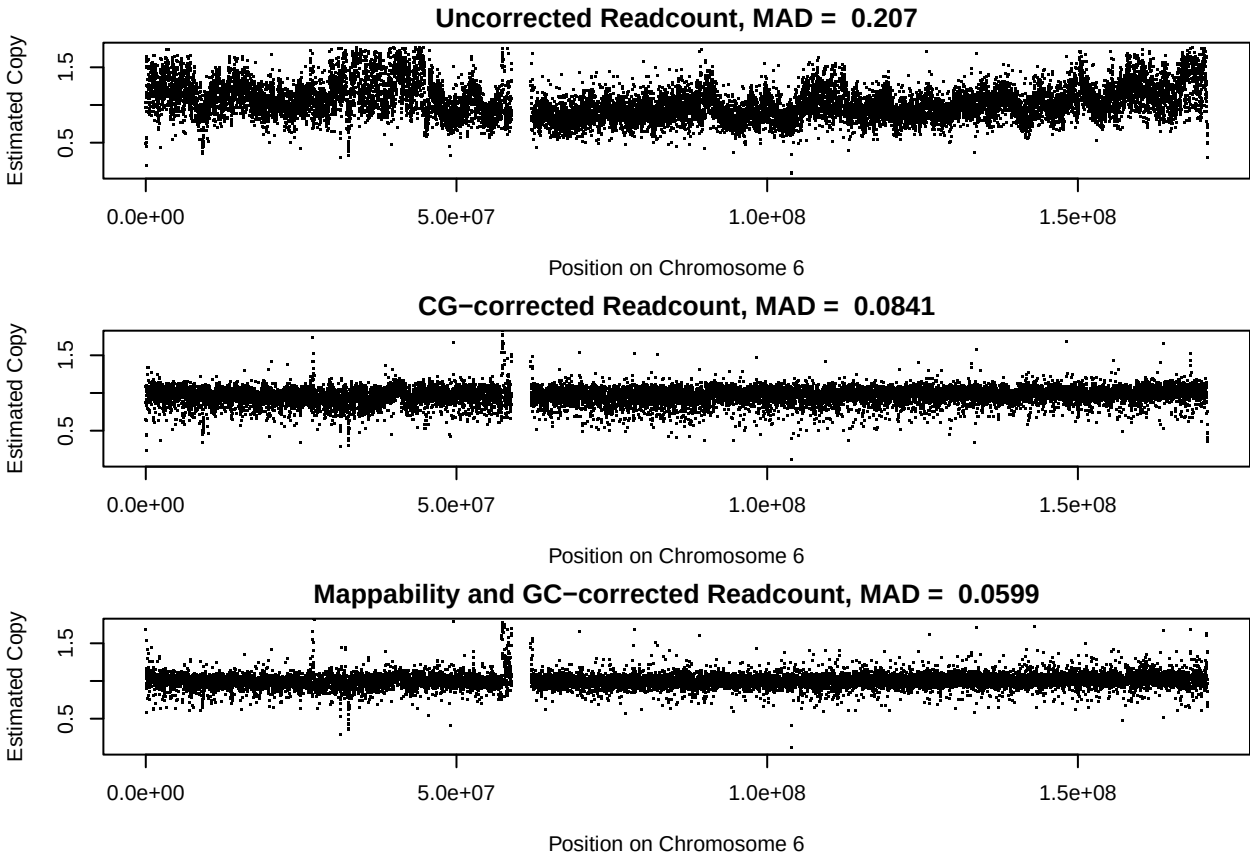
Visualizing the effects of correction

```
par(cex.main = 0.7, cex.lab =0.7, cex.axis =0.7, mar = c(4,4,2, 0.5))
plotBias(normal_copy, pch =20, cex = 0.5)
```



Visualizing corrected copy number profiles

```
par(mar = c(4,4,2, 0))
plotCorrection(normal_copy, pch = ".")
```

