Easy visualization of the read coverage using the **CoverageView** package

Ernesto Lowy
European Bioinformatics Institute
EMBL

May 3, 2016

```
> options(width=40)
```

```
> library(CoverageView)
```

1 Introduction

This package is used for the visualization of the read coverage. It allows the user to calculate the read coverage and to generate different types of coverage displays. It is designed to be used with the files produced after the alignment of the reads generated in different types of next-generation sequencing experiments against the reference sequences (i.e. genome or transcriptome). CoverageView produces a visual representation of the genomic regions that are read-enriched and therefore have greater coverage depth; this is specially useful for ChIP-seq or genome-wide nucleosome positioning experiments, where the enrichment along a certain genomic region can be visualized in a single sample or across different samples. Additionally, CoverageView can also be used in RNA-seq experiments in order to compare how the coverage is distributed along the transcriptome.

2 Input files

CoverageView supports 2 types of files: BAM and bigWIG. The BAM format [**?**] is the compressed flavour of the SAM format and contains the result of the alignment of the short reads against a reference genome or transcriptome. The (bigWIG) format [**?**] is the indexed binary generated from a WIG file and is used to represent continuous data, as it is the case of the coverage values calculated for a certain genomic region.

2.1 CoverageBamFile and CoverageBigWigFile classes

The first step for using this package is to instantiate an object of the class CoverageBamFile or CoverageBigWigFile. An example of how a CoverageBamFile object is created is illustrated here:

```
> treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")
> trm<-CoverageBamFile(treatBAMfile,reads_mapped=28564510)
```

In this way we create an object of the class CoverageBamFile, specifying the URL of the BAM file containing the alignment. Additionally, we can use the reads_mapped argument in order to specify

the total number of reads aligned against the genome. This information is used with normalization purposes and if not provided then CoverageView is capable of calculating this number automatically.

In the case of the CoverageBigWigFile class, files in BigWIG format are accepted. An example of how to create a CoverageBigWigFile object pointing to a file in the BigWIG format is illustrated below:

```
> treatBigWIGfile<-system.file("extdata","treat.bw",package="CoverageView")
> trm<-CoverageBigWigFile(treatBigWIGfile,reads_mapped=28564510)
```

3 CoverageView is a useful tool for the analysis of ChIP-seq data

ChIP-seq experiments [**?**] combine chromatin immunoprecipitation with massively parallel DNA sequencing to analyze either proteins interacting with DNA or the distribution of the different epigenetic chromatin modifications.

In order to analyze the data produced in these types of experiments the first step consists on the alignment of the short-reads against the reference genome using one of the multiple short-read aligners available [**?**]. Then, the alignment file is passsed to an algorithm (MACS [**?**] for example) capable of detecting the regions of read-enrichment with respect to the basal signal for the genomic background in the same sample being analyzed or the enrichment over the same genomic region in a control sample. After identifying these enriched regions, MACS generates a BED file with the coordinates of these regions that, together with the alignment file in the BAM or BigWIG formats, can be fed into CoverageView.

3.1 Using CoverageView to analyze the ChIP-seq data for the FoxA1 transcription factor

3.1.1 Coverage profile around transcription start sites

In order to illustrate how CoverageView can be used to generate a coverage profile displaying the DNA binding pattern of a certain protein, we have used the FoxA1 transcription factor ChIP-seq experiment for the T-47D cell line, used as an illustrative example in the cited Nature protocol paper [**?**]. The Fastq files were aligned into the Human reference genome using Bowtie ([**?**]) and the alignment file in the BAM format was processed with MACS in order to identify peaks of coverage enrichment. Then, a BED format file ([**?**]) was generated containing the coordinates of the transcription start sites (TSS) of the genes that had a valid MACS peak in the region spanning $\pm$ 2500 nucleotides around the TSS. This region was arbitrarily made of this length, and its width will depend on the characteristics of the promoters of the genes regulated by the transcription factor that is being analyzed. In this way and using the right CoverageView function, one can generate several coverage profiles with different widths in order to define the right promoter dimensions.

