Calling copy-number using sWGS or SNP arrays

Dr Geoff Macintyre - geoff.macintyre@cruk.cam.ac.uk (mailto:geoff.macintyre@cruk.cam.ac.uk)

In this practical you will learn how to identify copy-number changes in a breast cancer cell-line (HCC1143) using two different datatypes: shallow whole-genome sequencing and SNP arrays. The code for each task has been hidden and will only be provided after the practical. Links have been provided to documentation that will assist you in carrying out each task. It is recommended you understand the commands being executed at each step, rather than simply cut and paste code.

Data

The following files will be required for this practical:

- HCC1143.mix1.n20t80.subsampled.bam - a bam alignment file from sWGS of HCC1143 cell-line
- HCC1143.mix1.n20t80.subsampled.bam.bai - bam index
- HCC1143.normal.BAF.txt - SNP6 HCC1143BL matched normal cell-line b-allele frequency
- HCC1143.normal.LogR.txt - SNP6 HCC1143BL matched normal cell-line logR
- HCC1143.tumor.BAF.txt - SNP6 HCC1143 cell-line b-allele frequency
- HCC1143.tumor.LogR.txt - SNP6 HCC1143 cell-line logR
- GC_AffySNP6_102015.txt - annotation file for GC correction
- sWGS_helper_functions.R - helper function for extracting segment table

You will find these files in the course material directory for Day2

Exercise 1: Relative copy-number calling using shallow whole-genome sequencing

Tasks:

1. Install the QDNAseq package (instuctions here (http://bioconductor.org/packages/release/bioc/html/QDNAseq.html)).

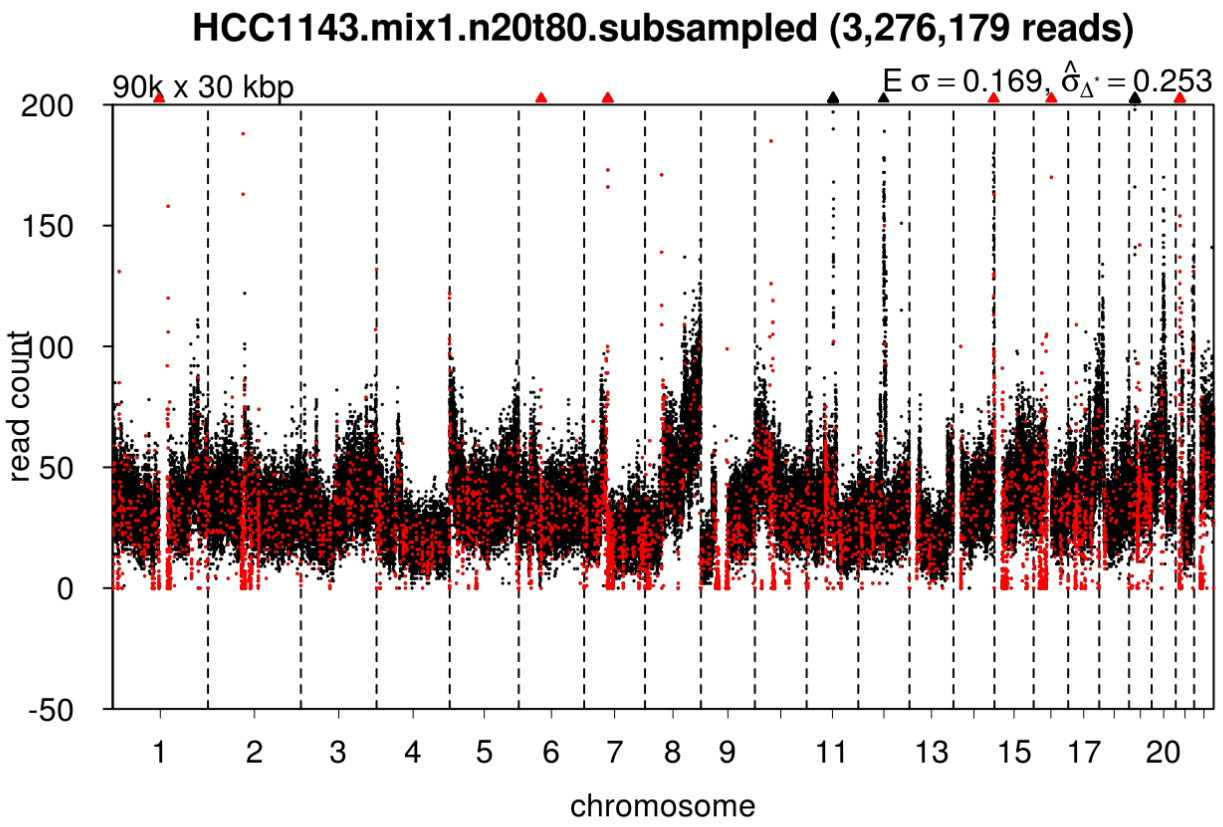
```
suppressMessages(source("https://bioconductor.org/biocLite.R"))
suppressMessages(biocLite("QDNAseq"))
suppressMessages(library(QDNAseq))
```