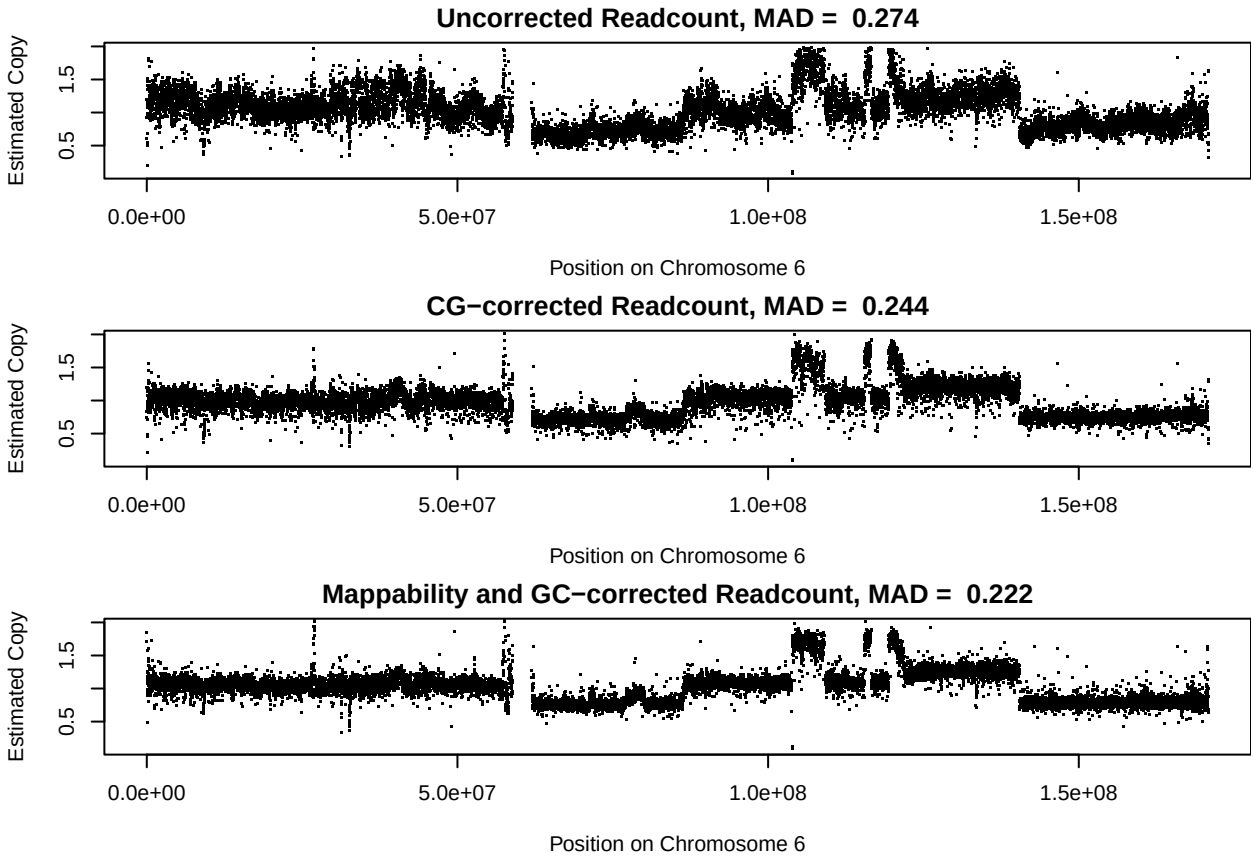


Correcting and visualizing tumor copy number profiles

```
tfile <- system.file("extdata","tumour.wig", package = "HMMcopy")
tumor_copy <- correctReadcount(wigsToRangedData(tfile, gfile, mfile))
```

```
## Applying filter on data...

## Correcting for GC bias...

## Correcting for mappability bias...
```

```
par(mar = c(4,4,2, 0))
plotCorrection(tumor_copy, pch = ".")
```