Computing a coverage profile is straigthforward using CoverageView, it just requires a CoverageBamFile or CoverageBigWigFile object and a BED format file containing, in this case, the TSS for the set of genes obtained as explained above. An example of how this profile is computed is shown below:

```
> # get the FoxA1 BAM file for chr19
> FoxA1_chr19_BAM_url="http://www.ebi.ac.uk/~ernesto/FoxA1.chr19.bam"
> download.file(FoxA1_chr19_BAM_url,"./FoxA1.chr19.bam")
> # get the FoxA1 BAM index file
> FoxA1_chr19_BAI_url="http://www.ebi.ac.uk/~ernesto/FoxA1.chr19.bam.bai"
```

```
> download.file(FoxA1_chr19_BAI_url,"./FoxA1.chr19.bam.bai")
> ##
> # now instantiate a CoverageBamFile object
> trm<-CoverageBamFile("./FoxA1.chr19.bam",reads_mapped=168042)
> tss_chr19_bed<-system.file("extdata","FoxA1.chr19.bed",package="CoverageView")
```

Firstly, we instantiate an object called trm by passing to the class constructor the path to the BAM alignment file containing the alignment of the FoxA1 reads into chromosome 19. We are also providing the number of reads mapping the genome. This number will be used with normalization purposes and will be automatically calculated by the cov.matrix function if not provided. Additionally, we are also getting the path to the BED file with the TSS coordinates for the set genes having an enrichment peak called by MACS.

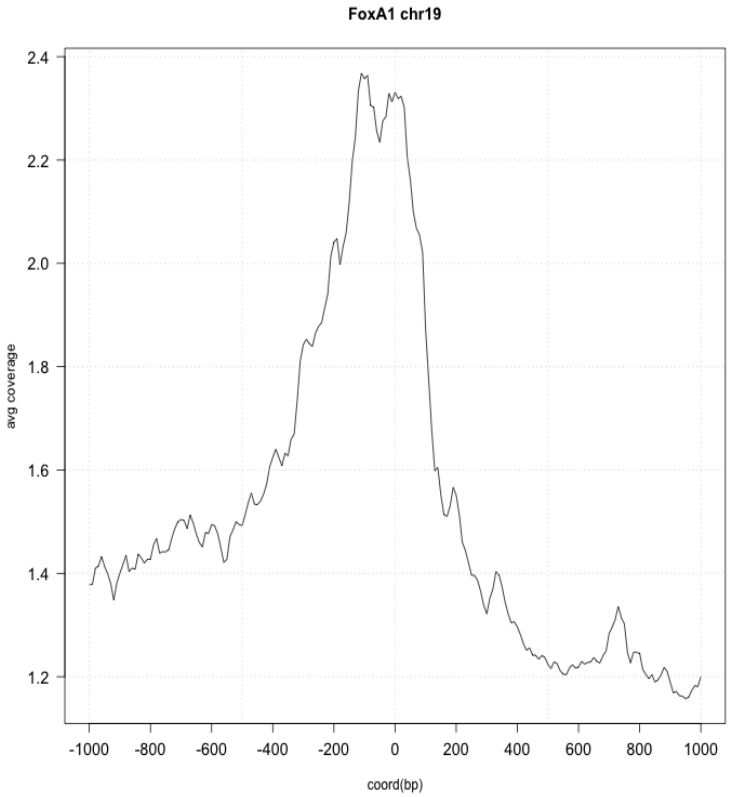
```
> FoxA1_chr19<-cov.matrix(trm,coordfile=tss_chr19_bed,extend=1000,num_cores=2,
+ bin_width=10)

> FoxA1_chr19[1:3,1:5]

     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    1    1    1    1    1
[3,]    1    1    1    1    1
```

Then, we pass to the cov.matrix function the trm CoverageBamFile object to analyze, the path to the BED file with the TSS coordinates and we use the extend parameter to set the length of the region around each TSS coordinate to 1000 nts on each side and for which the profile will be computed. Additionally, we use the num_cores option to use two processors for this specific analysis. Lastly, we use the bin_width option to split the genomic region being analyzed into bins of 10 nts for which the average coverage will be calculated.

