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

March 2, 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, including the comparison of metagene profiles against reference profiles, and the comparison of single QC-metrics with the compendium values.

Contents

1 Installation

Prerequisites of CHIC are the installation of the data package *chic.data* via Bioconductor and spp

```
> install.packages("spp",dependencies=TRUE)
```

The tutorial shows example outputs and usage of several functions using an example dataset from ENCODE for H3K36me3 (ID: ENCFF000BLL) and its input (ID: ENCFF000BKA) (https://www.encodeproject.org/). The sample has been flagged by ENCODE with a red audit because of low read depth.

2 QC-metrics based on sharp-peak profiles and cross-correlation analysis

QC_scoresCC_PC is a wrapper function that reads the bam files and calculates a number of QC-metrics from cross-correlation analysis and from peak-calling.

```
chipName="ENCFF000BLL"
inputName="ENCFF000BKA"

CC_Result=QC_scoresCC_PC(chipName=chipName,
    inputName=inputName,
    read_length=36,
    dataPath=getwd(),
    annotationID="hg19",
    savePlotPath=getwd())
```

The wrapper is based on the following functions:

2.1 Reading BAM files

The first step reads ChIP-seq data in .bam file-format. The function expects exactly two bam files: one for the immunoprecipitation (ChIP) and one for the control (Input). The "dataPath" should contain the path to the directory in which the bam files are stored.

```
chip.data=readBamFile(chipName, dataPath)
input.data=readBamsFile(inputName,dataPath)
```

2.2 Calculate QC-metrics from CrossCorrelation analysis

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

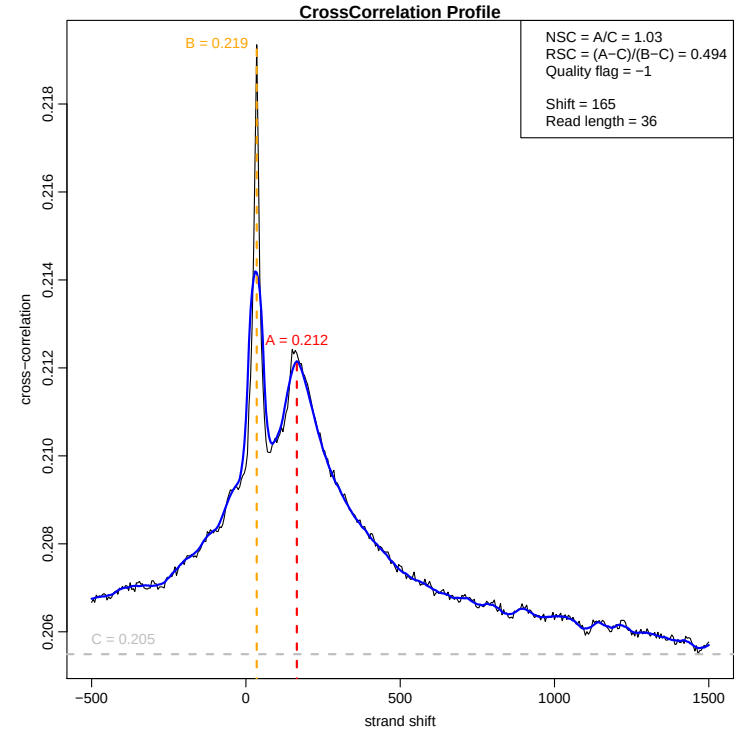


Figure 1: Cross-correlation plot for sample "ENCFF000BLL"

```
#calculate cross correlation QC-metrics for ChIP data
crossvalues_Chip<-calculateCrossCorrelation(chip.data,
    chip_binding.characteristics,
    read_length=36,
    savePlotPath=getwd(),
    plotname="ChIP")
```