2. Get 30kb bin annotations for hg19 genome (instructions here (http://bioconductor.org/packages/release/bioc/vignettes/QDNAseq/inst/doc/QDNAseq.pdf)).

```
bins <- getBinAnnotations(binSize=30)
```

3. Plot the readcounts with filtered reads highlighted.

```
readCounts <- binReadCounts(bins,"HCC1143.mix1.n20t80.subsampled.bam")
plot(readCounts, logTransform=FALSE, ylim=c(-50, 200))
highlightFilters(readCounts, logTransform=FALSE,residual=TRUE, blacklist=TRUE)
```



4. Apply QDNAseq filters.

```
readCountsFiltered <- applyFilters(readCounts,residual=TRUE, blacklist=TRUE)
```

5. Calculate CG correction.

```
readCountsFiltered <- estimateCorrection(readCountsFiltered)
```

6. Apply GC correction.

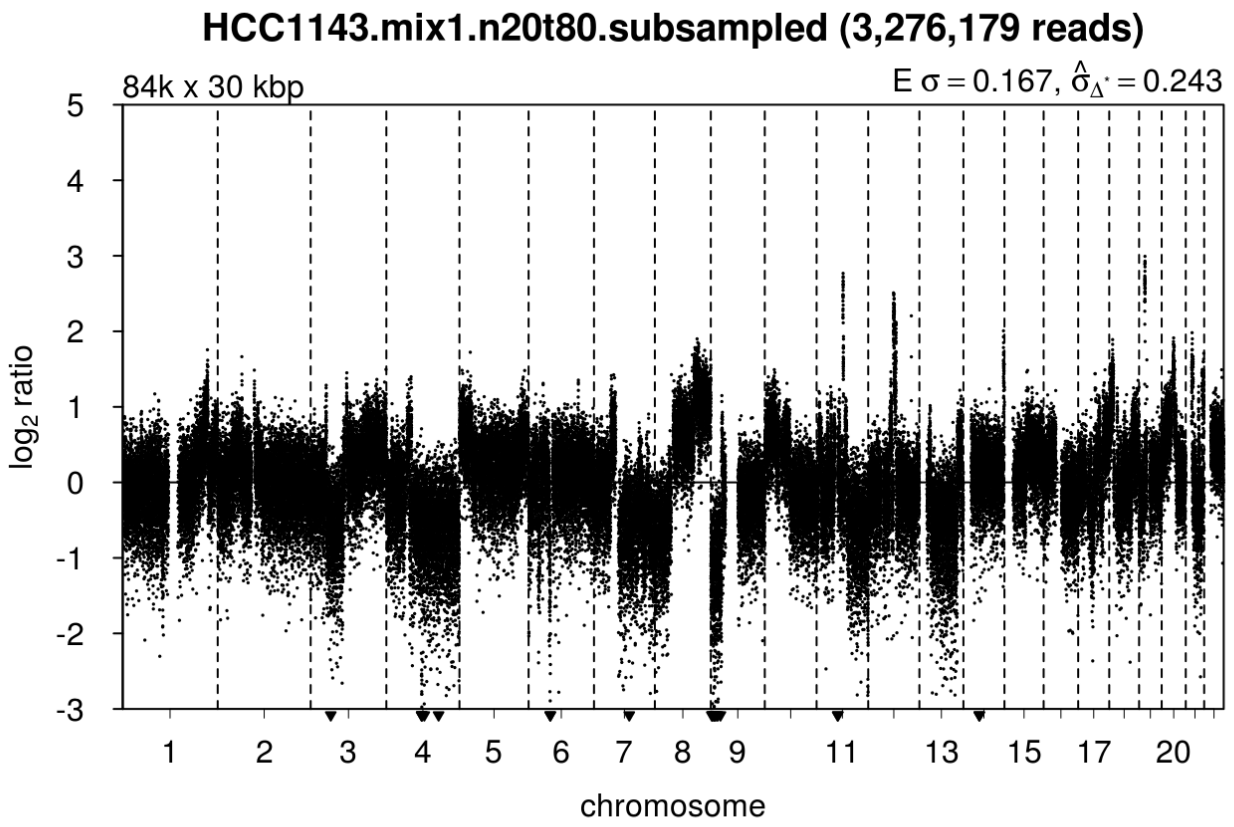
```
copyNumbers <- correctBins(readCountsFiltered)
```

