The ChIP-seq quality Control package **ChIC**: A short introduction

April 30, 2018

Abstract

The **ChI**P-seq quality **C**ontrol package (ChIC) provides functions and data structures to assess the quality of ChIP-seq data. The tool computes three different categories of QC metrics: QC-metrics designed for narrow-peak profiles and general metrics, QC-metrics based on global read distribution and QC-metrics from local signal enrichments around annotated genes. User-friendly functions allow to perform the analysis with a single command, whereas step by step functions are also available for more experienced users. The package comes with a large reference compendium of precomputed QC-metrics from public ChIP-seq samples. Key features of the package are functions for calculating, visualizing and creating summary plots of QC-metrics, tools for the comparison of metagene profiles against reference profiles, tools for the comparison of single QC-metrics with the compendium values and finally a random forestl model to compute the single value quality score.

Contents

1 ENCODE Metrics (EM) based on sharp-peak profiles and cross-correlation analysis 3
1.1 Reading BAM files 4
1.2 Calculate QC-metrics from CrossCorrelation analysis 4
1.3 Remove anomalies in the read distrbution 5
1.4 Calculate QC-metrics from peak calls 7
1.5 Profile smoothing 7

2 Global enrichment profile Metrics (GM) and Fingerprint-plot 7

3 Metagene profiles and local enrichment profile metrics (LM) 8

4 Quality assessment using the compendium of QC-metrics as reference 9
4.1 Comparing local enrichment profiles 9
4.2 Comparing QC-metrics with reference values of the compendium 11
4.3 Assessing data quality with machine learning 13

References 14

To run the example code the user must provide 2 bam files: one for ChIP and one for the input". Here we used ChIP-seq data from ENCODE. Two example files can be downloaded using the following link:
https://www.encodeproject.org/files/ENCFF000BFX/
https://www.encodeproject.org/files/ENCFF000BDQ/

The tutorial illustrates output and input of ChIC functions using a ENCODE ChIP-seq dataset for H3K36me3 (ID: ENCFF000BFX) and its input (ID: ENCFF000BDQ).

For timing reasons in this tutorial we will start from already read bam file reads for a subset of chromosomes. Therefore we are loading input and chip data from our datapackage "ChIC.data":

```
> ##load ChIC
> library(ChIC)
> ## set path for working directory
> #filepath=tempdir()
> #setwd(filepath)
> filepath=getwd()
> #load tag-list with reads aligned to a subset of chromosomes
>
>
> data("chipSubset", package = "ChIC.data", envir = environment())
> chipBam=chipSubset
> data("inputSubset", package = "ChIC.data", envir = environment())
> inputBam=inputSubset
```

1 ENCODE Metrics (EM) based on sharp-peak profiles and cross-correlation analysis

qualityScores_EM is a wrapper function that reads the bam files and calculates a number of QC-metrics from cross-correlation analysis and from peak-calling. We will refer to this measures as ENCODE Metrics (EM).

```
> ##caluclate first set of QC-metrics: EM
> mc=3
> filepath=tempdir()
> setwd(filepath)
> system("wget
+ https://www.encodeproject.org/files/ENCFF000BFX/@@download/ENCFF000BFX.bam")
> system("wget
+ https://www.encodeproject.org/files/ENCFF000BDQ/@@download/ENCFF000BDQ.bam")
> chipName=file.path(filepath,"ENCFF000BFX")
> inputName=file.path(filepath,"ENCFF000BDQ")
> CC_Result=qualityScores_EM(chipName=chipName, inputName=inputName,
+ read_length=36, mc=mc)
> finalTagShift=CC_Result$QCscores_ChIP$tag.shift
```

The function expects two bam files: one for the immunoprecipitation (ChIP) and one for the control (Input). The read length (read_length parameter) can

be taken from the bam file itself. An additional option is the use of the 'mc' parameter, set to 1 per default. When changed it triggers the parallelization of a few processes and speeds up the calculations.

