Predicting the Chromatin Conformation From ChIPseq CebpE: part 1 (Component Analysis)

Ricky Lim[1], Samuel Collombet[2], Agus Salim[3], Touati Benoukraf[1]

[1]Cancer Science Institute of Singapore, National University of Singapore

[2]Institut de Biologie de l'Ecole Normale Superieur de Paris

[3]Department of Mathematics and Statistics

benoukraf@nus.edu.sg

```
source('/home/ricky/Rlim/ChromatinConformation/ComponentCalls/ComponentCalls.R')
work_dir = '/home/ricky/Rlim/ChromatinConformation/ComponentCalls/CebpE/'
```

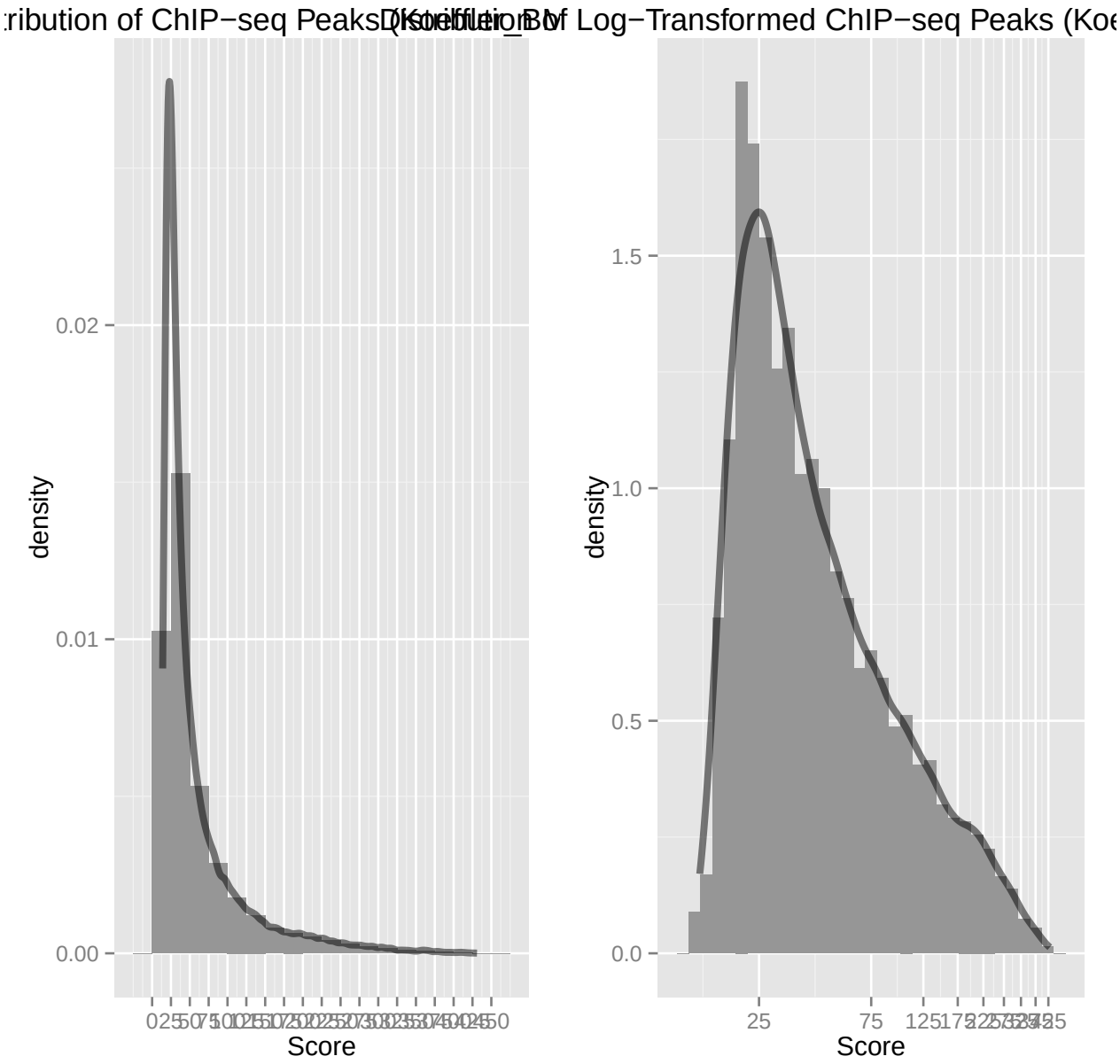
1 Load Dataset

Proof-of-concept: We used ChIP-seq peak signals of CebpE in bone marrow cells from Koeffler's Lab. Here is the quick insight on Cebp Protein from Pubmedylicious.

```
Koeffler_BM_CebpE <- read.table(paste0(work_dir,
                        'Input/KoefflerLab_BM_ChIPseq_CebpE_mm10_q10rmdup_peaks.xls'),
                        header=T, sep='\t', row.names=10, skip=28)

p1 <- ggplot(Koeffler_BM_CebpE, aes(x=pileup)) +
        geom_histogram(aes(y = ..density..), fill='#969696', binwidth=25) +
        geom_line(stat='density', alpha=0.5, size=1.5) +
        scale_x_continuous(breaks=seq(0,450, 25)) +
        ggtitle('Distribution of ChIP-seq Peaks (Koeffler_BM_CebpE)')+
        xlab('Score')
ggsave('figs/Koeffler_BM_CebpE_CountChipSeqPeaks.pdf', p1)

## Saving 7 x 7 in image
```




```
#summary(Koeffler_BM_CebpE$pileup)
#summary(log10(Koeffler_BM_CebpE$pileup))
p2 <- ggplot(Koeffler_BM_CebpE, aes(x=pileup)) +
        geom_histogram(aes(y = ..density..), fill='#969696', binwidth=.05) +
        geom_line(stat='density', alpha=0.5, size=1.5) +
        scale_x_log10(breaks=seq(25,450, 50)) +
        ggtitle('Distribution of Log-Transformed ChIP-seq Peaks (Koeffler_BM_CebpE)') +
        xlab('Score')
ggsave('figs/Koeffler_BM_CebpE_CountChipSeqPeaks_log.pdf', p2)
```

```
## Saving 20 x 12.2 in image
```

```
#grid.arrange(p1, p2, ncol=2)
#p <- arrangeGrob(p1,p2, ncol=2)
#ggsave('figs/Koeffler_BM_CebpE_CountChipSeqPeaks.pdf', p)
```