3. Segmentation and Classification of Copy Number Profiles

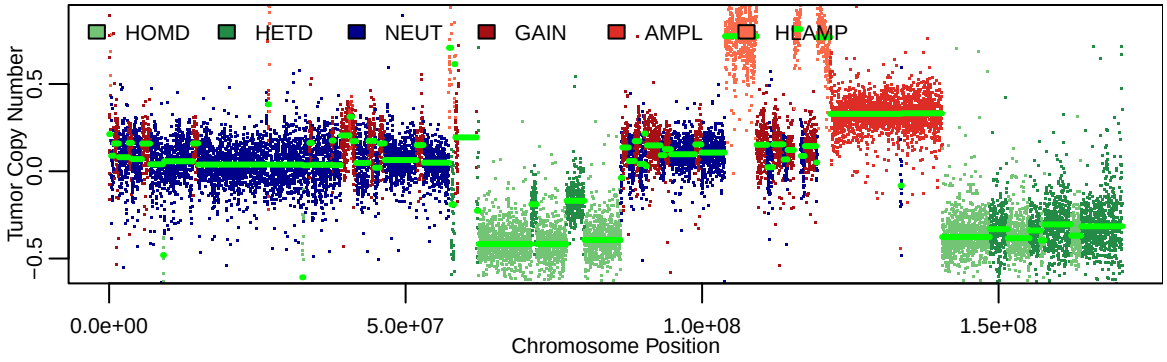
```
tumor_segments <- HMMsegment(tumor_copy)
```

```
## Initialization

## EM iteration: 1 Log likelihood: -Inf

## Expectation

## Maximization

## EM iteration: 2 Log likelihood: 3723.38977598144

## Expectation

## Maximization

## EM iteration: 3 Log likelihood: 8651.17813251384

## Expectation
```

```
## Maximization

## EM iteration: 4 Log likelihood: 9776.42997349625

## Expectation

## Maximization

## EM iteration: 5 Log likelihood: 10073.1602591075

## Expectation

## Maximization

## EM iteration: 6 Log likelihood: 10155.0644113838

## Expectation

## Maximization

## EM iteration: 7 Log likelihood: 10175.7127890004

## Expectation

## Maximization

## EM iteration: 8 Log likelihood: 10181.1575769066

## Expectation

## Maximization

## EM iteration: 9 Log likelihood: 10183.4033459711

## Expectation

## Maximization

## EM iteration: 10 Log likelihood: 10184.3672316225

## Expectation

## Maximization

## Re-calculating latest responsibilties for output

## Optimal parameters found, segmenting and classifying
```

Visualizing segments and classified states

```
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
plotSegments(tumor_copy, tumor_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
cols <- stateCols() #6 default state colors
legend("topleft", c("HOMD", "HETD", "NEUT", "GAIN", "AMPL", "HLAMP"),
       fill = cols, horiz = TRUE, bty = "n", cex = 0.7)
```



Improving segmentation performance

```
default_param <- HMMsegment(tumor_copy, getparam = TRUE)
default_param
```

```
##   strength         e          mu lambda  nu kappa           m   eta gamma
## 1    1e+07 0.9999999 -0.42054605     20 2.1    50 -0.42054605 5e+04     3
## 2    1e+07 0.9999999 -0.28184226     20 2.1    50 -0.28184226 5e+04     3
## 3    1e+07 0.9999999  0.04200362     20 2.1   700  0.04200362 5e+05     3
## 4    1e+07 0.9999999  0.18884920     20 2.1   100  0.18884920 5e+04     3
## 5    1e+07 0.9999999  0.36472889     20 2.1    50  0.36472889 5e+04     3
## 6    1e+07 0.9999999  0.89363465     20 2.1    50  0.89363465 5e+04     3
##            S
## 1 0.01858295
```

```
## 2 0.01858295
## 3 0.01858295
## 4 0.01858295
## 5 0.01858295
## 6 0.01858295
```

```
#6 states, 10 parameters matrix
```

