Copy Number Analysis for DNA-Seq Data

Oscar M Rueda

July 25, 2016

Contents

1 Introduction

The purpose of this practical is to introduce different methods for analysing copy number data from next-generation sequencing data.

2 Analysis of Copy Number Data using depth

In your folder you should have a file named HCC1143.mix1.n20t80.subsampled.bam from whole-genome sequencing with low coverage (shallow sequencing).

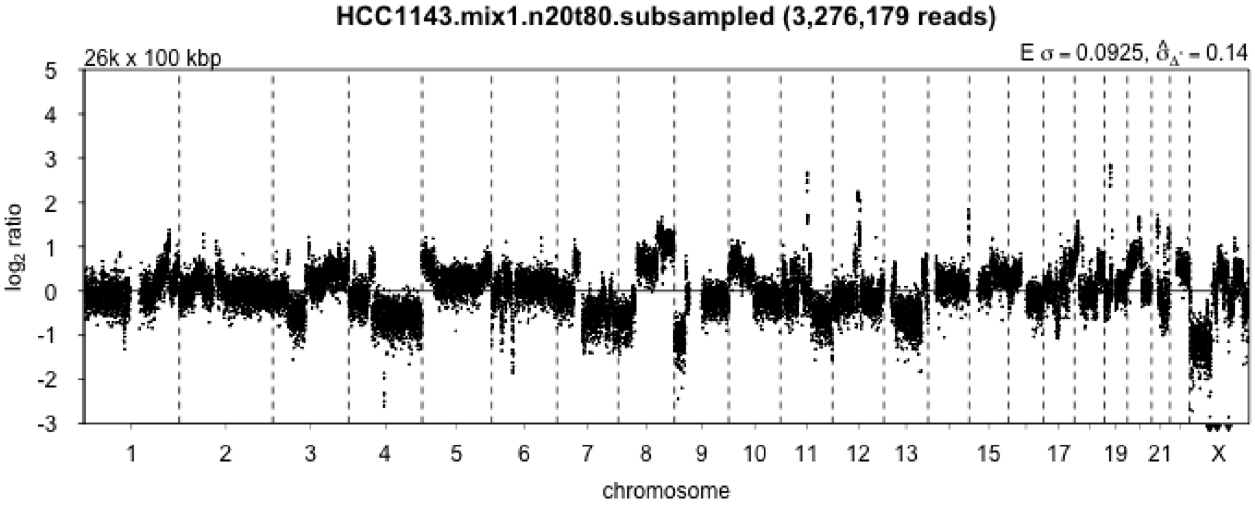
☞Try to find out how many reads we have in this dataset.

Use Case: We are going to use a bin size of 100kb because these we have very shallow sequencing. This is equivalent to a 26k CGH array.

```r
library(QDNAseq)
sample.files <- "HCC1143.mix1.n20t80.subsampled.bam"
binSize <- 100
bins <- getBinAnnotations(binSize=binSize)
```

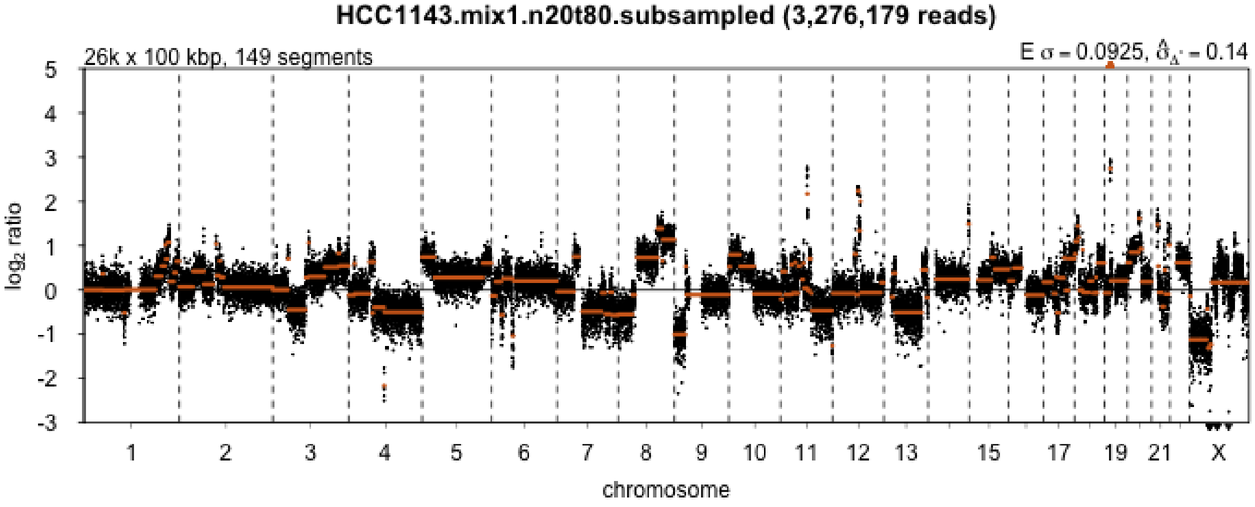
The next thing we will do is normalise the data and plot our data:

```r
x <- binReadCounts(bins, bamfiles=sample.files)
x.f <- applyFilters(x, residual=TRUE, blacklist=TRUE, chromosomes="Y")
x.f <- estimateCorrection(x.f)
cn <- correctBins(x.f)
cn <- normalizeBins(cn)
cn <- smoothOutlierBins(cn)
plot(cn)
```