7. Normalise and smooth outliers.

```
copyNumbersNormalized <- normalizeBins(copyNumbers)
copyNumbersSmooth <- smoothOutlierBins(copyNumbersNormalized)
```

8. Plot the smoothed copy-number.

```
plot(copyNumbersSmooth)
```

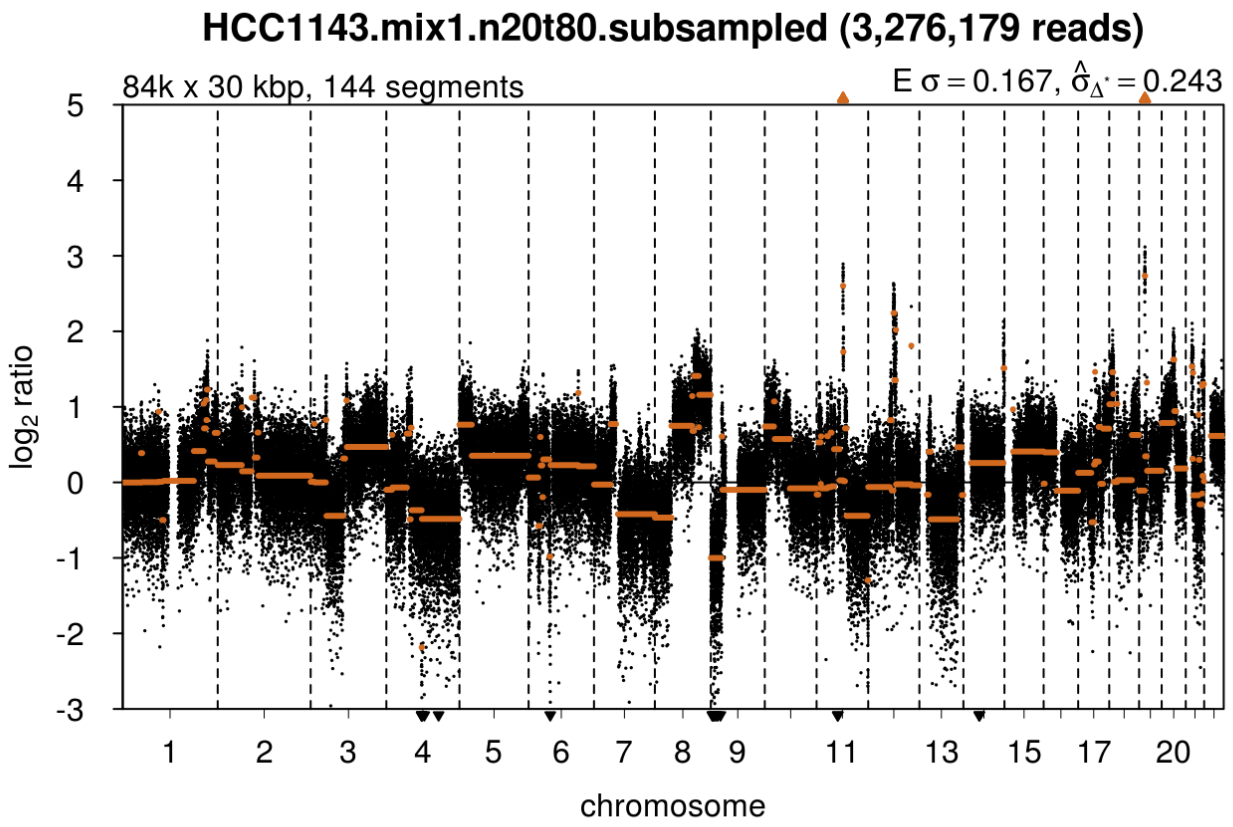


9. Segment the copy-number profile.

```
copyNumbersSegmented <- segmentBins(copyNumbersSmooth, transformFun="sqrt")
copyNumbersSegmented <- normalizeSegmentedBins(copyNumbersSegmented)
```