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

calculateCrossCorrelation has to be used on both, ChIP and Input. 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).

```
> str(CC_Result_Chip)

List of 20
 $ CC_StrandShift                : num 165
 $ tag.shift                     : num 82
 $ N1                            : num 17767863
 $ Nd                            : num 18898775
 $ CC_PBC                        : num 0.94
 $ CC_readLength                 : num 36
```

```
 $ CC_UNIQUE_TAGS_LibSizeadjusted: num 9654631
 $ CC_NSC                        : num 1.03
 $ CC_RSC                        : num 0.494
 $ CC_QualityFlag                : num -1
 $ CC_shift.                     : num 165
 $ CC_A.                         : num 0.212
 $ CC_B.                         : num 0.219
 $ CC_C.                         : num 0.205
 $ CC_ALL_TAGS                   : int 20191301
 $ CC_UNIQUE_TAGS                : int 18898775
 $ CC_UNIQUE_TAGS_nostrand       : int 18826294
 $ CC_NRF                        : num 0.936
 $ CC_NRF_nostrand               : num 0.932
 $ CC_NRF_LibSizeadjusted        : num 0.965
```

```
> final.tag.shift<-CC_Result_Chip$tag.shift
```

2.3 Calculate QC-metrics from peak calls

The last set of QC-metrics come from the called peaks using *getBindingRegionsScores*. The *removeLocalTagAnomalies* function performs two steps: it selects only tags with acceptable alignment quality (disabled in the current version) and removes local tag anomalies like regions with extremely high tag counts compared to the neighborhood (for more details see [1]).

```
selectedTags=removeLovalTagAnomalies(chip.data,
    input.data,
    chip_binding.characteristics,
    input_binding.characteristics)

chip.dataSelected=selectedTags$chip.dataSelected

#get QC-values from peak calling
bindingScores=getBindingRegionsScores(chip.data,
    input.data,
    chip.dataSelected,
    input.dataSelected,
    final.tag.shift)
```

The function *getBindingRegionsScores* returns a number of QC-metrics [**?**].

```
> str(CC_Result_Chip)
```

```
List of 6
 $ CC_FDRpeaks          : num 0
 $ CC_evalpeaks         : num 2
 $ CC_FRiP_broadPeak    : num 0.066
 $ CC_FRiP_sharpPeak    : num 0
 $ CC_outcountsBroadPeak: int 1340797
 $ CC_outcountsSharpPeak: int 354
```

2.4 Profile smoothing

```
smoothed.densityChip=tagDensity(chip.dataSelected,
    final.tag.shift,
    srngl=rngl)
```

A necessary step is the smoothing (using a Gaussian kernel) of the tag profile to obtain the "tag density profile" (for more details see [1]). The tag density profile is needed to calculate the next categories of QC-metrics.

3 QC-metrics on global read distribution and "fingerprint" plot

This set of values is based on the global read distribution along the genome for ChIP and Input data [2]. The function *QCscores_global* reproduces the so-called "fingerprint" plot (Figure 2) and returns a list of 9 QC-metrics that can be sampled 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=QCscores_global(densityChip=smoothedDensityChip,
    densityInput=smoothedDensityInput,
    savePlotPath=getwd())
```

```
> str(Ch_Results)
```

```
List of 9
 $ Ch_X.axis                       : num 0.661
 $ Ch_Y.Input                      : num 0.091
 $ Ch_Y.Chip                       : num 0.127
 $ Ch_sign_chipVSinput             : num 1
 $ Ch_FractionReadsTopbins_chip    : num 0.097
 $ Ch_FractionReadsTopbins_input   : num 0.104
 $ Ch_Fractions_without_reads_chip : num 0.362
 $ Ch_Fractions_without_reads_input: num 0.426
 $ Ch_DistanceInputChip            : num 0.036
```