Now we will segment the data using DNAcopy:

```
cn <- segmentBins(cn)
cn <- normalizeSegmentedBins(cn)
plot(cn)
```



☞You can increase the resolution reducing the bin size. Try that and compare the plots. **Use Case:** Now let's export the segmented means and the logratios to a file

```
exportBins(cn, file="LogRatios-Tumour.txt", format="tsv", type="copynumber")
exportBins(cn, file="SegmentedMeans.txt", format="tsv", type="segments")
exportBins(cn, file="SegmentedMeans.igv", format="igv", type="segments")
```

Use Case: Build a data.frame with the log-ratios and the segments together

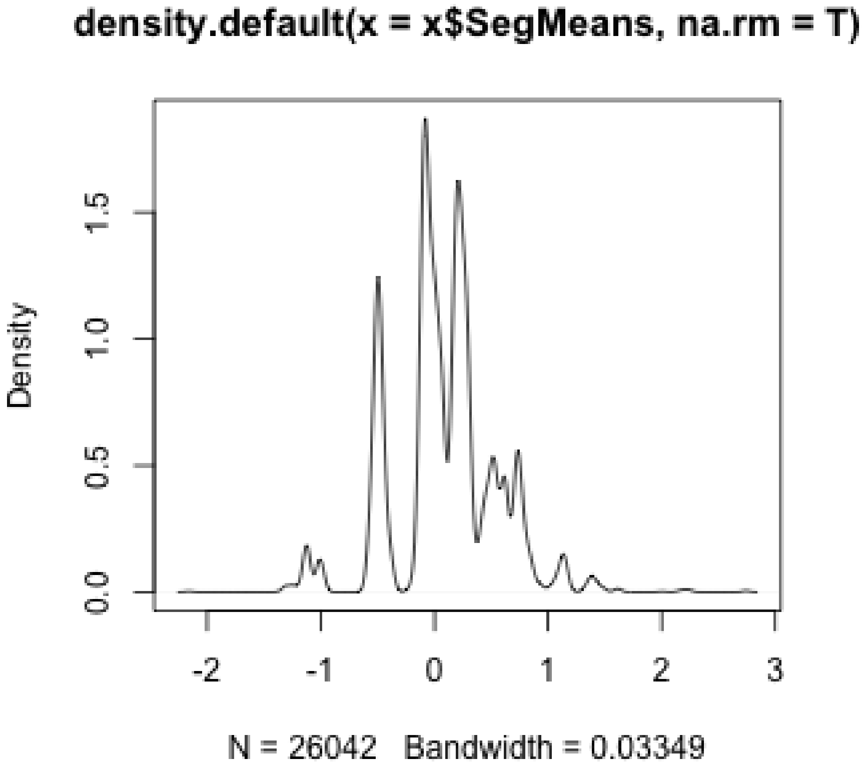
```
x <- read.table("LogRatios-Tumour.txt", sep="\t", header=T)
colnames(x)[5] <- "LogRatios"
y <- read.table("SegmentedMeans.txt", sep="\t", header=T)
colnames(y)[5] <- "SegMeans"
x <- merge(x, y)
x$chromosome <- factor(x$chromosome, levels=c(1:22, "X"))
x <- x[order(x$chromosome, x$start),]
```