The cov.matrix function generates a matrix in which each column represents the TSS for one of the selected genes and each row represents a certain bin within the selected genomic interval ($\pm$ 1000 nts around the TSS).

The last step in the generation of the profile consists on using the draw.profile function with the coverage matrix we have just created:

```
> draw.profile(FoxA1_chr19,ylab="avg coverage",outfile="FoxA1_chr19.png",
+ main="FoxA1 chr19")
```



By inspecting this coverage profile we can have a good impression of how the FoxA1 binding sites that are actually recognised by the transcription factor are distributed around the TSS.

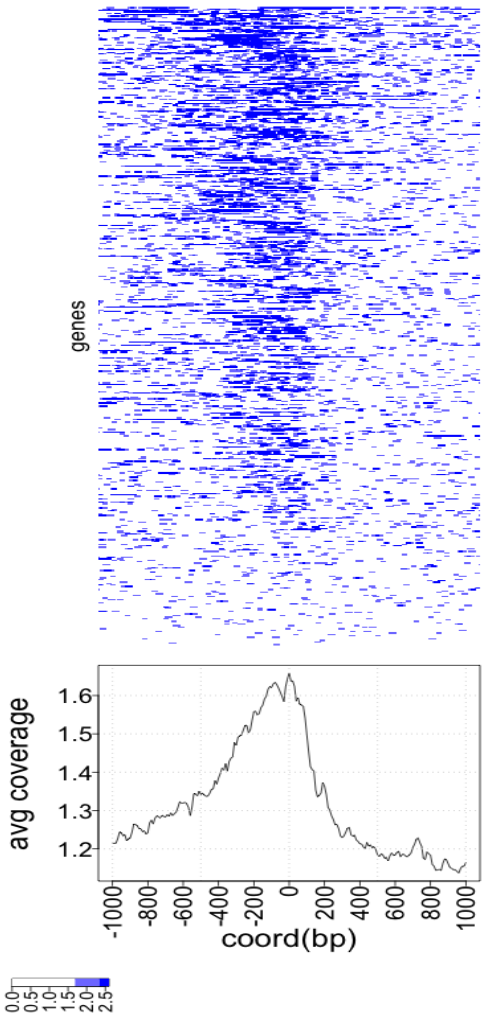
Finally, we can use the write.profile function to save the coverage matrix into a file:

```
> write.profile(FoxA1_chr19,outfile="FoxA1_chr19.txt")
```

3.1.2 Coverage heatmap around transcription start sites

Another way to represent visually the distribution of the coverage for a genomic region is to use a heatmap for which the intensity of the colours is directly correlated with the read depth. CoverageView offers this functionality by means of the draw.heatmap function. An example of how to use this function with the same FoxA1 matrix calculated in the previous section is illustrated below:

```
> draw.heatmap(FoxA1_chr19,outfile="FoxA1_chr19_heatmap.png")
```



3.2 Using CoverageView to analyze the H3K4me3 methylation data in a sample versus its control

ChIP-seq experiments can also be used to analyze the different histone modification patterns occurrying in the cell. Again, CoverageView can be of great help to generate a visual representation of how the signal is distributed along the promoter and gene body of the candidates for a certain epigenetic modification. As an example, we have used the ChIP-seq data designed to study the H3K4me3 methylation patterns in the UW_K562 cell line from the same Nature protocol paper [**?**] used in the previous section. This study consisted on one sample and its respective control so this time, we will use CoverageView in order to analyze the differences in the coverage enrichment for promoters and gene bodies between the sample and the control.

Firstly, the two FASTQ files were aligned into the Human reference genome using Bowtie [**?**] and then, the alignment file in the BAM format was processed with MACS in order to calls the peaks of coverage enrichment. After this, a BED format file was prepared with the start and end coordinates for all genes having a valid MACS peak in the region starting from 2500 nucleotides upstream the TSS and ending in the TES (transcription end site of each gene). For illustrative purposes, only the chr19 will be analyzed in the following example.