```
> f_fingerPrintPlot(cumSumChip,cumSumInput)
```

4 Metagene profiles and QC-metrics based on local enrichment

Metagene plots show the signal enrichment around a region of interest like the transcription start site (TSS) or over a predefined set of genes. The tag density of the immunoprecipitation is taken over all RefSeg annotated human genes,

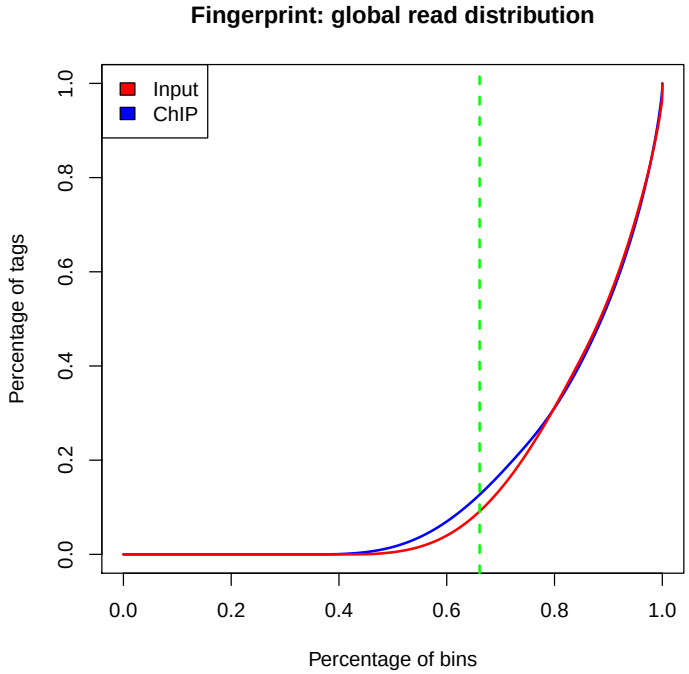


Figure 2: Fingerprint plot of sample ENCFF000BLL and its input ENCFF000BKA.

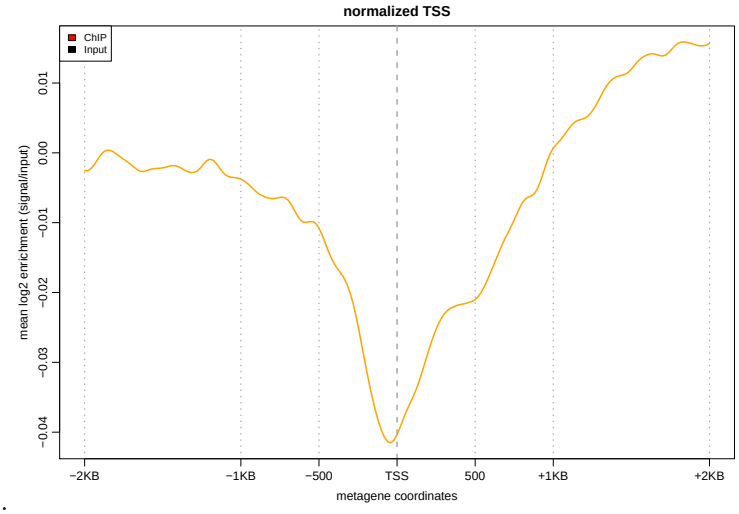


Figure 3: Normalized non-scaled metagene profile for the signal enrichment around the TSS

averaged and log2 transformed. The same is done for the input. The normalized profile (Figure 3) 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. 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 4.

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

```
Meta_Result=createMetageneProfile(smoothedDensityChip,
    smoothedDensityInput,
    tag.shift,
    annotationID="hg19")
```

The objects can be used to create the final metagene plots and to get the respective QC-values for the non-scaled profiles (Figure 3).

```
> #create non-scaled metagene profile around the TSS
> TSS_Plot=nonScaledMetageneProfile(Meta_Result$TSS$chip,
+     Meta_Result$TSS$input,
+     tag="TSS")
> #create non-scaled metagene profile around the TES
> TES_Plot=nonScaledMetageneProfile(Meta_Result$TES$chip,
+     Meta_Result$TES$input,
+     tag="TES")
```

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

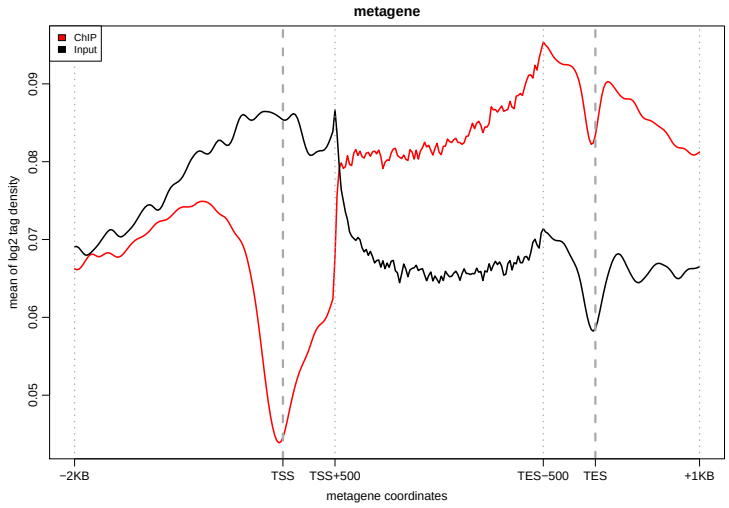


Figure 4: Scaled metagene profile for the tag density across annotated genes for ChIP (ENCFF000BLL) and Input (ENCFF000BKA)