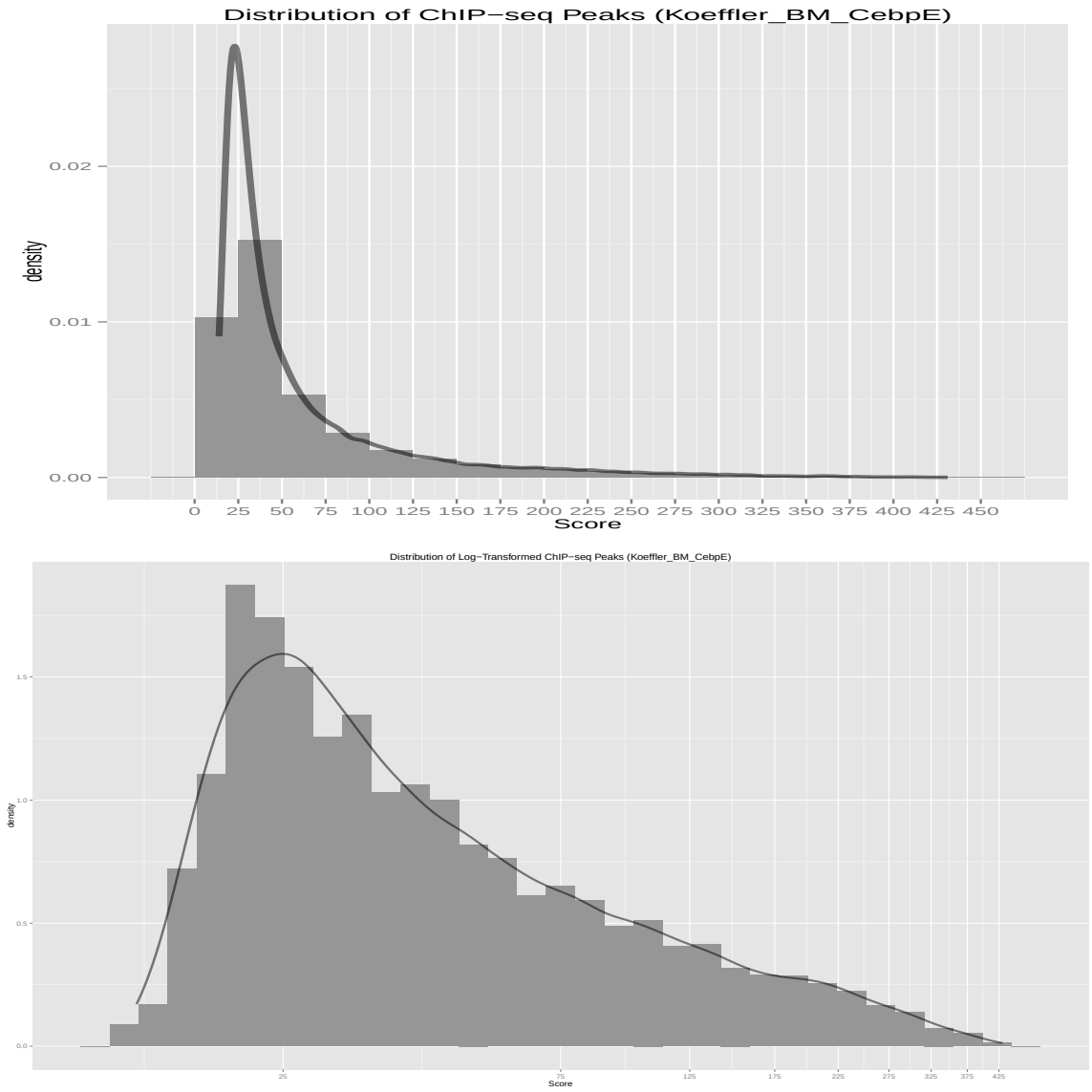


Figure 1: Distibution of ChIPseq Peak Pileup Scores.