The function returns a number of QC-metrics, amongst others the "tag.shift" value which represents an input parameter for further steps (i.e. peak-calling and metagene calculation).

The wrapper is executing the following functions:

1.1 Reading BAM files

The first step in the *qualityScores_EM* function reads ChIP-seq data in .bam file-format. The function expects only the filename, that can also contain the pathname.

PLEASE NOTE: The following code chunk is executed if the user starts from the bam file. As we have already loaded the bam file, we skip this part.

```
> chipName=file.path(filepath,"ENCFF000BFX")
> inputName=file.path(filepath,"ENCFF000BDQ")
> chipBam=readBamFile(chipName)
> inputBam=readBamFile(inputName)
```

1.2 Calculate QC-metrics from CrossCorrelation analysis

The next function called is the *getCrossCorrelationScores* that calculates QC-metrics from the crosscorrelation analysis and other general metrics, e.g. the non-redundant fractions of mapped reads. An important parameter required by *getCrossCorrelationScores* is the binding-characteristics, calculated using *spp::get.binding.characteristics* function. The binding-characteristics structure provides information about the peak separation distance and the cross-correlation profile (for more details see [1]).

```
> cluster <- parallel::makeCluster( mc )
> ## calculate binding characteristics
>
> chip_binding.characteristics<-spp::get.binding.characteristics(
+     chipBam, srange=c(0,500), bin = 5, accept.all.tags = TRUE,
+     cluster = cluster)
> input_binding.characteristics<-spp::get.binding.characteristics(
+     inputBam, srange=c(0,500), bin = 5, accept.all.tags = TRUE,
+     cluster = cluster)
> parallel::stopCluster( cluster )
>

> ## calculate cross correlation QC-metrics
> crossvalues_Chip<-getCrossCorrelationScores( chipBam ,
+     chip_binding.characteristics, read_length = 36,
+     savePlotPath = filepath, mc = mc)
```

The output of the function is a list with calculated QC-metrics. Additionally we have to save the calculated tag.shift value for further steps:

```
> str(crossvalues_Chip)

List of 20
 $ CC_StrandShift                : num 195
 $ tag.shift                     : num 98
 $ N1                            : num 943998
 $ Nd                            : num 982966
 $ CC_PBC                        : num 0.96
 $ CC_readLength                 : num 36
 $ CC_UNIQUE_TAGS_LibSizeadjusted: num 982912
 $ CC_NSC                        : num 1.44
 $ CC_RSC                        : num 1.13
 $ CC_QualityFlag                : num 1
 $ CC_shift.                     : num 200
 $ CC_A.                         : num 0.256
 $ CC_B.                         : num 0.247
 $ CC_C.                         : num 0.178
 $ CC_ALL_TAGS                   : int 1025541
 $ CC_UNIQUE_TAGS                : int 982966
 $ CC_UNIQUE_TAGS_nostrand       : int 967887
 $ CC_NRF                        : num 0.958
 $ CC_NRF_nostrand               : num 0.944
 $ CC_NRF_LibSizeadjusted        : num 0.0983

> finalTagShift <- crossvalues_Chip$tag.shift
```

The user can also choose to calculate the cross-correlation for the input by simply using the following command:

```
> ## calculate cross correlation QC-metrics for input
> crossvalues_input <- getCrossCorrelationScores(inputBam,
+     chip_binding.characteristics, read_length = 36,
+     savePlotPath = filepath, mc = mc)
```

savePlotPath sets the path in which the cross-correlation plot (as pdf) should be saved. If not provided the plot will be forwarded to default DISPLAY. An example of a cross-correlation profile is shown in Figure 1.

getCrossCorrelationScores must be used on both, ChIP and Input.