```
> #create scaled metagene profile
> geneBody_Plot=scaledMetageneProfile(Meta_Result$twopoint$chip,
+     Meta_Result$twopoint$input)
```

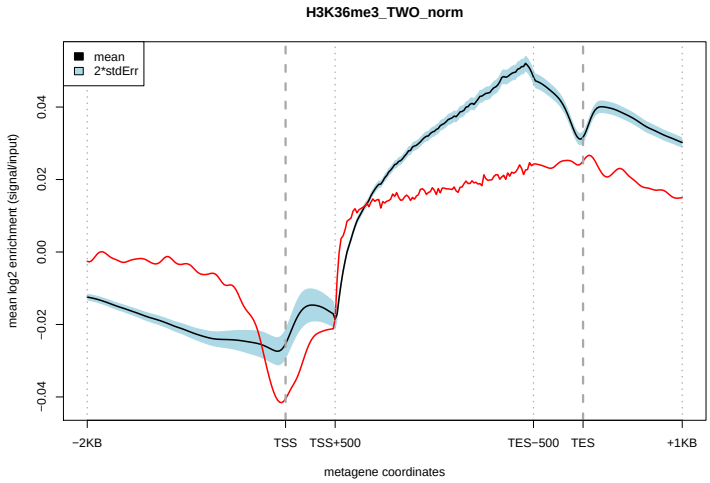
5 Quality assessment using the compendium of QC-metrics as reference

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 two functions for that: *metagenePlotsForComparison* to compare the metagene plots with the compendium and *plotReferenceDistribution* to compare a QC-metric with the compendium values.

5.1 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.

```
> metagenePlotsForComparison(chrommark="H3K36me3",
+     Meta_Result$twopoint,
+     Meta_Result$TSS,
+     Meta_Result$TES)
```



.

Figure 5: The comparison against the compendium (mean and standard error) can highlight potential problems in the signal distribution for a specific mark, in case of experiment failure.

5.2 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 **??**).

```
> currentRSCforSample=CC_Result_Chip$QCscores_ChIP$CC_RSC
> plotReferenceDistribution(chrommark="H3K36me3",
+     metricToBePlotted="RSC",
+     currentValue=currentRSCforSample)
```

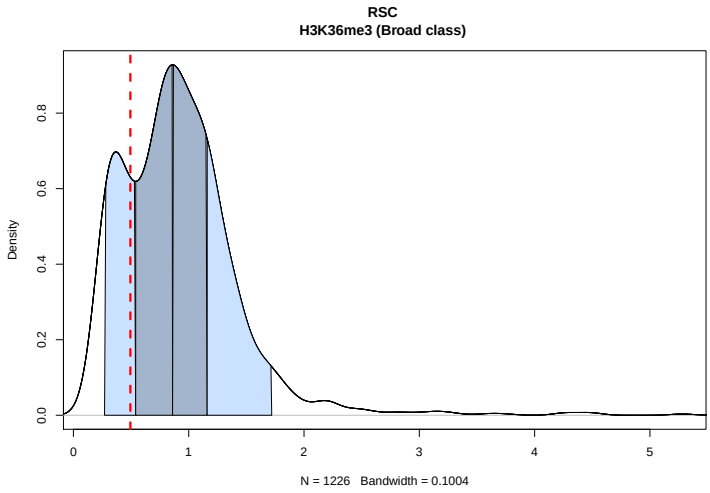


Figure 6: Density plot showing the QC-metric RSC (red dashed line) of ENCFF000BLLs against the reference distribution of the compendium values stratified by chromatin classes.

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.