2 Models Fitted on ChIP-seq Peaks

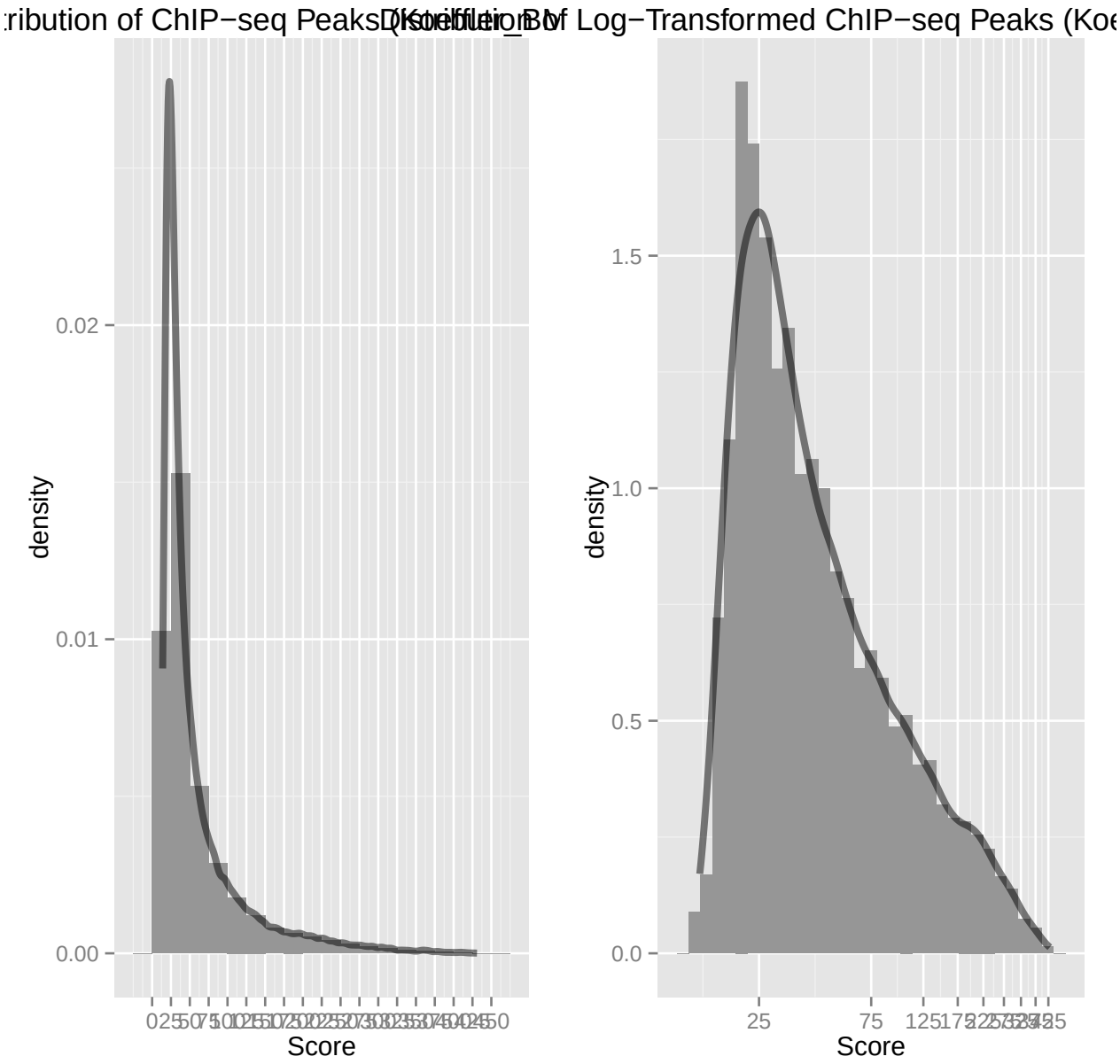
Here, we attempt to fit ChIP-seq peaks with Normal, and Negative Binomial distributions. Figure 2 demonstrates that Negative Binomial distribution provides better fit with higher loglik (-4978.32), compared to normal distribution with lower loglik of -5429.51.

```
library(MASS)

set.seed(123)
chip_data <- sort(sample(Koeffler_BM_CebpE$pileup, 1000))
n <- length(chip_data)

# fitting with models
normal.par <- fitdistr(chip_data, 'normal')
poisson.par <- fitdistr(chip_data, 'Poisson')
negbi.par <- fitdistr(chip_data, 'negative binomial')

# counts estimates generated by models
normal.estimate <- rnorm(n, normal.par$estimate)
poisson.estimate <- rpois(n, poisson.par$estimate)
negbi.estimate <- rnbinom(n, size=negbi.par$estimate['size'], mu=negbi.par$estimate['mu'])
```




```
## X11cairo
##        2
## X11cairo
##        2
## X11cairo
##        2
```

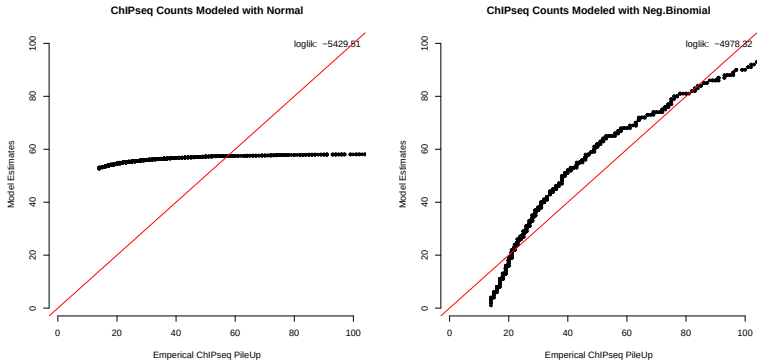


Figure 2: Models Fitting: normal, and neg.binomial. The perfect fit is drawn on the red line, in which the model estimates match perfectly to the empirical observation. The more points lying on this line, the better the fit.

3 Mixture Models Fitted on ChIP-seq Peaks

Figure 2 shows that single model is not sufficient to fit the ChIPseq data. For this reason, here we fit the mixture models of Normal (Gaussian Mixture Models or GMM) and Negative Binomial distributions (Negative Binomial Models or NBMs).

3.1 GMMs Fitted on ChIP-seq Peaks

```
# fit with several k-component-GMMs
GMMs_list_Koeffler_BM_CebpE <- fitMMs(X=Koeffler_BM_CebpE$pileup, max_k=10, model = 'GMM')
```

3.2 NBMs fitted on ChIPseq Peaks

```
NBMs_list_Koeffler_BM_CebpE <- fitMMs(X=Koeffler_BM_CebpE$pileup, max_k=10, 'NBM')
```

4 Visualization of the Fitted MMs on ChIP-seq distributions

4.1 Fitted GMMs Visualized on ChiP-seq distributions

```
visualizeFitMMs(X=Koeffler_BM_CebpE, MMs_list=GMMs_list_Koeffler_BM_CebpE,
                max_k=10, output_n='figs/Koeffler_BM_CebpE_GMM_ModelVisualization',
                model = 'GMM', titleName='Koeffler_BM_CebpE')
```