Use Case: Now we are going to try to call copy numbers using the segmented means. First, we will remove some bad regions that QDNAseq didn't detect:

```
x$LogRatios[which(x$LogRatios< -3)] <- NA
x <- x[which(!is.na(x$LogRatios)),]
```

Now let's plot the density estimate of the segmented means:

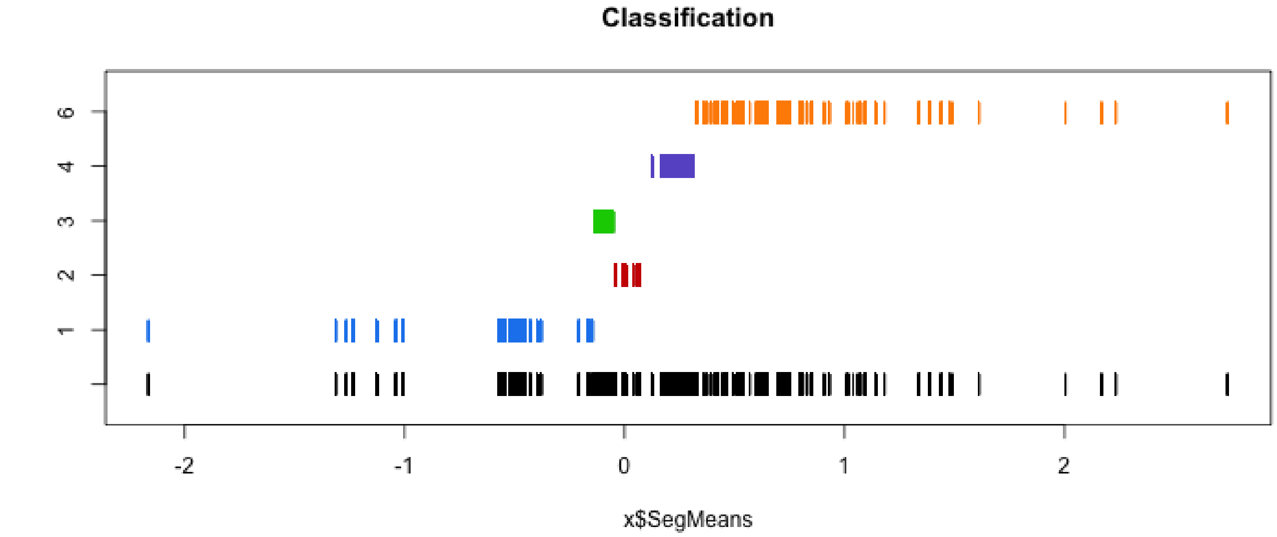
```
plot(density(x$SegMeans, na.rm=T))
```



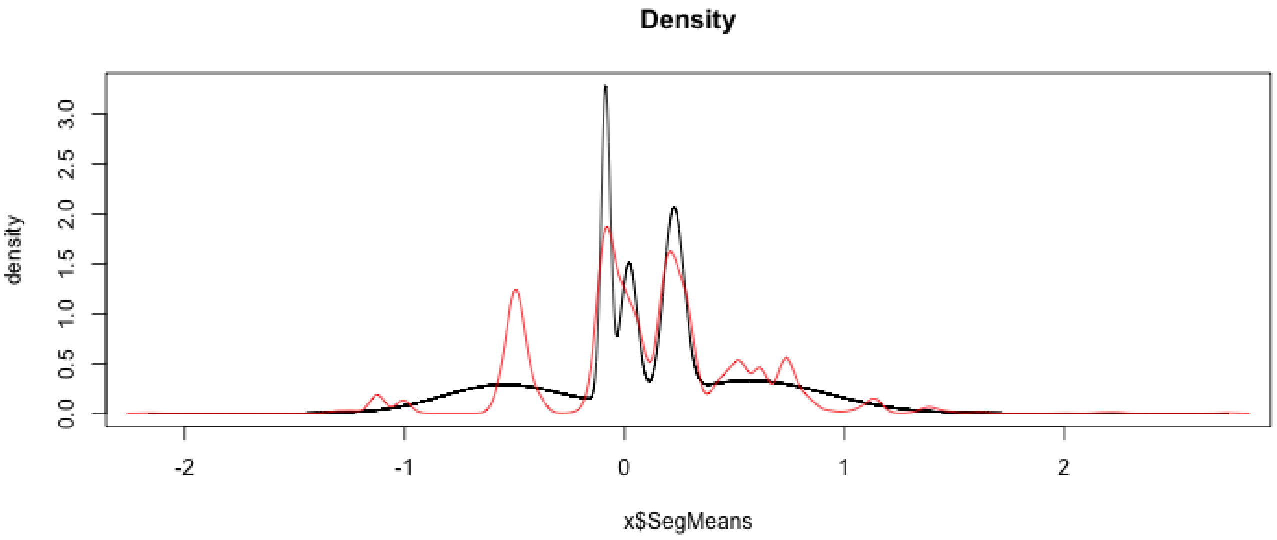
☞How many peaks do you see? How many different copy numbers do you see?