Reducing the number of segments a.k.a increasing the length of segments

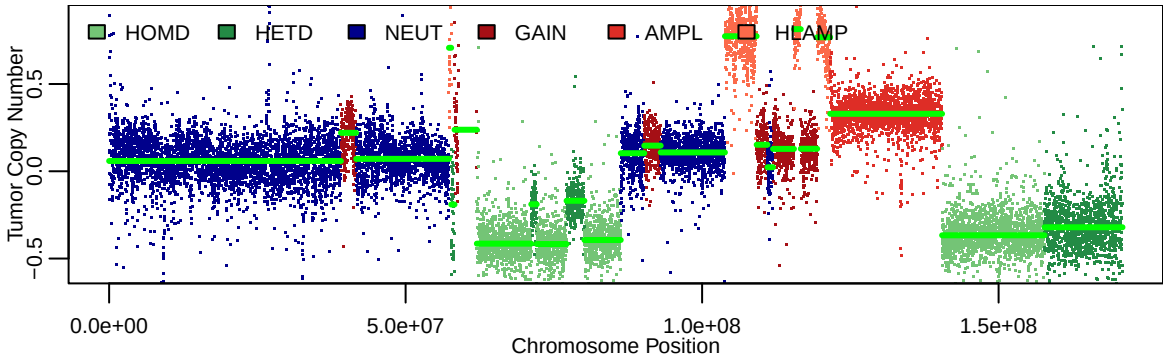
```
longseg_param <-  default_param
longseg_param$e <-  0.9999999999999999
longseg_param$strength <- 1e30
longseg_segments <-  HMMsegment(tumor_copy, longseg_param)
```

```
## Initialization

## EM iteration: 1 Log likelihood: -Inf

## Expectation

## Maximization

## EM iteration: 2 Log likelihood: 3338.8911576833

## Expectation

## Maximization

## EM iteration: 3 Log likelihood: 7873.64684519957

## Expectation

## Maximization

## EM iteration: 4 Log likelihood: 8823.34678123053

## Expectation

## Maximization

## EM iteration: 5 Log likelihood: 9043.12460118821

## Expectation

## Maximization
```

```
## EM iteration: 6 Log likelihood: 9093.46182766672

## Expectation

## Maximization

## EM iteration: 7 Log likelihood: 9099.25718532247

## Expectation

## Maximization

## Re-calculating latest responsibilties for output

## Optimal parameters found, segmenting and classifying
```

Visualizing segments and classified states to confirm decrease in segments as intended

```
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
plotSegments(tumor_copy, longseg_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
cols <- stateCols() #6 default state colors
legend("topleft", c("HOMD", "HETD", "NEUT", "GAIN", "AMPL", "HLAMP"),
       fill = cols, horiz = TRUE, bty = "n", cex = 0.7)
```



Adjusting copy number state ranges - correcting the incorrect median of each copy number state in the plot above

problem with mu parameter

```
#output of segmentation process: matrix of the median of 6 states (rows) after each iteration (7)
#of the optimization algorithm (columns)
longseg_segments$mus
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] -0.42054605 -0.41759475 -0.41853646 -0.41893133 -0.41910344 -0.41919012
## [2,] -0.28184226 -0.28220554 -0.28194739 -0.28192023 -0.28192061 -0.28192106
## [3,]  0.04200362  0.04259144  0.04240217  0.04229718  0.04227656  0.04227098
## [4,]  0.18884920  0.18753064  0.18790296  0.18779510  0.18803754  0.18817426
## [5,]  0.36472889  0.36339203  0.36354965  0.36374378  0.36383817  0.36388567
## [6,]  0.89363465  0.89184731  0.89192987  0.89200232  0.89203918  0.89205764
##             [,7]
## [1,] -0.41923682
## [2,] -0.28192207
## [3,]  0.04226849
## [4,]  0.18824202
## [5,]  0.36391053
## [6,]  0.89206682
```

first column initial suggested mu

```
longseg_param$mu
```

```
## [1] -0.42054605 -0.28184226  0.04200362  0.18884920  0.36472889  0.89363465
```