10. Plot the segmented profile.

```
plot(copyNumbersSegmented)
```



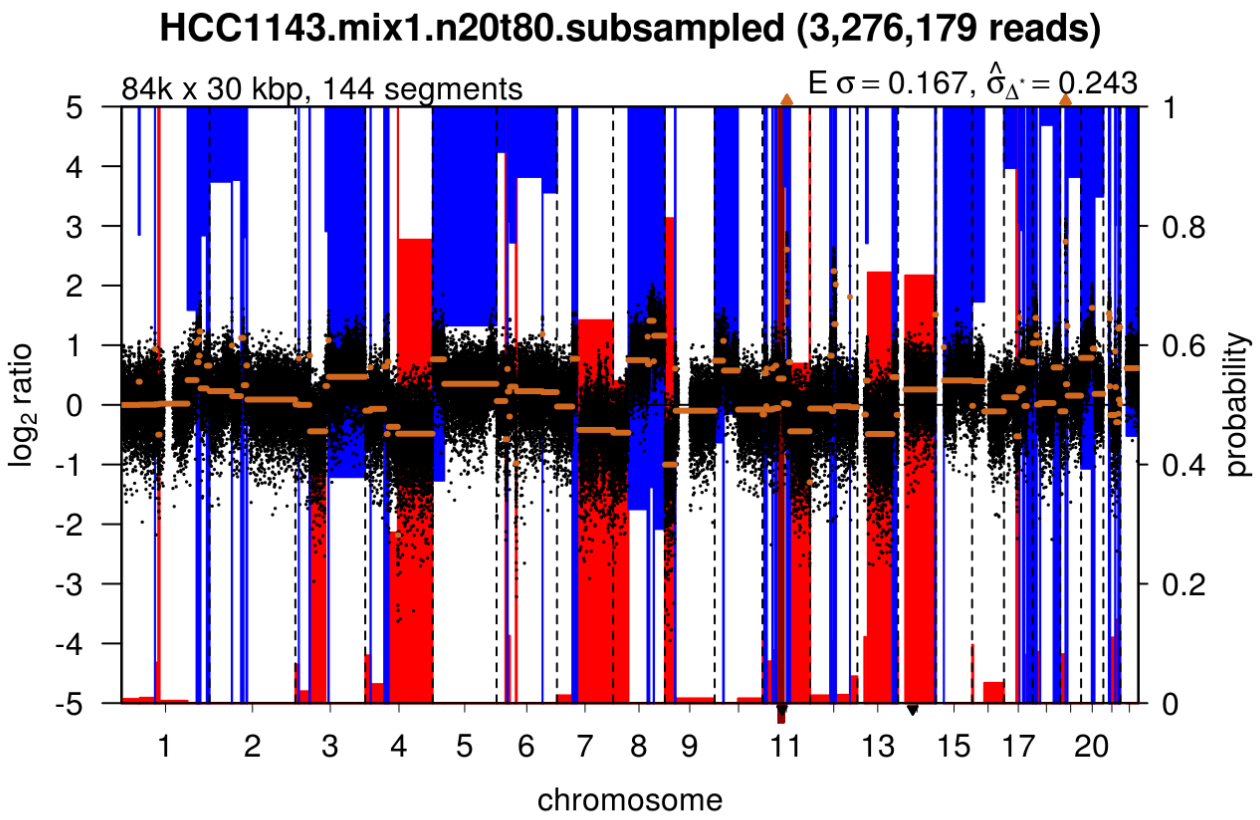
11. Call copy-number.

```
copyNumbersCalled <- callBins(copyNumbersSegmented)
```

```
## [1] "Total number of segments present in the data: 144"
## [1] "Number of segments used for fitting the model: 100"
```

12. Plot final profile.

```
plot(copyNumbersCalled)
```



Advanced: Output a table of all segments with a probability of loss is greater than 0.99

- Hint1: use expressionSet object functions in the Biobase package, fData and assayDataElement, to extract regions of interest.
- Hint2: use the function in sWGS_helper_function.R to get the segment table.

```
suppressMessages(library(Biobase))
filteredCN<-copyNumbersCalled[fData(copyNumbersCalled)$use,]
regions_of_loss<-filteredCN[assayDataElement(filteredCN,"probloss")>0.99,]
source("sWGS_helper_functions.R")
relative_loss<-getSegTable(regions_of_loss)
relative_loss
```

```
##   chromosome     start       end            segVal
## 1          1 103890001 107910000 0.707718979377022
## 2          4  60720001  63720000 0.712891086263012
## 3          4  92280001  92880000 0.220081424113795
## 4          6  24990001     3e+07 0.670923976797914
## 5          6  54270001  57180000 0.506766045567058
## 6         11 133410001 134940000 0.407123510573713
## 7         17  34830001  41370000 0.694018754502645
```

