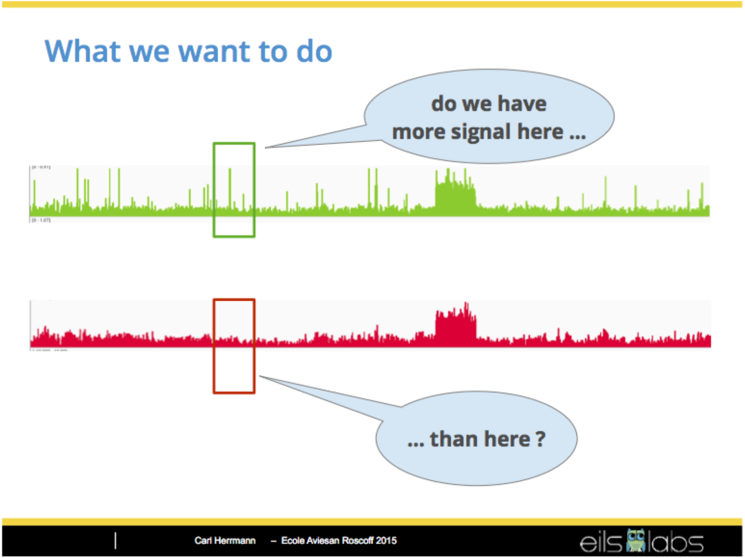
Practical: basic stats for peak calling

Jacques van Helden

2015-10-01

Peak-calling: Carl's question



What we want to do

do we have more signal here ...

... than here ?

Carl Herrmann – Ecole Aviesan Roscoff 2015

Defining the data directory

Open a connection to the lab web browser.

We will first define the URL from which the data can be downloaded, by concatenating the URL fo the course with the path to our dataset.

To concatenate paths, it is *recommended* to use the **R** command `file.path()`.

```
url.course <- "http://jvanheld.github.io/EBA15_stats_RStudi
url.data <- file.path(url.course, "practicals", "02_peak-ca
```

Loading a data table

R enables to download data directly from the Web.

Load counts per window in chip sample.

```r
## Define URL of the ChIP file
chip.bedg.file <- file.path(url.data, "FNR_200bp.bedg")

## Load the file content in an R data.frame
chip.bedg <- read.table(chip.bedg.file)

## Set column names
names(chip.bedg) <- c("chrom", "start", "end","counts")
```

Exploring a data frame: dim()

Before anything else, let us inspect the size of the data frame, in order to check that it was properly lodaded.

```r
dim(chip.bedg)
```

```
## [1] 23199     4
```

Checking the n first rows: head()

The function `head()` displays the first rows of a table.

```
head(chip.bedg, n = 5)
```

```
##                          chrom start  end counts
## 1 gi|49175990|ref|NC_000913.2|     0  200   1594
## 2 gi|49175990|ref|NC_000913.2|   200  400    834
## 3 gi|49175990|ref|NC_000913.2|   400  600    222
## 4 gi|49175990|ref|NC_000913.2|   600  800    172
## 5 gi|49175990|ref|NC_000913.2|   800 1000    123
```

Checking the n last rows: tail()

The function `tail()` displays the first rows of a table.

```r
head(chip.bedg, n = 5)
```

```
##                           chrom start  end counts
## 1 gi|49175990|ref|NC_000913.2|     0  200   1594
## 2 gi|49175990|ref|NC_000913.2|   200  400    834
## 3 gi|49175990|ref|NC_000913.2|   400  600    222
## 4 gi|49175990|ref|NC_000913.2|   600  800    172
## 5 gi|49175990|ref|NC_000913.2|   800 1000    123
```

Viewing a table

The function `View()` displays the full table in a user-friendly mode.

```
View(chip.bedg)
```

Selecting arbitrary rows

```r
chip.bedg[100:105,]
```

```
##                             chrom start   end counts
## 100 gi|49175990|ref|NC_000913.2| 19800 20000     21
## 101 gi|49175990|ref|NC_000913.2| 20000 20200      0
## 102 gi|49175990|ref|NC_000913.2| 20200 20400      0
## 103 gi|49175990|ref|NC_000913.2| 20400 20600    108
## 104 gi|49175990|ref|NC_000913.2| 20600 20800    229
## 105 gi|49175990|ref|NC_000913.2| 20800 21000    245
```

Selecting arbitrary columns

```
chip.bedg[100:105, 2]
```

```
## [1] 19800 20000 20200 20400 20600 20800
```

```
chip.bedg[100:105, "start"]
```

```
## [1] 19800 20000 20200 20400 20600 20800
```

```
chip.bedg[100:105, c("start", "counts")]
```

```
##     start counts
## 100 19800     21
## 101 20000      0
## 102 20200      0
## 103 20400    108
## 104 20600    229
## 105 20800    245
```

Adding columns

We can add columns with the result of computations from other columns.