First, we get the file BAM files corresponding to the sample and the control:

```
> # get the sample BAM file for chr19
> H3K4me3_BAM_url="http://www.ebi.ac.uk/~ernesto/H3K4me3.chr19.bam"
> download.file(H3K4me3_BAM_url,"./H3K4me3.chr19.bam")
> # get also the index for the previous file
> H3K4me3_BAI_url="http://www.ebi.ac.uk/~ernesto/H3K4me3.chr19.bam.bai"
> download.file(H3K4me3_BAI_url,"./H3K4me3.chr19.bam.bai")
> #get the control BAM file for chr19
> H3K4me3_Control_BAM_url="http://www.ebi.ac.uk/~ernesto/H3K4me3_Control.chr19.bam"
> download.file(H3K4me3_Control_BAM_url,"./H3K4me3_Control.chr19.bam")
> # get also the index for the previous file
> H3K4me3_Control_BAI_url="http://www.ebi.ac.uk/~ernesto/H3K4me3_Control.chr19.bam.bai"
> download.file(H3K4me3_Control_BAI_url,"./H3K4me3_Control.chr19.bam.bai")
```

Then, we instantiate two CoverageBamFile objects corresponding to the sample and the control including also the the number of reads mapped. Again, this information is optional and if not provided then it will be automatically calculated by the cov.matrix function:

```
> trm<-CoverageBamFile("./H3K4me3.chr19.bam",reads_mapped=864924)
> ctl<-CoverageBamFile("./H3K4me3_Control.chr19.bam",reads_mapped=319369)
```

Now, we get the path to the BED file with the Human genes to analyze that were selected as explained previously:

```
> H3K4me3_chr19_bed<-system.file("extdata","H3K4me3.chr19.bed",
+ package="CoverageView")
```

This BED file contains the start and end coordinates of each of the genes to be analyzed and we can use it with the cov.matrix function in order to generate a coverage matrix. Before computing this matrix, each gene will be divided into the desired number of genomic bins as specified using the no_windows argument, each of the bins will have a width that will depend on the gene length. Finally, the average coverage for each bin will be calculated.

Additionally, with the offset argument we can determine for how many bins before the start and after the end coordinates the coverage should be calculated. This is useful if we want for example to extend the region further from the start coordinate in order to inspect how the coverage is distributed along the promoter.

Finally, we set the normalization argument to rpm, which stands for read per million normalization, to ensure that we can compare the coverages in the sample versus the control.

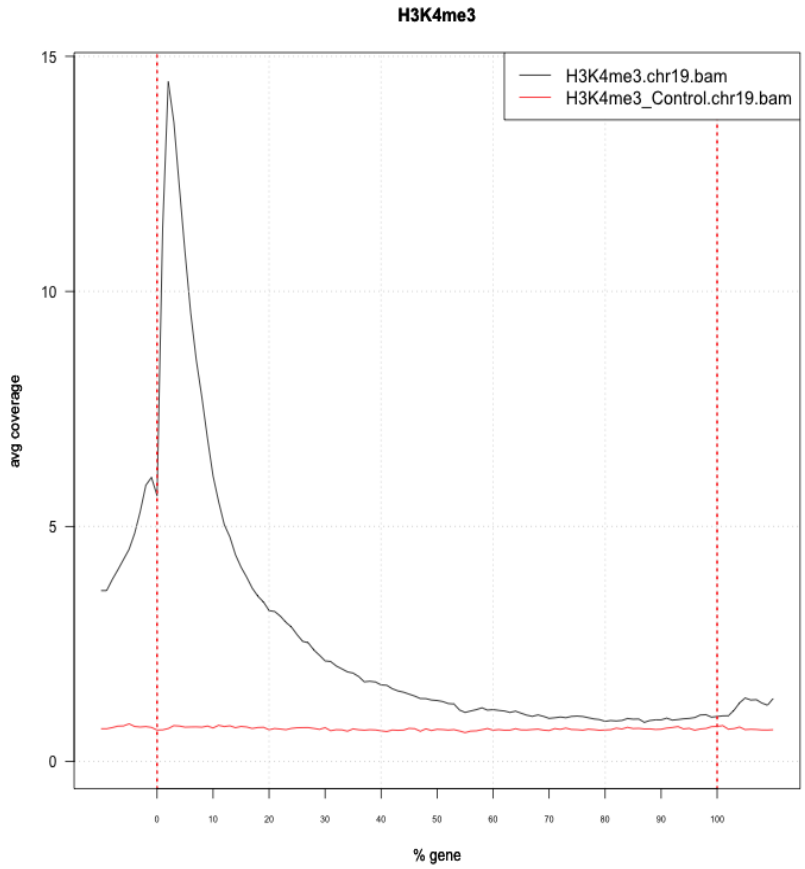
Both coverage matrices for the sample and the control respectively are generated by:

```
> DF_H3K4me3<-cov.matrix(trm,coordfile=H3K4me3_chr19_bed,no_windows=100,
+ offset=10,num_cores=2,normalization="rpm")
> DF_H3K4me3_ctl<-cov.matrix(ctl,coordfile=H3K4me3_chr19_bed,no_windows=100,
+ offset=10,num_cores=2,normalization="rpm")
```