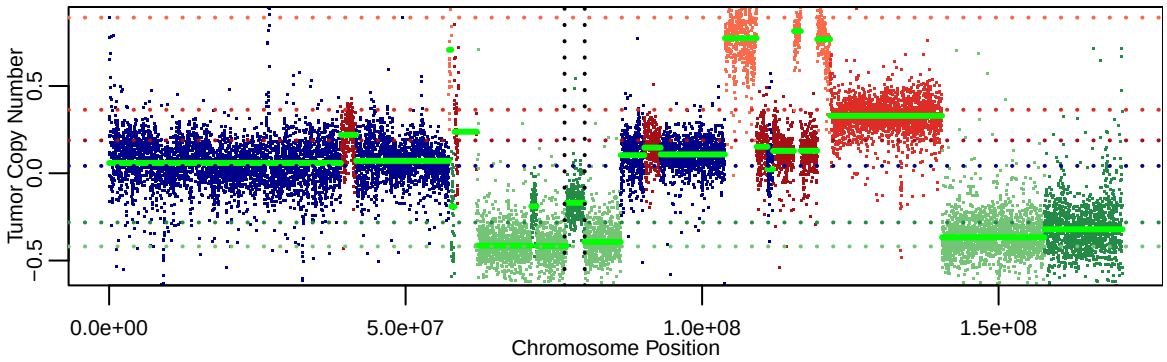
last column actual values used during the segmentation process

```
longseg_segments$mus[,7]
```

```
## [1] -0.41923682 -0.28192207  0.04226849  0.18824202  0.36391053  0.89206682
```

Visualising the problem - Medians not running through middle of segments of many states

```
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
plotSegments(tumor_copy, longseg_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
for (i in 1:nrow(longseg_segments$mus)) {
  abline(h=longseg_segments$mus[i, ncol(longseg_segments$mus)], col = cols[i], lwd =2, lty =3)
}
abline(v = 7.68e7, lwd = 2, lty = 3)
abline(v = 8.02e7, lwd = 2, lty = 3)
```



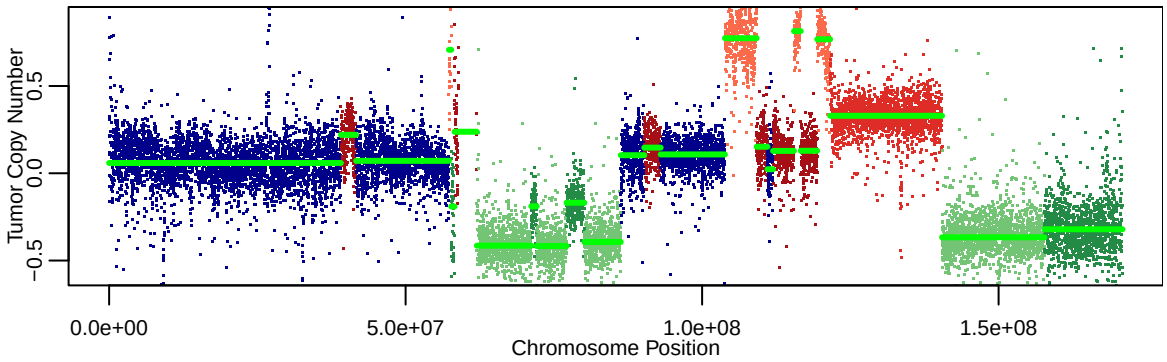
Update/Re-initialiaze mu param to solve the problem

```
newmu_param <- longseg_param
newmu_param$mu <-  c(-0.5, -0.4,-0.15, 0.1,0.4, 0.7) # why these values? nothing has changed
newmu_segments <- HMMsegment(tumor_copy, newmu_param)
```

```
## Initialization

## EM iteration: 1 Log likelihood: -Inf

## Expectation

## Maximization

## EM iteration: 2 Log likelihood: 3522.90225659908

## Expectation

## Maximization

## EM iteration: 3 Log likelihood: 6653.89468619305
```

```
## Expectation

## Maximization

## EM iteration: 4 Log likelihood: 8235.83083410406

## Expectation

## Maximization

## EM iteration: 5 Log likelihood: 8900.86391071681

## Expectation

## Maximization

## EM iteration: 6 Log likelihood: 9067.98507720073

## Expectation

## Maximization

## EM iteration: 7 Log likelihood: 9104.30417259847

## Expectation

## Maximization

## EM iteration: 8 Log likelihood: 9105.3370234295

## Expectation

## Maximization

## Re-calculating latest responsibilties for output

## Optimal parameters found, segmenting and classifying
```

```
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
plotSegments(tumor_copy, newmu_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
# for (i in 1:nrow(newmu_segments$mus)) {
#   abline(h=newmu_segments$mus[i, ncol(newmu_segments$mus)], col = cols[i], lwd =2, lty =3)
# }
#  abline(v = 7.68e7, lwd = 2, lty = 3)
#  abline(v = 8.02e7, lwd = 2, lty = 3)
```