1.3 Remove anomalies in the read distrbution

The data has to be procesed further using *removeLocalTagAnomalies* and removes local read anomalies like regions with extremely high read counts compared to the neighborhood (for more details see [1]).

```
> ##get chromosome information and order chip and input by it
> chrl_final <- intersect(names(chipBam$tags), names(inputBam$tags))
> chipBam$tags <- chipBam$tags[chrl_final]
> chipBam$quality <- chipBam$quality[chrl_final]
> inputBam$tags <- inputBam$tags[chrl_final]
> inputBam$quality <- inputBam$quality[chrl_final]
```

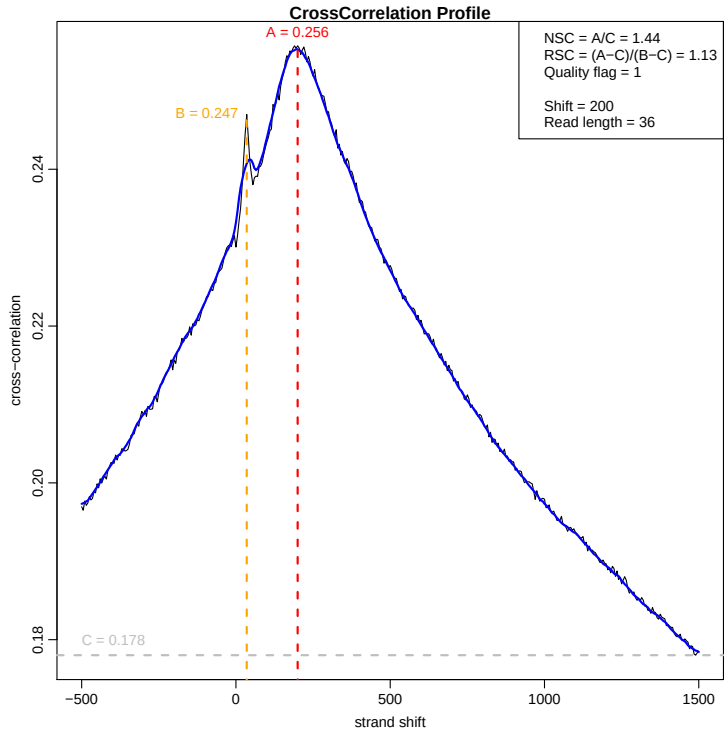


Figure 1: Cross-correlation plot for chromosome1 of the ChIP.

```
> ##remove sigular positions with extremely high read counts with
> ##respect to the neighbourhood
> selectedTags <- removeLocalTagAnomalies(chipBam, inputBam,
+     chip_binding.characteristics, input_binding.characteristics)
> inputBamSelected <- selectedTags$input.dataSelected
> chipBamSelected <- selectedTags$chip.dataSelected
```

1.4 Calculate QC-metrics from peak calls

The last set of QC-metrics are calculated on the number of called peaks using *getPeakCallingScores*.

```
finding background exclusion regions ... done
determining peaks on provided 1 control datasets:
using reversed signal for FDR calculations
bg.weight= 1.440726  excluding systematic background anomalies ... done
determining peaks on real data:
bg.weight= 0.6940943  excluding systematic background anomalies ... done
calculating statistical thresholds
FDR 0.01 threshold= 3.165151
finding background exclusion regions ... done
determining peaks on provided 1 control datasets:
using reversed signal for FDR calculations
bg.weight= 1.440726  excluding systematic background anomalies ... done
determining peaks on real data:
bg.weight= 0.6940943  excluding systematic background anomalies ... done
calculating statistical thresholds
E-value 10 threshold= 5.545838
```

1.5 Profile smoothing

The last step executed is the smoothing (using a Gaussian kernel) of the read profile to obtain the "tag density profile" (for more details see [1]). The read density profile is needed to calculate the next two categories of QC-metrics: the Global enrichment profile Metrics (GM) and the local enrichment profile metrics (LM).

```
> smoothedChip <- tagDensity(chipBamSelected,
+     tag.shift = finalTagShift, mc = mc)
> smoothedInput <- tagDensity(inputBamSelected,
+     tag.shift = finalTagShift, mc = mc)
```