Use Case: We are going to fit manually a mixture model to classify each region into 6 different copy numbers with the package mclust:

```
library(mclust)
fit <- Mclust(x$SegMeans, 6, na.rm=T)
plot(fit, "classification")
```



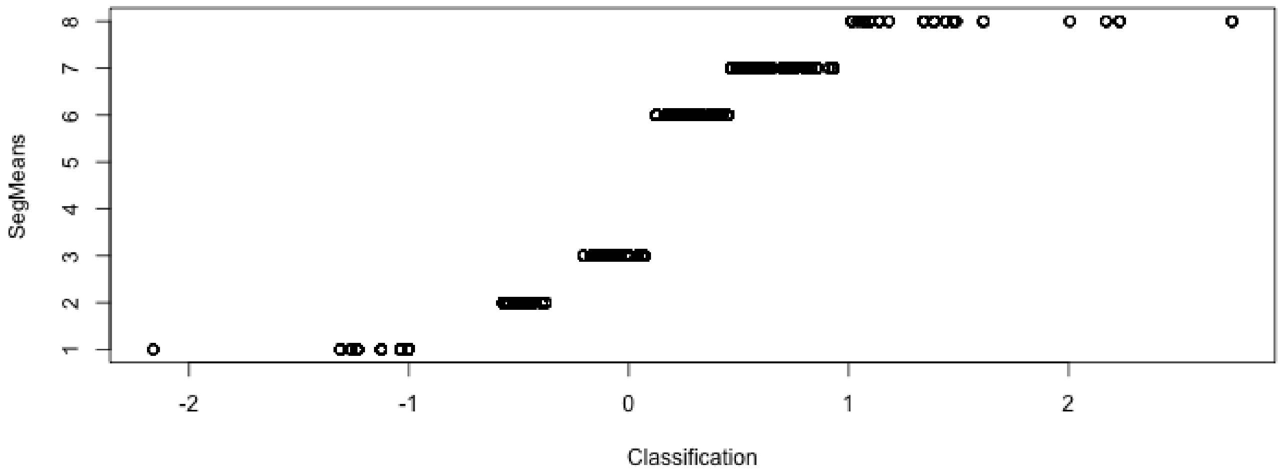
```
plot(fit, "density")
lines(density(x$SegMeans), col=2)
```



☞Does it look good? Why?

Use Case: We might need to constraint the model to mixtures with the same variance. You can do this setting the argument modelNames="E". Experiment with different number of components until you get a reasonable fit

```
plot(x$SegMeans, fit$classification, xlab="Classification", ylab="SegMeans")
```



Use Case: Finally, we add the information to the file

```
calls <- fit$classification
x$calls <- calls
```

☞But what are the copy numbers of each region?

☞Do the same with the normal sample: HCC1143.NORMAL.subsampled.bam. We don't need to call copy numbers for it, but save the logratios to another file (Logratios-Normal.txt).

☞We have a version of the tumour with more depth: HCC1143.mix1.n20t80.subsampledV2.bam. You can repeat the analysis with a larger nunmber of regions (reduce the bin size). Do you get better resolution? Do you gain more insight into the copy number landscape?

3 Analysis of Copy Number Data using depth and VAF

We are going to include the information on allelic frequencies in a series of SNPs on both the normal and the tumour cell line that we have previously computed. They are included in the files HCC1143_BAF_Normal.txt and HCC1143_BAF_Tumour.txt. We will use the ASCAT algorithm in the version that was originally designed for SNP microarrays. We need to read the source code:

```
source("ascat.R")
Normal <- read.table("HCC1143_BAF_ASCAT_Normal.txt", header=T, sep="\t")
Tumour <- read.table("HCC1143_BAF_ASCAT_Tumour.txt", header=T, sep="\t")
```

☞Note that the positions in both files are the same!