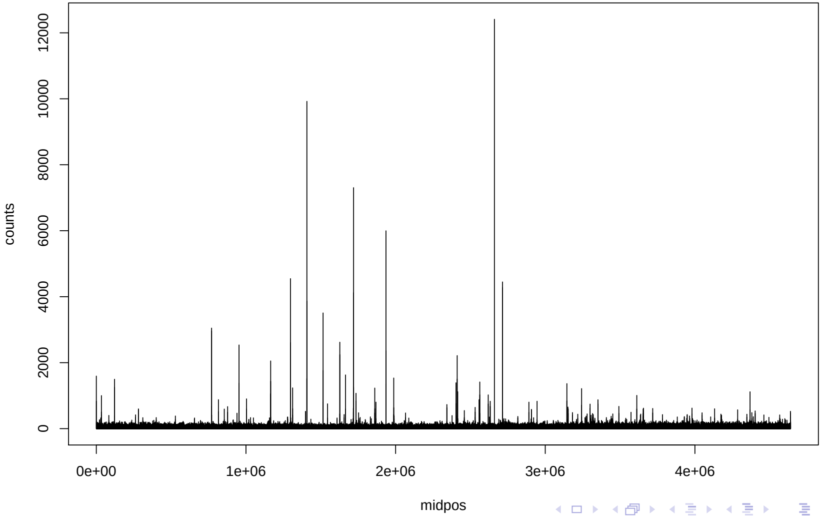
```
chip.bedg$midpos <- (chip.bedg$start + chip.bedg$end)/2
head(chip.bedg)
```

```
##                          chrom start  end counts midpos
## 1 gi|49175990|ref|NC_000913.2|     0  200   1594    100
## 2 gi|49175990|ref|NC_000913.2|   200  400    834    300
## 3 gi|49175990|ref|NC_000913.2|   400  600    222    500
## 4 gi|49175990|ref|NC_000913.2|   600  800    172    700
## 5 gi|49175990|ref|NC_000913.2|   800 1000    123    900
## 6 gi|49175990|ref|NC_000913.2|  1000 1200    116   1100
```

Plotting a density profile

We can readily print a plot with the counts per window.

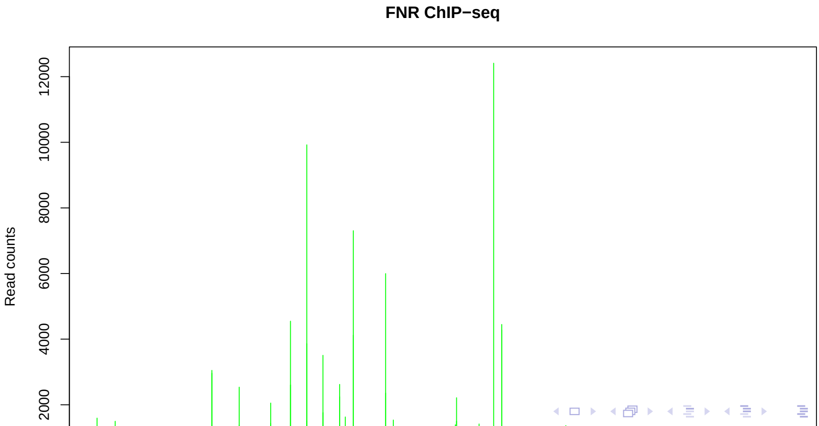
```
plot(chip.bedg[, c("midpos", "counts")], type="h")
```



Plotting a density profile

Let us improve the plot

```
plot(chip.bedg[, c("midpos", "counts")], type="h",
     col="green", xlab="Genomic position (200bp windows)",
     ylab= "Read counts",
     main="FNR ChIP-seq")
```



Exercise: exploring the background

We already loaded the count table for the FNR ChIP counts per window.

The background level will be estimated by loading counts per window in a genomic input sample. These counts are available in the same directory a file named `input_200bp.bedg`

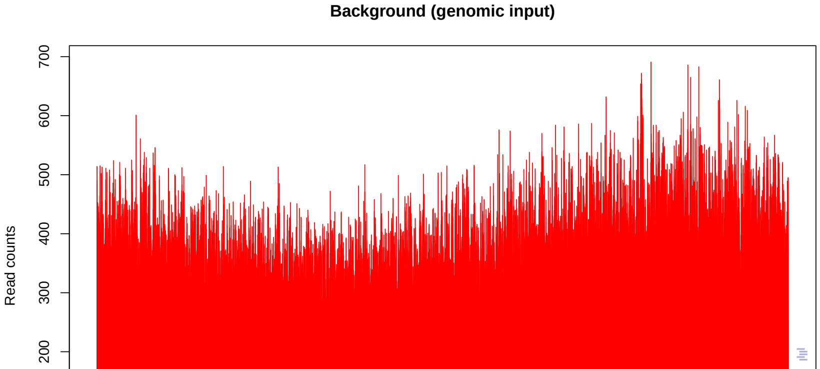
1. Load the counts per window in the input sample (genome sequencing).
2. Plot the density profile of the input
3. Compare chip-seq and input density profiles
4. Compare counts per window between chip-seq and input