2 Global enrichment profile Metrics (GM) and Fingerprint-plot

This set of QC-metrics is based on the global read distribution along the genome for ChIP and Input data [2]. The function *qualityScores_GM* reproduces the so-called Fingerprint plot (Figure 2) and returns a list of 9 QC-metrics that are taken from the cumulative distribution of the plot. Examples of these metrics

are the (a) fraction of bins without reads for ChIP and input, (b) the point of maximum distance between the ChIP and input (x-coordinate, y-coordinate for ChIP and input, the distance calculated as absolute difference between the two y-coordinates, the sign of the difference), (c) the fraction of reads in the top 1 percent of bins with highest coverage for ChIP and input.

```
> Ch_Results <- qualityScores_GM(densityChip = smoothedChip,
+     densityInput = smoothedInput, savePlotPath = filepath)
```

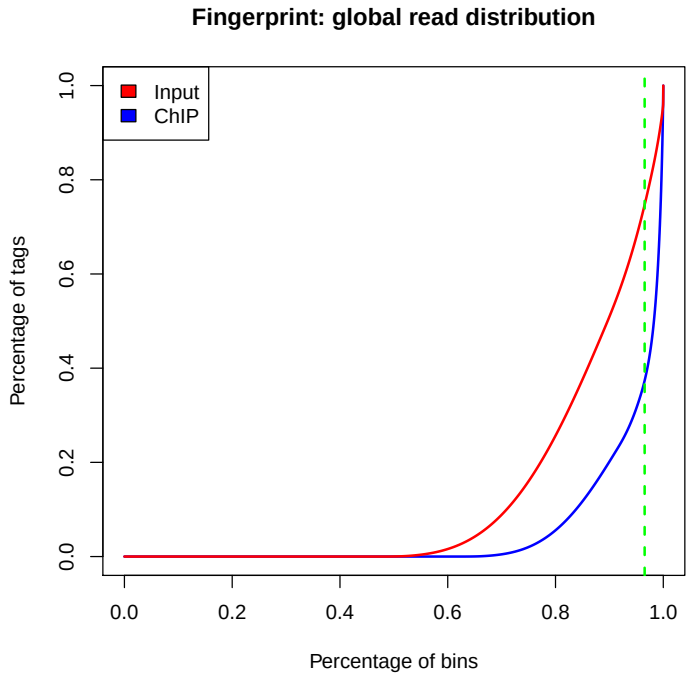


Figure 2: Fingerprint plot of sample ENCFF000BFX and its input ENCFF000BDQ.

3 Metagene profiles and local enrichment profile metrics (LM)

Metagene plots show the signal enrichment around a region of interest like the transcription start site (TSS) or over the gene body. ChIC creates two types of metagene profiles: a non-scaled profile for the TSS and transcription end site, and a scaled profile for the entire gene, including the gene body like in Figure **??**. For the metagene profile the tag density of the immunoprecipitation is taken over all RefSeg annotated human genes, averaged and log2 transformed.The same is done for the input (Figure **??** A). The normalized profile (Figure **??** B) is calculated as the signal enrichment (immunoprecipitation over the input)

and plotted on the y-axis, whereas the genomic coordinates of the genes like the TSS or regions up- and downstream are shown on the x-axis.

createMetageneProfile creates the metagene profiles for scaled and non-scaled profiles and returns a list with three items: "geneBody", "TSS" and "TES". Each item is again a list with the metagene-profiles for ChIP and input.

```
> Meta_Result <- createMetageneProfile(
+     smoothed.densityChip = smoothedChip,
+     smoothed.densityInput = smoothedInput,
+     tag.shift = finalTagShift, mc = mc)
```

The objects in 'Meta_Result' can be used to create the final metagene plots and to get the respective QC-values for the non-scaled profiles around the TSS and TES.

```
> TSS_Scores <- qualityScores_LM(data = Meta_Result$TSS, tag = "TSS",
+     savePlotPath = filepath)
> TES_Scores <- qualityScores_LM(data = Meta_Result$TES, tag = "TES",
+     savePlotPath = filepath)
```

The "geneBody" object can be used to plot the scaled metagene profile and to get its respective QC-values:

```
> #create scaled metagene profile
> geneBody_Scores <- qualityScores_LMgenebody(Meta_Result$geneBody,
+     savePlotPath = filepath)
```

4 Quality assessment using the compendium of QC-metrics as reference

The comprehensive set of QC-metrics, computed over a large set of ChIP-seq samples, constitutes in itself a valuable compendium that can be used as a reference for comparison to new samples.