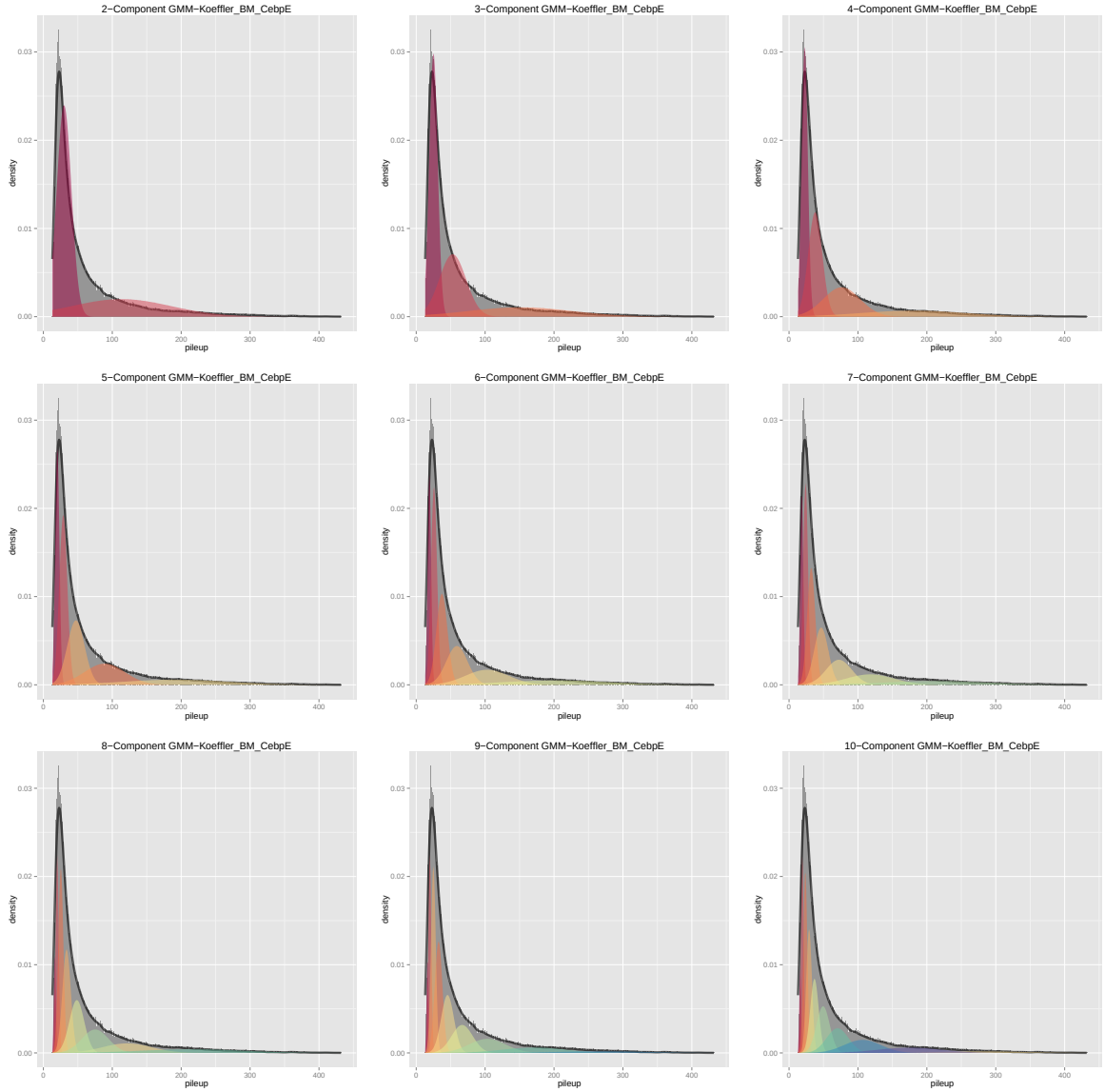


Figure 3: ChIPseq Peaks Fit with GMMs Ranging from Two to Ten Components

4.2 Fitted NBMs Visualized on ChiP-seq distributions

```
visualizeFitMMs(X=Koeffler_BM_CebpE, MMs_list=NBMs_list_Koeffler_BM_CebpE,
                max_k=10, output_n='figs/Koeffler_BM_CebpE_NBM_ModelVisualization',
                model = 'NBM', titleName='Koeffler_BM_CebpE')
```

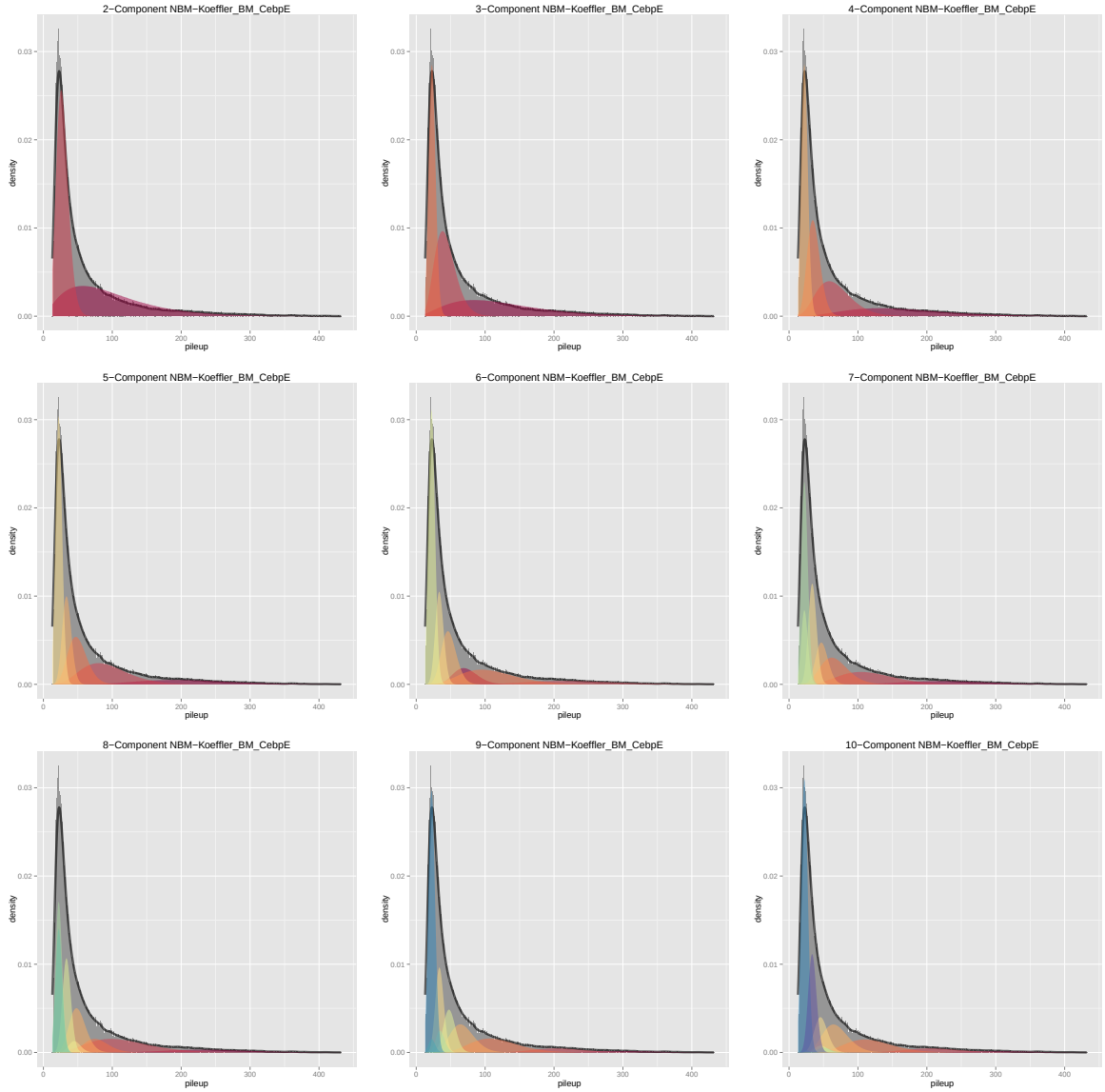


Figure 4: ChIPseq Peaks Fit with NBMs Ranging from Two to Ten Components

5 Emperical vs Estimate Distribution

Here, we assess for the possible overfit of our fitting mixture models by comparing the emperical and overal estimate distribution from the mixture models.