Understanding parameter convergence

```
newmu_segments$mus
```

```
##       [,1]        [,2]        [,3]       [,4]        [,5]        [,6]
## [1,] -0.50 -0.42054605 -0.41943464 -0.4193608 -0.41933370 -0.41932098
## [2,] -0.40 -0.29262755 -0.28206379 -0.2819277 -0.28192558 -0.28192978
## [3,] -0.15  0.04183712  0.04199752  0.0422006  0.04230104  0.04228193
## [4,]  0.10  0.17033469  0.17508744  0.1852921  0.18762264  0.18789902
## [5,]  0.40  0.36329444  0.36360858  0.3637754  0.36385429  0.36389415
## [6,]  0.70  0.89064400  0.89162707  0.8918601  0.89196792  0.89202067
##            [,7]        [,8]
## [1,] -0.4193143 -0.41931052
## [2,] -0.2819329 -0.28193448
## [3,]  0.0422720  0.04226889
## [4,]  0.1881261  0.18821599
## [5,]  0.3639151  0.36392641
## [6,]  0.8920473  0.89206090
```

```
#newmu_param$mu <-  c(-0.5, -0.4,-0.15, 0.1,0.4, 0.7) #alg ignores these param values
```

Understanding parameter convergence

```
longseg_param$mu
```

```
## [1] -0.42054605 -0.28184226  0.04200362  0.18884920  0.36472889  0.89363465
```