ChIC provides tre functions for that:

- *metagenePlotsForComparison* to compare the metagene plots with the compendium
- *plotReferenceDistribution* to compare a QC-metric with the compendium values
- *predictionScore* to obtain a single quality score from the previously computed QC-metrics

4.1 Comparing local enrichment profiles

The *metagenePlotsForComparison* function is used to compare the local enrichment profile to the reference compendium by plotting the metagene profile against the expected metagene for the same type of chromatin mark **??**. The expected metagene is provided by the compendium mean (black line) and standard error (blue shadow) shown in Figure 5.

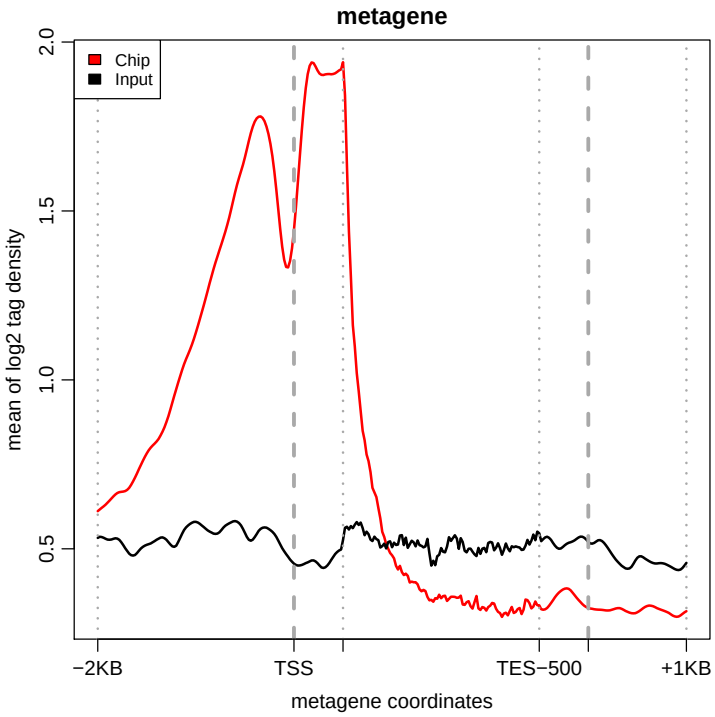
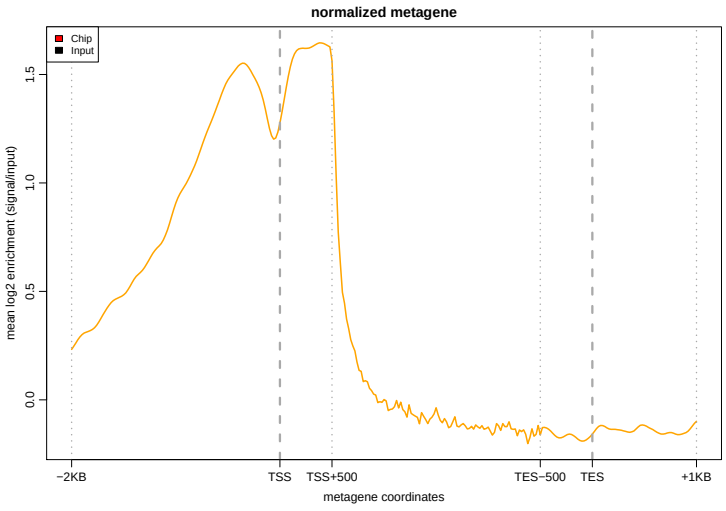


Figure 3: Metagene profiles: Signal enrichment for ChIP and Input along the gene body.