5.1 Comparing Density Estimates: Empirical vs Gaussian Mixture (GMM)

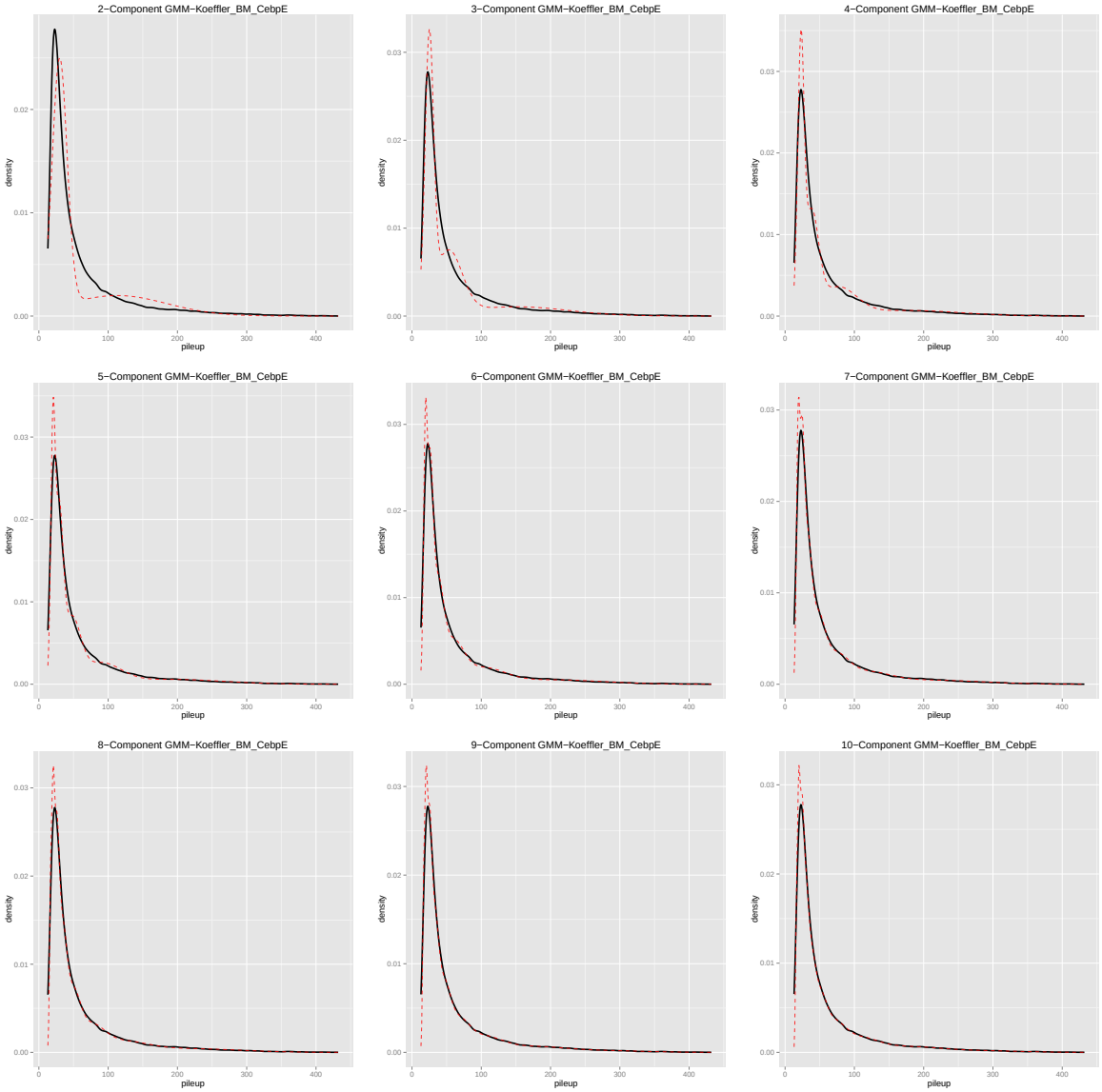


Figure 5: Comparing Density Estimates: Empirical Vs Gaussian Mixture. Empirical and gaussian density are coloured with black and dashed-red, respectively.