Now we need to create two more files for the log2ratios (that we already computed in the last section of the practical) in those positions in the normal and the tumour sample. We can do that with a loop, or much faster using Genomic Ranges:

```r
library(GenomicRanges)
CN.Normal <- read.table("LogRatios-Normal.txt", header=T, sep="\t",
                        stringsAsFactors=F)
BAF.Pos <- GRanges(seqnames=Normal$chrs, IRanges(start=Normal$pos, end=Normal$pos))
CN.Pos <- GRanges(seqnames=CN.Normal$chromosome, IRanges(start=CN.Normal$start,
                      end=CN.Normal$end))
tmp <- findOverlaps(BAF.Pos, CN.Pos)
res <- data.frame(chrs=Normal$chrs[queryHits(tmp)],
                  pos=Normal$pos[queryHits(tmp)])
res$HCC1143 <- CN.Normal[subjectHits(tmp),5]
rownames(res) <- paste0("SNV", rownames(res))
write.table(res, file="HCC1143_LogR_ASCAT_Normal.txt", sep="\t", quote=F)
Normal <- Normal[queryHits(tmp),]
rownames(Normal) <- paste0("SNV", 1:nrow(Normal))
write.table(Normal, file="HCC1143_BAF_ASCAT_Normal.txt", sep="\t", quote=F)

CN.Tumour <- read.table("LogRatios-Tumour.txt", header=T, sep="\t",
                        stringsAsFactors=F)
BAF.Pos <- GRanges(seqnames=Tumour$chrs, IRanges(start=Tumour$pos, end=Tumour$pos))
CN.Pos <- GRanges(seqnames=CN.Tumour$chromosome, IRanges(start=CN.Tumour$start,
                      end=CN.Tumour$end))
tmp <- findOverlaps(BAF.Pos, CN.Pos)
res <- data.frame(chrs=Tumour$chrs[queryHits(tmp)],
                  pos=Tumour$pos[queryHits(tmp)])
res$HCC1143 <- CN.Tumour[subjectHits(tmp),5]
rownames(res) <- paste0("SNV", rownames(res))
write.table(res, file="HCC1143_LogR_ASCAT_Tumour.txt", sep="\t", quote=F)
Tumour <- Tumour[queryHits(tmp),]
rownames(Tumour) <- paste0("SNV", 1:nrow(Tumour))
write.table(Tumour, file="HCC1143_BAF_ASCAT_Tumour.txt", sep="\t", quote=F)
```

And now we can start running ASCAT. First, we need to load the data and then plot it:

```r
ascat.bc <- ascat.loadData("HCC1143_LogR_ASCAT_Tumour.txt",
"HCC1143_BAF_ASCAT_Tumour.txt", "HCC1143_LogR_ASCAT_Normal.txt",
"HCC1143_BAF_ASCAT_Normal.txt",
                            chrs=c(1:22, "X"), gender="XX")
```

```
## [1] Reading Tumor LogR data...
## [1] Reading Tumor BAF data...
## [1] Reading Germline LogR data...
## [1] Reading Germline BAF data...
## [1] Registering SNP locations...
## [1] Splitting genome in distinct chunks...
```

```
  ascat.plotRawData(ascat.bc)
```

```
## [1] Plotting tumor data
## [1] Plotting germline data
```

☞Do you recognise any of the patterns seen in the Lecture?

Now we are going to segment both the log2ratios and the VAF plots. ASCAT uses its own segmentation method, ASPCF:

```
ascat.bc <-  ascat.aspcf(ascat.bc)
```

```
## [1] Sample HCC1143 (1/1)
```

```
ascat.plotSegmentedData(ascat.bc)
```

☞Inspect the plots. How do they look?

Finally, we run the ASCAT algorithm. We are changing the default gamma parameter, as it is related to the platform. The default value works well with arrays, but usually it is not adequate for sequencing data.

```
ascat.output <- ascat.runAscat(ascat.bc, gamma=1)
```

```
## [1] Sample HCC1143 (1/1)
```

☞Inspect the plots generated by ASCAT. Is the combination of ploidy/cellularity proposed a clear optimum? ☞This cell line is not a pure cell line, it is an insilico mixture with 80% tumour and 20% normal. Do the ASCAT results match?

☞You can save the copy numbers doing

```
segments.ascat <- data.frame(ascat.output$segments)
```