In these matrices, each column is one the genes specified in the BED file and each of the rows is one of the genomic bins produced after dividing the genomic interval specified in the BED file into the number of windows specified by the no_windows parameter. Each element in the matrix will be the result of calculating the normalized average coverage for each bin.

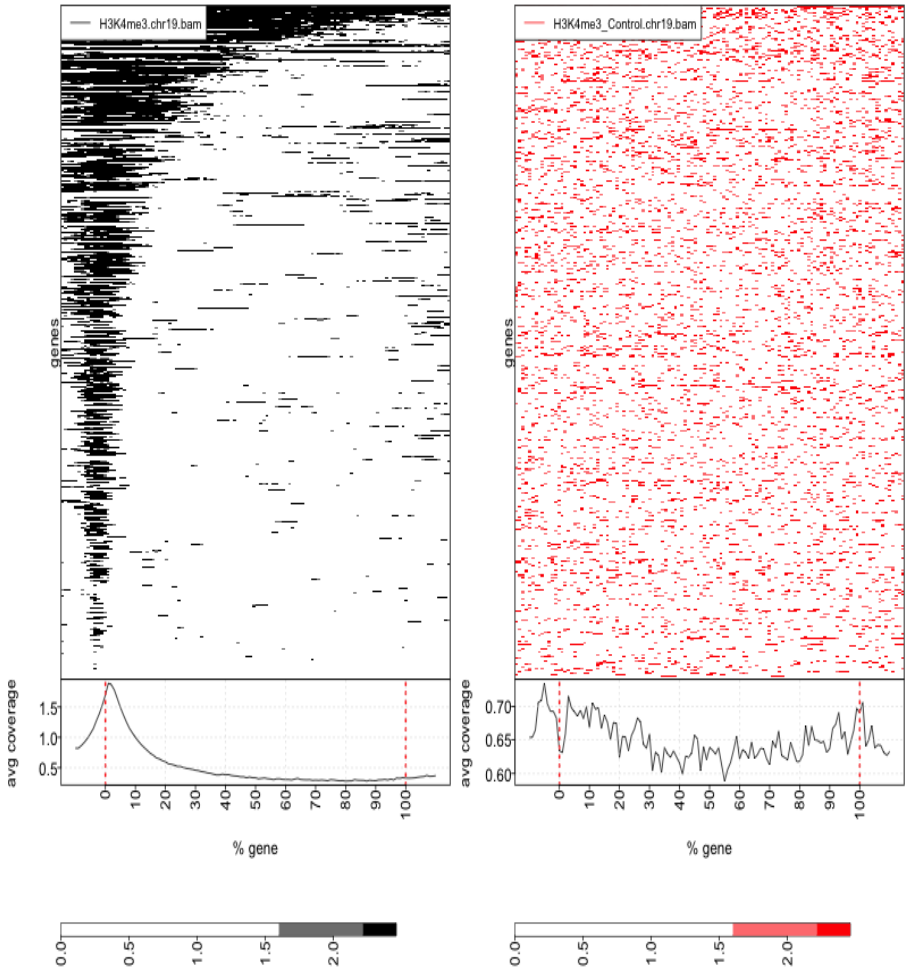
Finally, we can create a list with the two matrices and pass it to the draw.profile function in order to represent the two coverage matrices on the same axis. In this way, we can compare the coverage profiles for the sample and the control:

```
> input_list=list(DF_H3K4me3,DF_H3K4me3_ctl)
> draw.profile(data=input_list,ylab="avg coverage",outfile="H3K4me3cmp.png"
+ ,main="H3K4me3")
```