Solution: loading the input counts per window

```r
## Define URL of the input file
input.bedg.file <- file.path(url.data, "input_200bp.bedg")

## Load the file content in an R data.frame
input.bedg <- read.table(input.bedg.file)

## Set column names
names(input.bedg) <- c("chrom", "start", "end","counts")
```

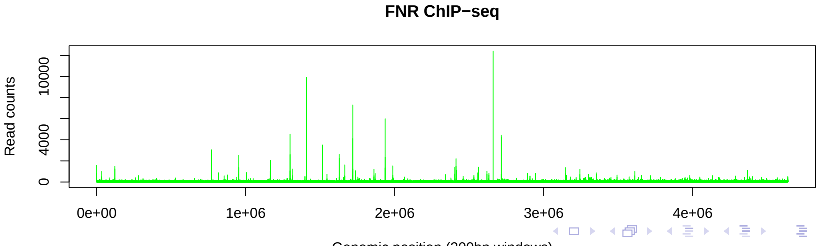
Solution: plotting the input density profile

```
## Compute middle positions per window
input.bedg$midpos <- (input.bedg$start + input.bedg$end)/2

plot(input.bedg[, c("midpos", "counts")], type="h",
     col="red", xlab="Genomic position (200bp windows)",
     ylab= "Read counts",
     main="Background (genomic input)")
```



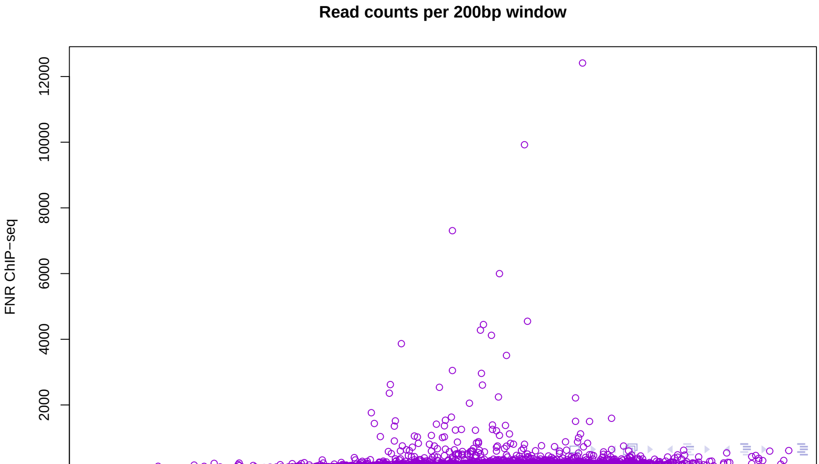
Solution: comparing chip-seq and background density profiles

```
par(mfrow=c(2,1)) ## Draw two panels on top of each other
plot(chip.bedg[, c("midpos", "counts")], type="h",
     col="green", xlab="Genomic position (200bp windows)",
     ylab= "Read counts",
     main="FNR ChIP-seq")
plot(input.bedg[, c("midpos", "counts")], type="h",
     col="red", xlab="Genomic position (200bp windows)",
     ylab= "Read counts",
     main="Background (genomic input)")
```



Solution: comparing counts per window between chip-seq and input

```
plot(input.bedg$counts, chip.bedg$counts, col="darkviolet",
     xlab="Genomic input", ylab="FNR ChIP-seq",
     main="Read counts per 200bp window")
```



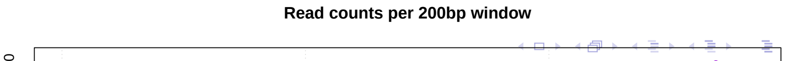
Solution: comparing counts per window between chip-seq and input

In order to better highight the dynamic range, we can use a log-based representation

```
plot(input.bedg$counts, chip.bedg$counts, col="darkviolet"
     xlab="Genomic input", ylab="FNR ChIP-seq",
     main="Read counts per 200bp window",
     log="xy")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 377 x
## from logarithmic plot

## Warning in xy.coords(x, y, xlabel, ylabel, log): 403 y
## from logarithmic plot
```

```
grid() ## add a grid
```



Read counts per 200bp window

Questions

- On the ChIP-seq versus input plot, how would you define peaks ?
- Where would you place the limit between peaks and background fluctuations ?

Exercises

1. Think about further drawing modes to improve your perception of the differences between signal and background.
2. We will formulate (together) a reasoning path to compute a p-value for each peak.