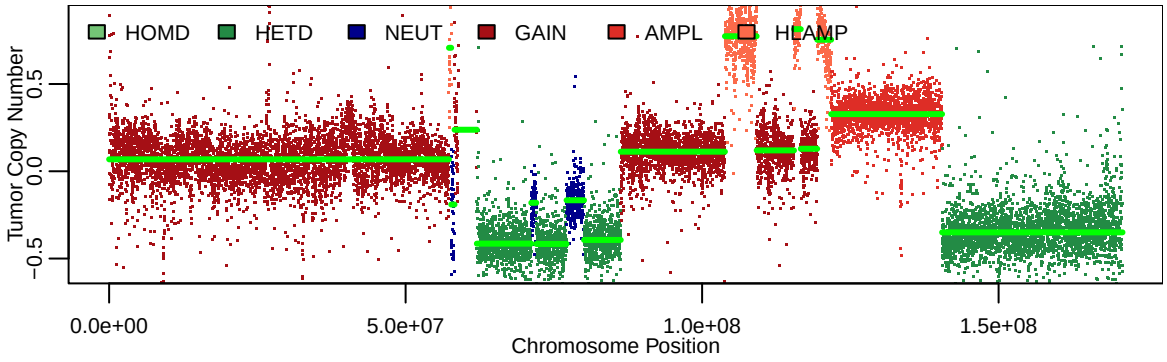
Overriding parameter convergence

```
#solution disallow the alg from making large shifts to mu
#achieved by setting the prior mean of n (i.e.m - optimal value of mu) to values identical to mu
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
newmu_param$m <- newmu_param$mu
realmu_segments <- HMMsegment(tumor_copy, newmu_param)
```

```
## Initialization

## EM iteration: 1 Log likelihood: -Inf

## Expectation

## Maximization

## EM iteration: 2 Log likelihood: 3522.95088427687

## Expectation

## Maximization

## EM iteration: 3 Log likelihood: 8183.45688428581

## Expectation

## Maximization

## EM iteration: 4 Log likelihood: 9082.73187208635

## Expectation

## Maximization

## EM iteration: 5 Log likelihood: 9287.48303693139

## Expectation

## Maximization
```

```
## EM iteration: 6 Log likelihood: 9334.3204375332

## Expectation

## Maximization

## EM iteration: 7 Log likelihood: 9342.47770043786

## Expectation

## Maximization

## Re-calculating latest responsibilties for output

## Optimal parameters found, segmenting and classifying
```

```
plotSegments(tumor_copy, realmu_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
# for (i in 1:nrow(realmu_segments$mus)) {
#   abline(h=realmu_segments$mus[i, ncol(realmu_segments$mus)], col = cols[i], lwd =2, lty =3)
# }
#  abline(v = 7.68e7, lwd = 2, lty = 3)
#  abline(v = 8.02e7, lwd = 2, lty = 3)
cols <- stateCols() #6 default state colors
legend("topleft", c("HOMD", "HETD", "NEUT", "GAIN", "AMPL", "HLAMP"),
       fill = cols, horiz = TRUE, bty = "n", cex = 0.7)
```



4. Matched Tumor-Normal Sample Correction

Normalizing tumor by normal copy number profiles

```
somatic_copy <- tumor_copy
#LOGARITHM IDENTITY: log(a) -log(b) == log(a/b)
somatic_copy$copy <- tumor_copy$copy - normal_copy$copy
```

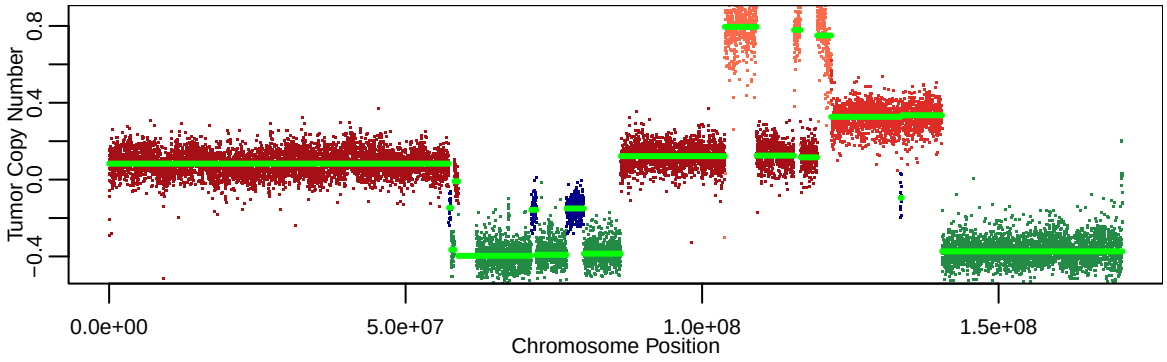
Segmentation and visualization of somatic copy number aberration

```
somatic_segments <- HMMsegment(somatic_copy, newmu_param)
```

```
## Initialization

## EM iteration: 1 Log likelihood: -Inf

## Expectation

## Maximization

## EM iteration: 2 Log likelihood: 5877.44091770824

## Expectation

## Maximization

## EM iteration: 3 Log likelihood: 17364.4135409784

## Expectation

## Maximization

## EM iteration: 4 Log likelihood: 18799.692523898

## Expectation

## Maximization

## EM iteration: 5 Log likelihood: 19073.7138794748

## Expectation

## Maximization

## EM iteration: 6 Log likelihood: 19130.9604020607
```

```
## Expectation

## Maximization

## EM iteration: 7 Log likelihood: 19140.0987141774

## Expectation

## Maximization

## Re-calculating latest responsibilties for output

## Optimal parameters found, segmenting and classifying
```

```
par(mfrow = c(2,1), cex.main = 0.5, cex.lab = 0.7, cex.axis =0.7,
    mar = c(4, 4,0,0),  mgp = c(1,0.5,0))
plotSegments(somatic_copy, somatic_segments, pch=".",
             ylab = "Tumor Copy Number", xlab = "Chromosome Position")
```