In the same way, we can also visualize the differences between the sample and the control by using the draw.heatmap function with the two matrices:

```
> input_list=list(DF_H3K4me3,DF_H3K4me3_ctl)
> draw.heatmap(data=input_list,outfile="H3K4me3cmp_heatmap.png")
```



3.3 Using the draw.heatmap function to compare the H3K4me3 versus the H3K36me3 methylation data

In this section, we will use the draw.heatmap function to compare two different histone epigenetic modifications: H3K4me3/H3K36me3 and also their respective controls. H3K4me3 was already used in the previous section, and the H3K36me3 ChIP-seq data was also described in the Nature Protocol paper [**?**] and analyzed with MACS following the same strategy than for H3K4me3. For this new data, a coverage matrix was produced using the cov.matrix function.

First, we get the coverage matrices for H3K36me3 and its control

```
> data(DF_H3K36me3)
> data(DF_H3K36me3_control)
```

Then, we create a list with the coverage matrices for H3K4me3/H3K36me3 and their controls and invoke the draw.heatmap function with this list:

```
> input_list=list(DF_H3K4me3,DF_H3K4me3_ctl,DF_H3K36me3,DF_H3K36me3_control)
> draw.heatmap(data=input_list,outfile="H3K4me3VSH3K36me3_heatmap.png")
```