.

Figure 4: Normalized metagene profile: signal enrichment for ChIP over Input along the gene body.

```
> metagenePlotsForComparison(data = Meta_Result$geneBody,
+     chrommark = "H3K4me3",
+     tag = "geneBody",
+     savePlotPath = filepath)
> metagenePlotsForComparison(data = Meta_Result$TSS,
+     chrommark = "H3K4me3",
+     tag = "TSS",
+     savePlotPath = filepath)
```

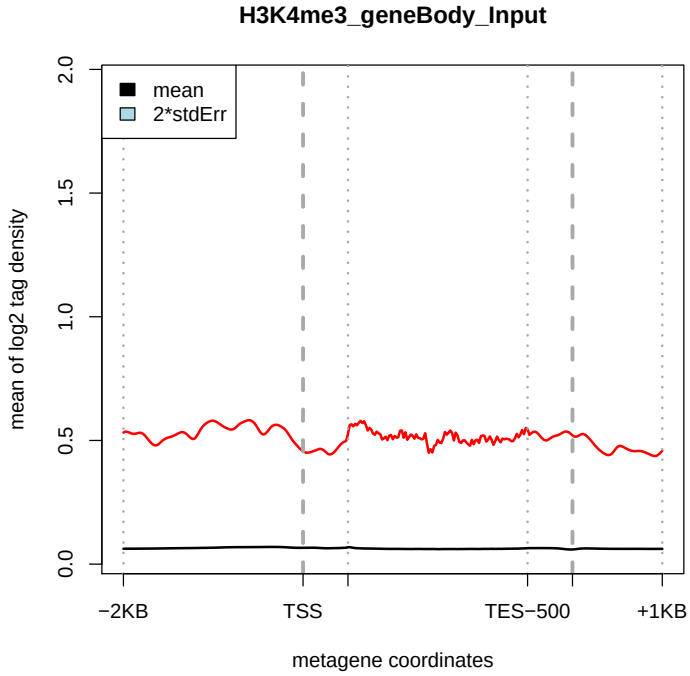


Figure 5: Enrichment profile and QC-metrics can be plotted against the pre-computed profiles of the compendium. The metagene profile shows the sample signal (red line) for the input when compared to the compendium mean signal (black line) and its 2x standard error (blue shadow).

4.2 Comparing QC-metrics with reference values of the compendium

The plot against the reference compendium of metrics add an extra level of information that can be easily used by less experienced users. Indeed, the function *plotReferenceDistribution* is helpful to visually compare the characteristics of an analysed sample with a large number of already published data (Figure 5).

```
> plotReferenceDistribution(chrommark = "H3K4me3",
+     metricToBePlotted = "RSC",
```