5.2 Comparing Density Estimates: Empirical vs Gaussian Mixture (NBM)

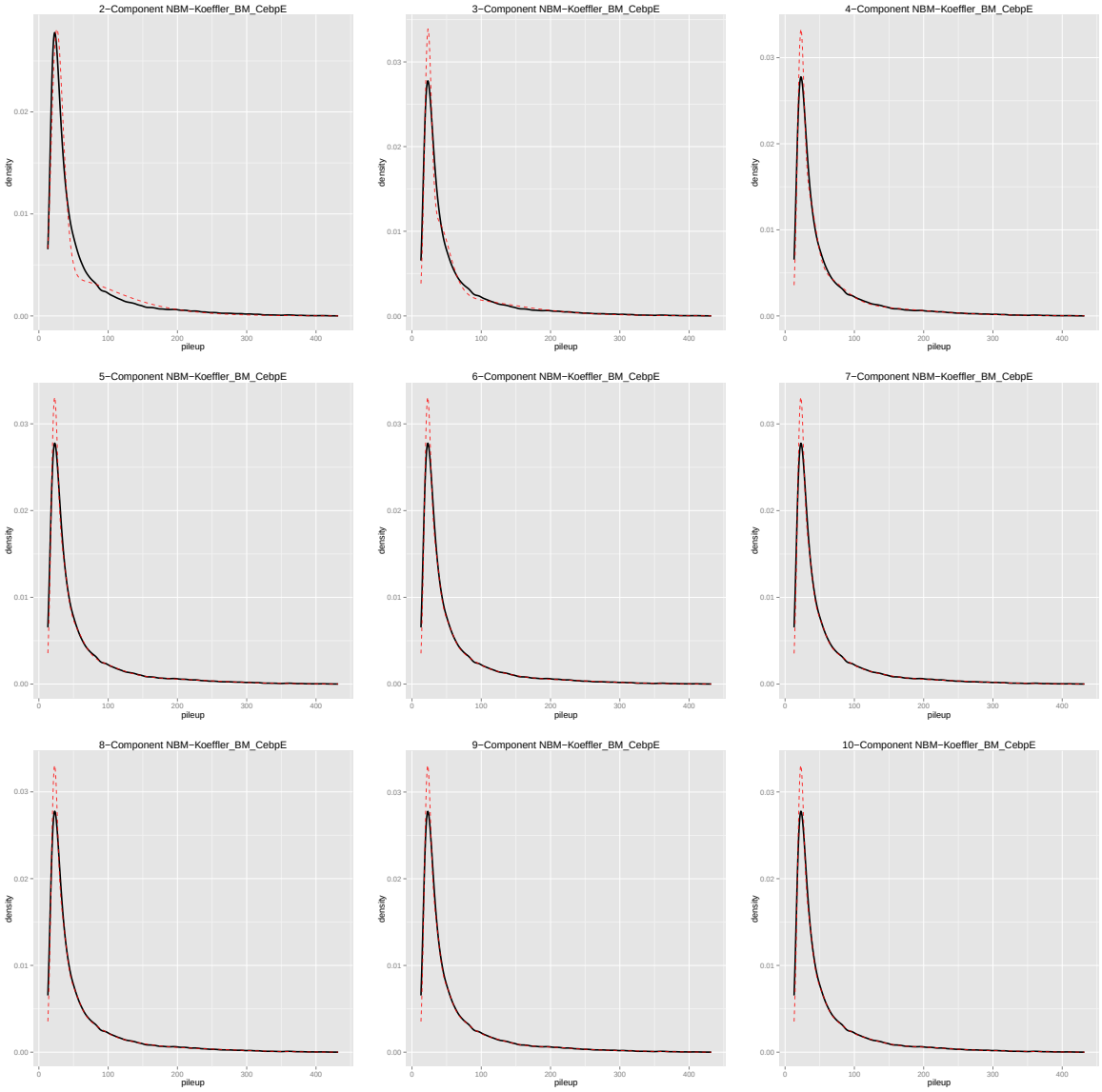


Figure 6: Comparing Density Estimates: Empirical Vs Gaussian Mixture. Empirical and gaussian density are coloured with black and dashed-red, respectively.

6 Stability of the Distribution of Component Assignments

Here, we checked the distribution of component assignments from 3 different runs (simulation) in order to assess the stability of the fitting models.

Stable component assignment assumes that the number for each component in the course of three different runs stays constant.

6.1 Stability of the Distribution of Component Assignments: GMMs

	Rep.1	Rep.2	Rep.3
Comp.1	16417	16417	16417
Comp.2	35060	35060	35060

	Rep.1	Rep.2	Rep.3
Comp.1	8866	8866	8866
Comp.2	16614	16614	16614
Comp.3	25997	25997	25997

	Rep.1	Rep.2	Rep.3
Comp.1	5601	5601	5601
Comp.2	10494	10494	10494
Comp.3	15643	15643	15643
Comp.4	19739	19739	19739

	Rep.1	Rep.2	Rep.3
Comp.1	4676	4676	4676
Comp.2	8306	8306	8306
Comp.3	11142	11142	11142
Comp.4	11708	11708	11708
Comp.5	15645	15645	15645

	Rep.1	Rep.2	Rep.3
Comp.1	3977	3977	3977
Comp.2	6386	6386	6386
Comp.3	8254	8254	8254
Comp.4	8638	8638	8638
Comp.5	11002	11002	11002
Comp.6	13220	13220	13220

	Rep.1	Rep.2	Rep.3
Comp.1	3350	3350	3350
Comp.2	4755	4755	4755
Comp.3	6561	6561	6561
Comp.4	6964	6964	6964
Comp.5	8206	8206	8206
Comp.6	9981	9981	9981
Comp.7	11660	11660	11660

	Rep.1	Rep.2	Rep.3
Comp.1	2520	2520	2520
Comp.2	3217	3217	3217
Comp.3	4561	4471	4471
Comp.4	6354	6444	6444
Comp.5	7666	7666	7666
Comp.6	8180	8180	8180
Comp.7	9426	9426	9426
Comp.8	9553	9553	9553

	Rep.1	Rep.2	Rep.3
Comp.1	1283	1283	1283
Comp.2	2520	2520	2520
Comp.3	3286	3286	3286
Comp.4	5041	5041	5041
Comp.5	6118	6118	6118
Comp.6	6187	6187	6187
Comp.7	7712	7712	7712
Comp.8	9344	9344	9344
Comp.9	9986	9986	9986

	Rep.1	Rep.2	Rep.3
Comp.1	1164	1164	1177
Comp.2	1421	1421	1421
Comp.3	3249	3249	3236
Comp.4	4808	4808	4808
Comp.5	5445	5445	5445
Comp.6	6549	6549	6549
Comp.7	6611	6611	6611
Comp.8	7217	7217	7217
Comp.9	7408	7408	7408
Comp.10	7605	7605	7605

Figure 7: Component Frequency over Three Different Runs: GMM

Stability of the Distribution of Component Assignments: NBMs

```
## pdf pdf pdf pdf
##   2   2   2   2
```

Note: Fitting with mixture negative binomial from seven components onwards does not assign the whole components for each ChiPseq peaks. For example, with 7-component NBMs, the components assigned are 1,2,3,4,5,6 but not 7.

```
# to adjust the margin of the pdf table
#for f in $(ls Koeffler_BM_CebpE_CompFreqTable_*); do pdfcrop --margins '5 10 5 20' $f $f;
done
```

	Rep.1	Rep.2	Rep.3
Comp.1	19853	19853	19853
Comp.2	31624	31624	31624

	Rep.1	Rep.2	Rep.3
Comp.1	12353	12353	12353
Comp.2	18292	18292	18292
Comp.3	20832	20832	20832

	Rep.1	Rep.2	Rep.3
Comp.1	6977	6977	6977
Comp.2	11640	11640	11640
Comp.3	13121	13121	13121
Comp.4	19739	19739	19739

	Rep.1	Rep.2	Rep.3
Comp.1	3934	3934	3934
Comp.2	8333	8333	8333
Comp.3	9107	9107	9107
Comp.4	9271	9271	9271
Comp.5	20832	20832	20832

Figure 8: Component Frequency over Three Different Runs: NBM

7 Model Assessment: Information Criterion

We assessed the model on the basis of information criterion consisting of AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), and log-likelihood.