Excercise 2: Absolute copy-number calling using affy SNP6 chip

Tasks:

1. Download ASCAT here (https://www.crick.ac.uk/peter-van-loo/software/ASCAT).
2. Load the ascat R source

```
source("ascat.R")
```

3. Load the BAF and logR input files using ASCAT (instructions can be found here: https://www.crick.ac.uk/peter-van-loo/software/ASCAT (https://www.crick.ac.uk/peter-van-loo/software/ASCAT))

```
file.tumor.LogR <- dir(pattern="tumor.LogR")
file.tumor.BAF <- dir(pattern="tumor.BAF")
file.normal.LogR <- dir(pattern="normal.LogR")
file.normal.BAF <- dir(pattern="normal.BAF")
samplename <- sub(".tumor.LogR.txt", "", file.tumor.LogR)
ascat.bc <- ascat.loadData(file.tumor.LogR, file.tumor.BAF, file.normal.LogR, fil
e.normal.BAF, chrs=c(1:22))
```

```
## [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...
```

4. Apply ASCAT's GC wave correction

```
ascat.bc <- ascat.GCcorrect(ascat.bc, "GC_AffySNP6_102015.txt")
```

```
## [1] Sample HCC1143 (1/1)
## weighted correlation:  X25bp 0.108 ; X50bp 0.092 ; X100bp 0.089 ; X200bp 0.088
; X500bp 0.087 ; X1000bp 0.082 ; X2000bp 0.078 ; X5000bp 0.075 ; X10000bp 0.072 ;
X20000bp 0.071 ; X50000bp 0.072 ; X100000bp 0.072 ; X200000bp 0.073 ; X500000bp 0.
076 ; X1M 0.080 ; X2M 0.087 ; X5M 0.099 ; X10M 0.113 ;
## Short window size:  X25bp
## Long window size:  X10M
```

5. Plot the raw data

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

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

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

6. Segment and plot

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

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

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

7. Run ASCAT

```
ascat.output <- ascat.runAscat(ascat.bc)
```

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

```
saveRDS(ascat.output,"ascat.output.rds")
```

8. Inspect the output files. Does this look like a good purity fit? Is the profile accurate? Are there any unusual observations? (Hint: use the SKY karyotyping (http://www.pawefish.path.cam.ac.uk /BreastCellLineDescriptions/HCC1143.html) of this cell-line to help you).

Advanced: Extract regions of loss (compared to a diploid genome) from the ascat calls. Compare these to those obtained using the relative copy-number profile above. Do they agree?

- Hint1: all data is contained the object output by the ascat.runAscat function
- Hint2: regions of loss should include those at 1 and 0 copies

```
#get segments from ascat output
segTab<-ascat.output$segments

#extract regions where at least one copy has been lost
absolute_loss<-segTab[segTab$nMinor==0,]

#load granges library to assit with comparing genomic regions
library(GenomicRanges)

#get affected chromosomes
abs_gr<-GRanges(seqnames=absolute_loss$chr,IRanges(start=absolute_loss$startpos,en
d=absolute_loss$endpos))
rel_gr<-GRanges(seqnames=relative_loss$chromosome,IRanges(start=as.numeric(relativ
e_loss$start),
                                                          end=as.numeric(relative_
loss$end)))
setdiff(abs_gr,rel_gr)
```

```
## GRanges object with 233 ranges and 0 metadata columns:
##         seqnames                ranges strand
##            <Rle>             <IRanges>  <Rle>
##     [1]        1 [144007049, 150428819]      *
##     [2]        1 [150430959, 150727394]      *
##     [3]        1 [150729793, 152555527]      *
##     [4]        1 [152555706, 152586594]      *
##     [5]        1 [152590494, 152759678]      *
##     ...      ...                   ...    ...
##   [229]       21   [36202440, 36223627]      *
##   [230]       21   [36223706, 42133621]      *
##   [231]       21   [42443465, 43238695]      *
##   [232]       21   [43523941, 45479055]      *
##   [233]       21   [47527681, 48096957]      *
##   -------
##   seqinfo: 20 sequences from an unspecified genome; no seqlengths
```