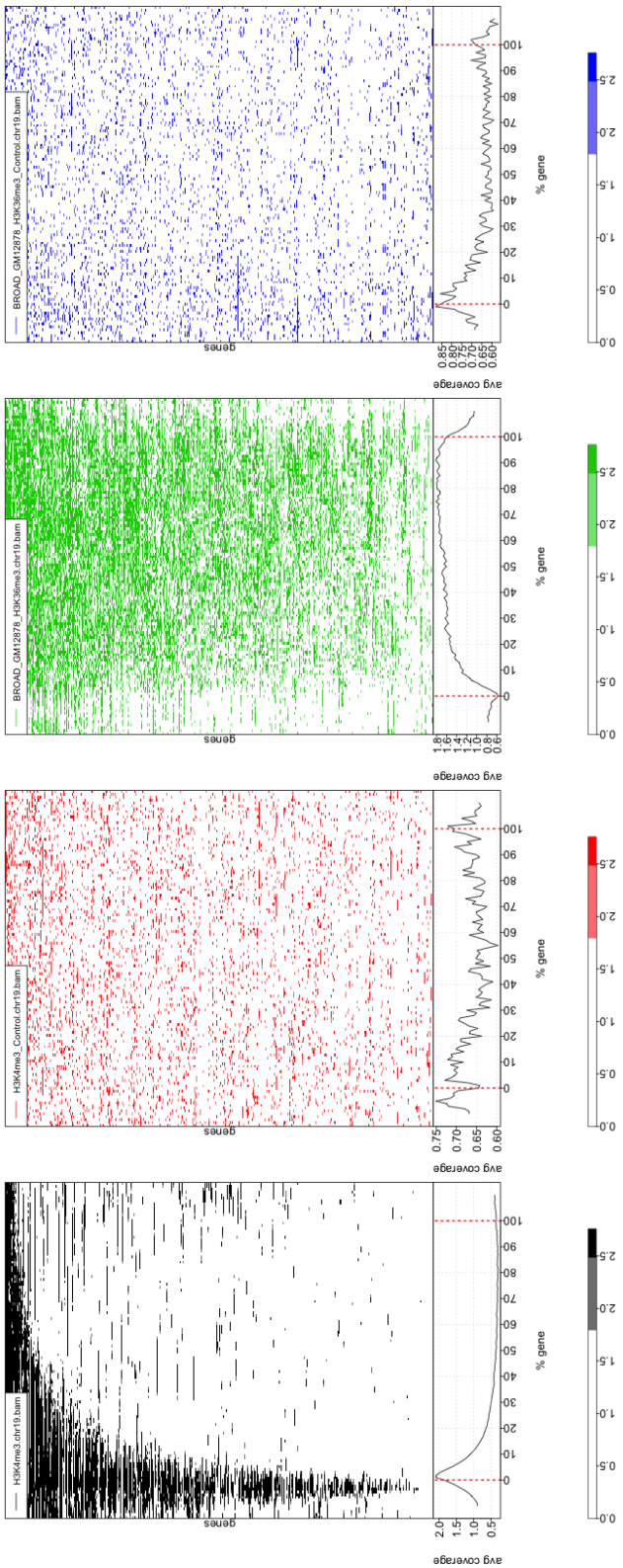


Figure 1: Coverage heatmaps for H3K4me3/H3K36me3 and their controls

3.4 Operating with 2 samples: ratios, log2ratios and substractions with the coverages

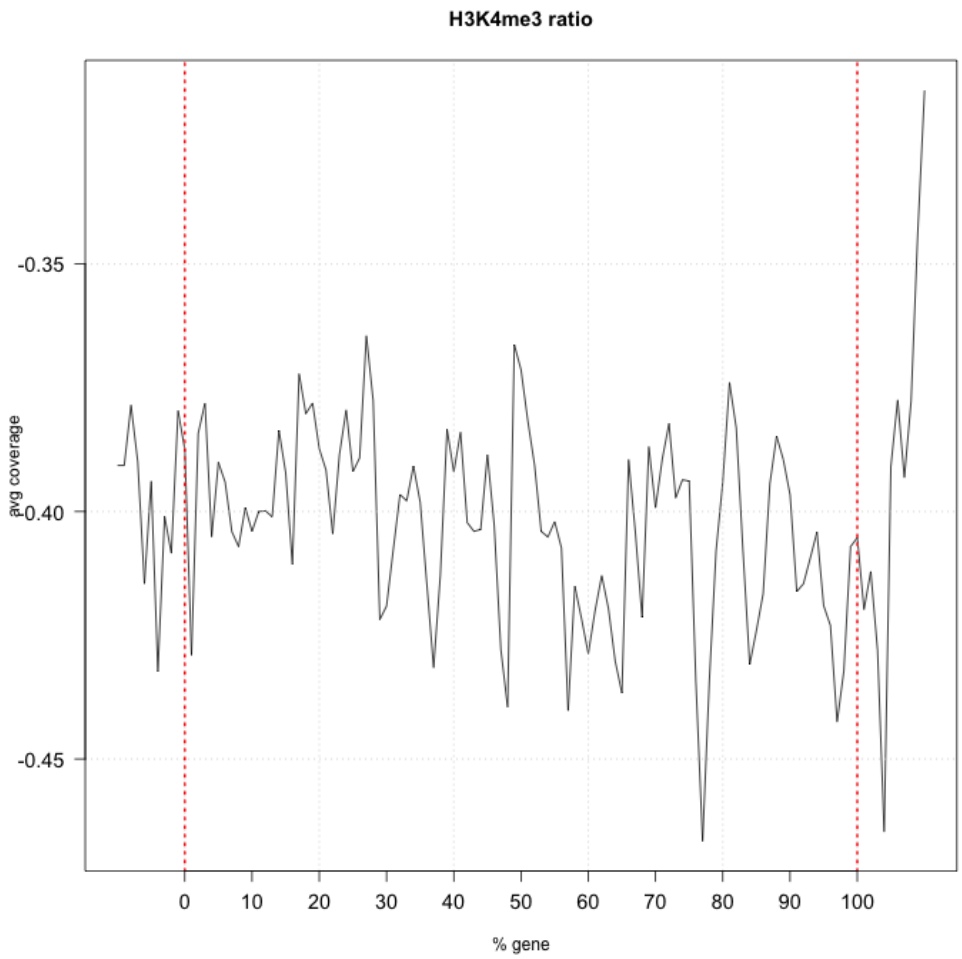
Another functionality of the cov.matrix function consists on the possility of operating with two different CoverageBamFile or CoverageWigFile objects simultaneously (i.e. one sample and one control respectively). In this way, we can generate a matrix for which each element is the result of calculating either the log2ratio/ratio/substraction of the coverage values in the sample versus the control.