Export somatic copy number aberration

```
#somatic_copy
somatic_segments$segs
```

```
##    chr     start       end state       median
## 1    6         1  57310001     4  0.083176334
## 2    6  57310001  57680001     3 -0.146329934
## 3    6  57680001  58250001     2 -0.364764414
## 4    6  58250001  58870001     4 -0.008655485
## 5    6  58870001  71180001     2 -0.396831248
## 6    6  71180001  72040001     3 -0.157282016
## 7    6  72040001  77130001     2 -0.391576047
## 8    6  77130001  80060001     3 -0.150318033
## 9    6  80060001  86280001     2 -0.385636445
## 10   6  86280001 103840001     4  0.122170348
## 11   6 103840001 109200001     6  0.795133381
## 12   6 109200001 115500001     4  0.124886236
## 13   6 115500001 116640001     6  0.779041776
## 14   6 116640001 119410001     4  0.115887334
## 15   6 119410001 121770001     6  0.750261711
## 16   6 121770001 133470001     5  0.326678421
## 17   6 133470001 133670001     3 -0.095143541
## 18   6 133670001 140470001     5  0.335157207
## 19   6 140470001 170900001     2 -0.373558341
```

```
#str(somatic_segments)
readr::write_delim(somatic_segments$segs,"somatic_segments_LAC.txt")
```

Session info

```
toLatex(sessionInfo())
```

```
## \begin{itemize}\raggedright
##   \item R version 4.1.2 (2021-11-01), \verb|x86_64-w64-mingw32|
##   \item Locale: \verb|LC_COLLATE=English_United States.1252|, \verb|LC_CTYPE=English_United States.1
##   \item Running under: \verb|Windows 10 x64 (build 19044)|
##   \item Matrix products: default
##   \item Base packages: base, datasets, graphics, grDevices, methods,
##     stats, utils
##   \item Other packages: data.table~1.14.2, HMMcopy~1.36.0
##   \item Loaded via a namespace (and not attached): bit~4.0.4,
##     bit64~4.0.5, cli~3.1.1, compiler~4.1.2, crayon~1.5.0,
##     digest~0.6.29, ellipsis~0.3.2, evaluate~0.15, fansi~1.0.2,
##     fastmap~1.1.0, glue~1.6.1, highr~0.9, hms~1.1.1, htmltools~0.5.2,
##     KernSmooth~2.23-20, knitr~1.37, lifecycle~1.0.1, magrittr~2.0.2,
##     parallel~4.1.2, pillar~1.7.0, pkgconfig~2.0.3, purrr~0.3.4,
##     R6~2.5.1, readr~2.1.2, rlang~1.0.1, rmarkdown~2.13,
##     rstudioapi~0.13, stringi~1.7.6, stringr~1.4.0, tibble~3.1.6,
##     tidyselect~1.1.2, tools~4.1.2, tzdb~0.2.0, utf8~1.2.2, vctrs~0.3.8,
```

```
##     vroom~1.5.7, xfun~0.29, yaml~2.2.2
## \end{itemize}
```