We expect the log-likelihood to be increasing as the number of components of GMMs fitted or the negative log-likelihood to be decreasing with increasing number of components.

AIC and BIC is based on Occam's razor principle, i.e, the simplest the better. As the assessment matrix, AIC and BIC would include the positive contribution of likelihood and also penalize the increasing number of parameters used to fit the model.
The equations of AIC and BIC are as follows:

$\text{AIC} = -2 \times \log L + 2 * P$
$\text{BIC} = -2 \times \log L + \log(n) * P$
L is likelihood
P is the number of parameters

7.1 GMMs Model Assessment: AIC and BIC

```
getModelAssessment(GMMs_list_Koeffler_BM_CebpE, max_k = 10,
                   output_n='figs/Koeffler_BM_CebpE_GMM_ModelAssessment',
                   model='GMM', titleName='Koeffler_BM_CebpE')
```

7.2 NBMs Model Assessment: AIC and BIC

```
getModelAssessment(NBMs_list_Koeffler_BM_CebpE, max_k = 10,
                   output_n='figs/Koeffler_BM_CebpE_NBM_ModelAssessment',
                   model='NBM', titleName='Koeffler_BM_CebpE')

## pdf
##   2
```

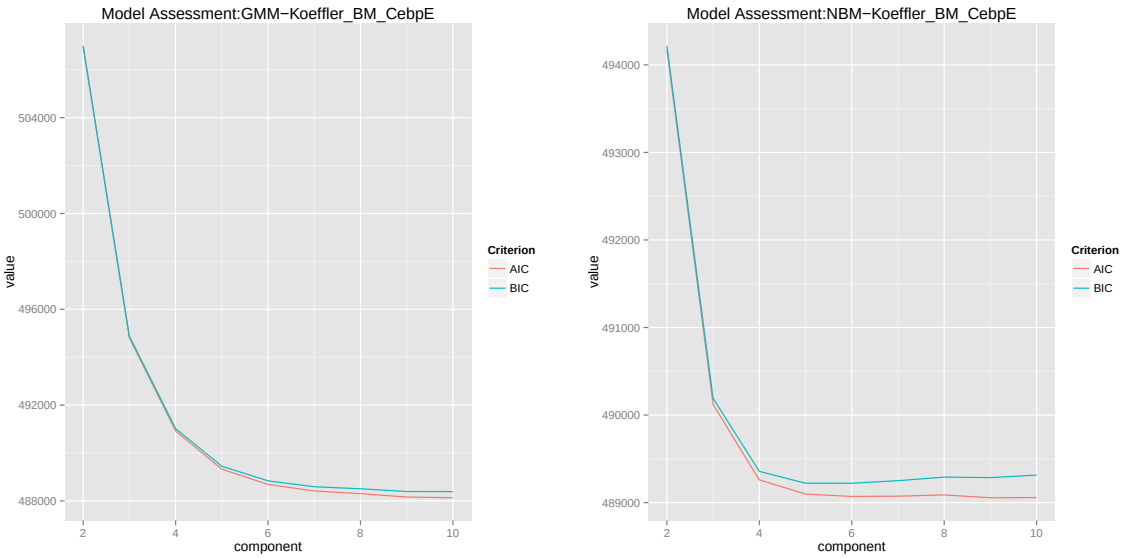


Figure 9: Gaussian (left) and Neg.Binomial (right) Mixture Models Assessment: AIC, and BIC. Note that BIC introduces the larger penalty term ($\log(n)$) compared with AIC (2). In this assessment, positive penalty term and negative likelihood are applied. Hence, the lower the score the better the fitting of the model while taking into account its complexity.

8 Component Assignment by Mixture Models

8.1 Component Assignment to ChIP-seq Peaks

```
assignComponentMMs(Koeffler_BM_CebpE, GMMs_list_Koeffler_BM_CebpE,
                   output_n=paste0(work_dir,
                    'Output/Koeffler_BM_CebpE_GMM_ModelAssignment'),
                    model='GMM', sort.comp=TRUE)
```

```
assignComponentMMs(Koeffler_BM_CebpE, NBMs_list_Koeffler_BM_CebpE,
                   output_n=paste0(work_dir,
                    'Output/Koeffler_BM_CebpE_NBM_ModelAssignment'),
                   model='NBM', sort.comp=TRUE)
```

9 Visualization Component Assignment

9.1 Visualization Component Assignment

Here, the sorted-5-component GMM and NBM (Figure 10) are visualized on chr14.