In order to set the operation to perform we will use the do parameter; for example, using the same sample and control BAM files corresponding to the H3K4me3 ChIP-seq that was used previously and a BED file containing the start and end coordinates of the genes that did not have any MACS peak and therefore do not have the H3K3 tri-methylation, we can generate a matrix with the log2 ratio of the coverages:

```
> H3K4me3_chr19_nopeaks_bed<-system.file("extdata","H3K4me3.chr19.nopeaks.bed",
+ package="CoverageView")
> DF_H3K4me3_nopeaks_ratios=cov.matrix(trm,ctl,coordfile=H3K4me3_chr19_nopeaks_bed,
+ no_windows=100,offset=10,num_cores=2,normalization="rpm",do="log2ratio")
> save(DF_H3K4me3_nopeaks_ratios,file="DF_H3K4me3_nopeaks_ratios.RData")
```

And draw another profile:

```
> draw.profile(DF_H3K4me3_nopeaks_ratios,ylab="avg coverage",
+ outfile="H3K4me3nopeaks_ratios.png",main="H3K4me3 ratio")
```



Where it seems that there is no coverage enrichment for the genes provided in the BED file and therefore this particular H3K3 tri-methylation, associated in the literature with activated genes [**?**], is not present.

3.4.1 Operating with two samples: ratios, log2ratios and substractions for a particular region

The cov.matrix function requires a file with the coordinates of the genomic regions to analyze. However, sometimes one is interested in determining the region for which a certain arithmetic operation with the coverages will be calculated as the arguments of a function. For this, CoverageView has a method named cov.interval, and whose use is illustrated below using again the two CoverageBamFile objects from the H3K4me3 ChIP-seq experiment explained in the previous section.

First, we get the file BAM files corresponding to the sample and the control:

```
> # get the sample BAM file for chr19
> H3K4me3_BAM_url="http://www.ebi.ac.uk/~ernesto/H3K4me3.chr19.bam"
> download.file(H3K4me3_BAM_url,"./H3K4me3.chr19.bam")
> # get also the index file for previous file
> H3K4me3_BAI_url="http://www.ebi.ac.uk/~ernesto/H3K4me3.chr19.bam.bai"
> download.file(H3K4me3_BAI_url,"./H3K4me3.chr19.bam.bai")
```

```
> #get the control BAM file for chr19
> H3K4me3_Control_BAM_url="http://www.ebi.ac.uk/~ernesto/H3K4me3_Control.chr19.bam"
> download.file(H3K4me3_Control_BAM_url,"./H3K4me3_Control.chr19.bam")
> # get also the index file for previous file
> H3K4me3_Control_BAI_url="http://www.ebi.ac.uk/~ernesto/H3K4me3_Control.chr19.bam.bai"
> download.file(H3K4me3_Control_BAI_url,"./H3K4me3_Control.chr19.bam.bai")
```

Then, we instantiate the 2 objects:

```
> trm<-CoverageBamFile("./H3K4me3.chr19.bam",reads_mapped=864924)
> ctl<-CoverageBamFile("./H3K4me3_Control.chr19.bam",reads_mapped=319369)
```

And we calculate the ratio between the coverages in the trm and ctl samples for the region chr19:246050-247000:

```
> chr19_246050_247000.ratios=cov.interval(trm,ctl,bin_width=1,chr="chr19",
+ start=246050,end=247000,do='ratio')
```

cov.interval returns a vector that can be exported in the WIG format using the export.wig function:

```
> export.wig(chr19_246050_247000.ratios,outfile="chr19_246050_247000.ratios.wig")
```

Also, the coverage ratios can be calculated for the entire chromosome 19 by doing:

```
> chr19.ratios=cov.interval(trm,ctl,bin_width=1,chr="chr19",do='ratio')
```