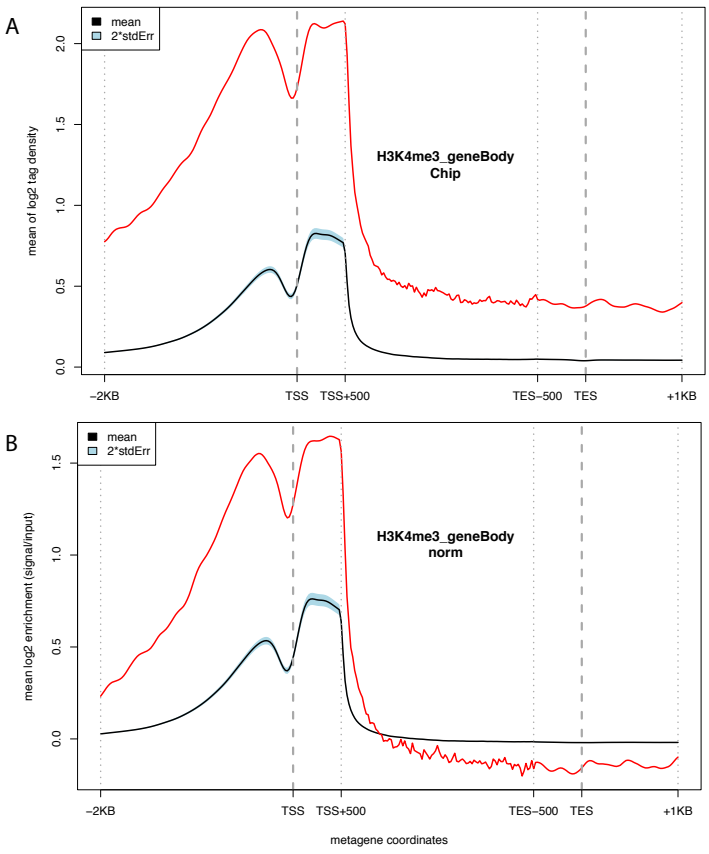


Figure 6: Enrichment profile plotted against the pre-computed profiles of the compendium. The metagene profiles show the sample signal (red line) for the ChIP (A) and the enrichment (chip over input) (B) when compared to the compendium mean signal (black line) and its 2x standard error (blue shadow).

```
+     currentValue = crossvalues_Chip$CC_RSC,
+     savePlotPath = filepath)
```

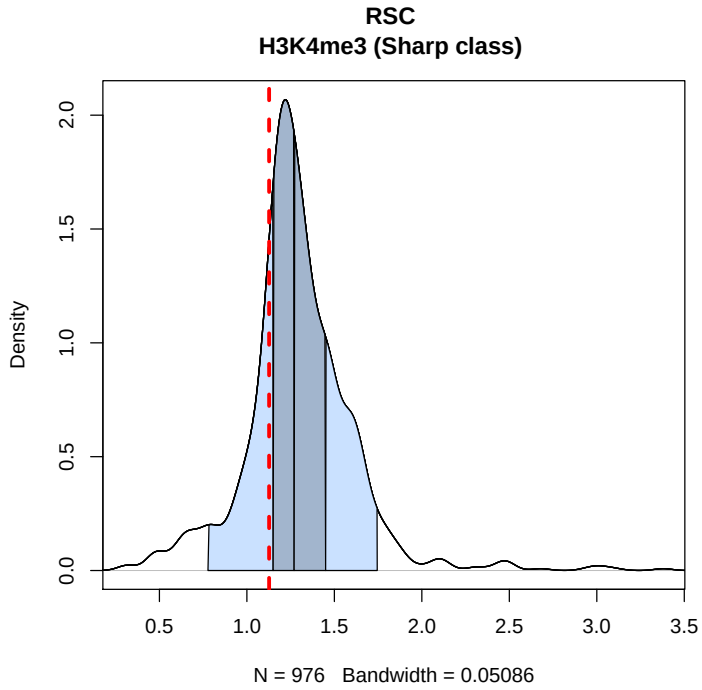


Figure 7: QC-metric of newly analysed ChIP-seq sample can be compared to the reference values of the compendium. The density plot shows the QC-metric RSC (red dashed line) of the current sample.

4.3 Assessing data quality with machine learning

Moreover, we used the compendium of metrics to train a machine learning model that summarizes the sample quality in a single score.

```
> te <- predictionScore(chrommark = "H3K4me3",
+     features_cc = CC_Result,
+     features_global = Ch_Results,
+     features_TSS = TSS_Scores,
+     features_TES = TES_Scores,
+     features_scaled = geneBody_Scores)
> print(te)

[1] 0.498
```

References

[1] Peter V Kharchenko, Michael Y Tolstorukov, and Peter J Park. Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nat. Biotechnol.*, 26(12):1351–1359, 2008.

[2] Aaron Diaz, Abhinav Nellore, and Jun S Song. CHANCE: comprehensive software for quality control and validation of ChIP-seq data. *Genome Biol.*, 13(10):R98, 2012.