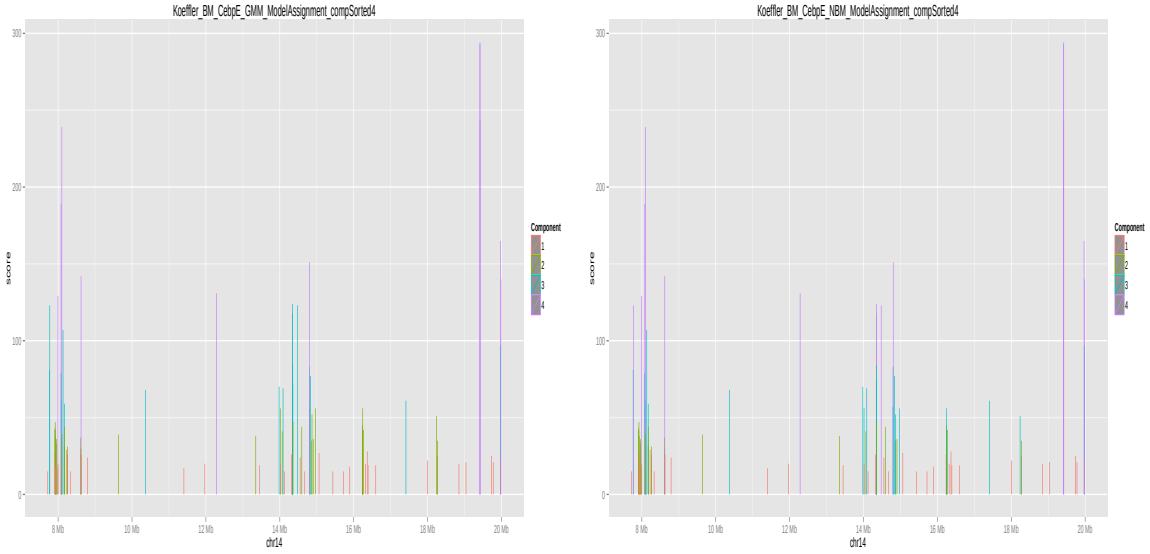


Figure 10: Sorted-4-Component GMM (left) and NBM (right) assignment on Chr14.

10 Score Distribution of Component Assignments

The score distribution of 5-component GMM and NBM are shown as boxplots (Figure **??**)

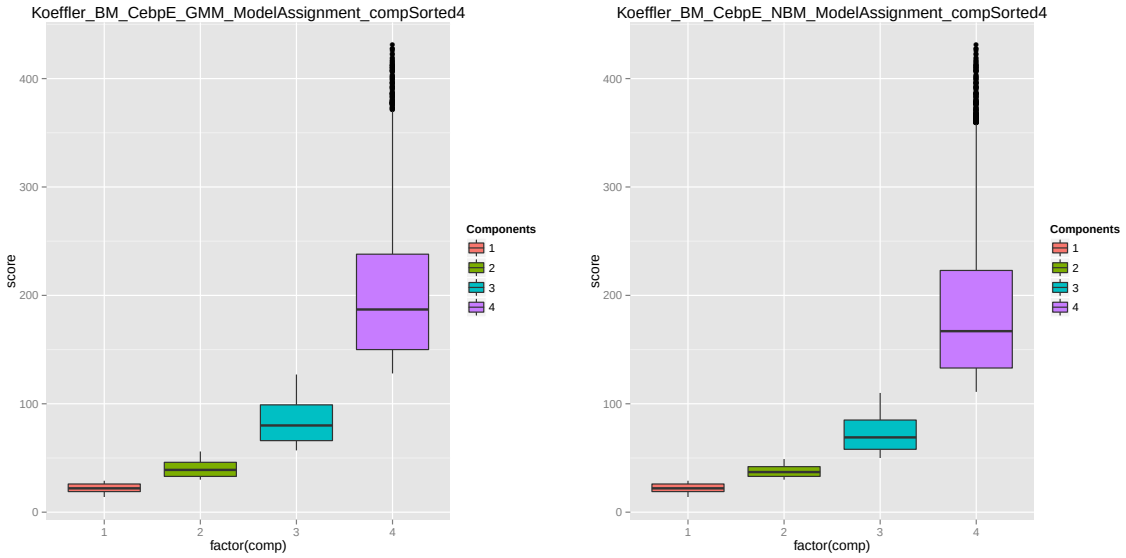


Figure 11: Boxplots of ChIP Peak Scores clustered in 4-component GMM (left) and NBM (right)

```
  Filename: chromatinConformation.Rnw
  Working directory: /home/ricky/Rlim/ChromatinConformation/ComponentCalls/CebpE/ComponentAnalysis
```

11 Metainfo

```
sessionInfo()

## R version 3.0.2 (2013-09-25)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_SG.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_SG.UTF-8        LC_COLLATE=en_SG.UTF-8
##  [5] LC_MONETARY=en_SG.UTF-8    LC_MESSAGES=en_SG.UTF-8
##  [7] LC_PAPER=en_SG.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_SG.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] plyr_1.8.1           gridExtra_0.9.1      broom_0.2
##  [4] wordcloud_2.5        BSgenome_1.30.0      Biostrings_2.30.1
##  [7] rtracklayer_1.22.7   GenomicRanges_1.14.4 XVector_0.2.0
## [10] IRanges_1.20.7       BiocGenerics_0.8.0   ggbio_1.10.16
## [13] RColorBrewer_1.0-5   scales_0.2.4         xtable_1.7-3
## [16] reshape2_1.4         ggplot2_1.0.0        mixtools_1.0.2
## [19] segmented_0.4-0.0    MASS_7.3-34          boot_1.3-11
## [22] knitr_1.6
##
## loaded via a namespace (and not attached):
##  [1] AnnotationDbi_1.24.0     Biobase_2.22.0
##  [3] biomaRt_2.18.0           biovizBase_1.10.8
##  [5] bitops_1.0-6             cluster_1.15.2
##  [7] codetools_0.2-9          colorspace_1.2-4
##  [9] DBI_0.2-7                dichromat_2.0-0
## [11] digest_0.6.4             evaluate_0.5.5
## [13] formatR_1.0              Formula_1.1-2
## [15] GenomicFeatures_1.14.5   gtable_0.1.2
## [17] highr_0.3                Hmisc_3.14-4
## [19] labeling_0.3             lattice_0.20-29
## [21] latticeExtra_0.6-26      munsell_0.4.2
## [23] proto_0.3-10             Rcpp_0.11.2
## [25] RCurl_1.95-4.3           Rsamtools_1.14.3
## [27] RSQLite_0.11.4           slam_0.1-32
## [29] splines_3.0.2            stats4_3.0.2
## [31] stringr_0.6.2            survival_2.37-7
## [33] tcltk_3.0.2              tools_3.0.2
## [35] VariantAnnotation_1.8.13 XML_3.98-1.1
## [37] zlibbioc_1.8.0
```

```
library(knitr)
purl("componentAnalysis.Rnw" ) # compile to tex

## [1] "componentAnalysis.R"

purl("componentAnalysis.Rnw", documentation = 0) # extract R code only

## [1] "componentAnalysis.R"

knit2pdf("componentAnalysis.Rnw")

## Error:  duplicate label 'setup'
```