ChIP-partitioning tool: shape based analysis of transcription factor binding tags Romain Groux, Sunil Kumar and Philipp Bucher Last update: 01.04.2016

Introduction

Probabilistic partitioning methods to discover significant patterns in ChIP-Seq data will be discussed in this section with multiple simulated and biological examples [Nair et al., 2014]. Our methods take into account signal magnitude, shape, strand orientation and shifts. We have compared this method with some of the existing methods and demonstrated significant improvements, especially with sparse data [please refer the publication for details on comparative analysis]. Besides pattern discovery and classification, probabilistic partitioning can serve other purposes in ChIP-Seq data analysis. In this section we will also exemplify its merits in the context of peak finding and partitioning of ChIP-seq tag distribution for transcription factor binding (ex: CTCF).

Hints and recipes

The exercise is divided into four parts. First part is using simulated data and other three parts on real experimental data. Before creating the input data, lets define ChIP partitioning functions.

Definition of function

Various variants of Expectation-Maximization partitioning methods can be downloaded form here: EM functions. Download and save the file as em_functions.R.

ChIP partitioning using different datasets

Part I: Simulated data [*To be performed in R*]

Model definition and data generation: R script

```
Composite peak data
K=2; L=63; r=list(1:21,22:42,43:63)
C=matrix(nrow=K, ncol=sum(L))
C[1,r[[1]]] = 2*dnorm(1:21, 8,1.0)
C[1,r[[2]]] = 5*dnorm(1:21, 8,3.0)
C[1,r[[3]]] = 1*dnorm(1:21,11,0.5)
C[2,r[[1]]] = 4*dnorm(1:21, 8,3.0)
C[2,r[[2]]] = 3*dnorm(1:21,14,1.0)
C[2,r[[3]]] = 1*dnorm(1:21, 8,0.5)
Q = c(0.6,0.4)
```

Definition of random samples

```
N=10000
samples=matrix(data=0, nrow=N, ncol=K)
rs=sample(1:K,N,rep=T,prob=Q)
for(i in 1:N) {samples[i,rs[i]]=1}
```

Random data generation

```
data=matrix(nrow=N, ncol=L)
for (i in 1:N) {data[i,] = rpois(L,C[which(samples[i,] == 1),])}
# Let's have a look at the aggregation plot of the complete set
plot(colMeans(data), type = "1")
```

Various partitioning methods: R script

```
Load the em_functions.R file:
source("em_functions.R")
```

Partitioning with random seeds

```
c=matrix(nrow=L, ncol=K)
p=matrix(nrow=N, ncol=K)
for(i in 1:K) {p[,i] = rbeta(N,N**-0.5,1)}
c = (t(p) %*% data)/colSums(p)
q=rep(1/K,K)
for(i in 1:20) {em_shape_comp(c,r,q,data);
plot classes comp(c,r); print(q)}
```

Iterative partitioning

```
c = matrix(data=colMeans(data), nrow=1, ncol=L)
flat =matrix(data= mean(data), nrow=1, ncol=L)
q = 1
for (m in 1:(K-1)) {
c = rbind(flat,c)
q = c(1/m,q); q = q/sum(q)
for(i in 1:20) {em_shape_comp(c,r,q,data);
plot_classes_comp(c,r); print(q)}
}
```

Iterative partitioning with untrained flat class

```
c = matrix(data=colMeans(data), nrow=1, ncol=L)
flat =matrix(data= mean(data), nrow=1, ncol=L)
q = 1
for (m in 1:(K)) {
c = rbind(flat,c)
q = c(1/m,q); q = q/sum(q)
for(i in 1:20) {
c[1,]=flat; em_shape_comp(c,r,q,data); plot_classes_comp(c,r);
print(q)}
}
```

Model evaluation: R script

cor(p, samples) # Pearson's correlation between the real and predicted class probabilities **Part II: CTCF dataset** [GEO series GSE29611 Histone Modifications by ChIP-seq from ENCODE/Broad Institute]

In the current exercise we will be trying to reproduce one of the figures from following paper:

Nair et al., 2014, Probabilistic partitioning methods to find significant patterns in ChIP-Seq data, Bioinformatics, 30, 2406-2013, PMID 24812341.



Shape-based peak evaluation with shifting. The figure illustrates the effects of probabilistic partitioning on a CTCF peak list provided by ENCODE in terms of motif enrichment. (a) Probabilistic partitioning with shifting. (b) Partitioning based on original *P*-values.

In order to reproduce the above figure, we will need an ENCODE peak list and associated tags data.

- ENCODE peak list for CTCF from HUVEC cell line. Download the peak list.
- Process tags data from HUVEC cell line. We will use it directly from the server.

Pre-processing of data to generate tag count matrix.

We will use ChIP-Extract Analysis Module to generate a tag count matrix in defined bins around CTCF sites. Select appropriate target feature and then click submit.

- Reference feature: above peak list
- Target feature: hg19; ENCODE ChIP-seq; GSE29611; HUVEC CTCF (do appropriate centering)
- Analysis parameters: -500 to 500; Window width:10; Count cut-off:1

Download the Ref SGA File and Table (TEXT) and save as huvec_peaks.sga and huvec_ctcf.txt respectively.

Perform iterative shape based partitioning with shifting and prepare data for motif enrichment analysis. R script

```
Read data:
      data = as.matrix(read.table("huvec ctcf.txt"))
Define classes and shifts:
      K=1; S=11; N=dim(data)[1]; L=dim(data)[2]-S+1; ITER=10
Iterative shape based partitioning with shifting
      c = colMeans(data[,(floor(S/2)+1):(floor(S/2)+L)])
      flat = matrix(data=mean(data), nrow=1, ncol=L)
      q=q0 = dnorm(1:S, floor(S/2)+1, 1)/sum(dnorm(1:S, floor(S/2)+1, 1))
      for (m in 1:(K)) {
      c = rbind(flat, c)
      q = rbind(q0/m,q); q/sum(q)
                                                             reg shift(q);
      for(i in 1:ITER) {reg_shift(q); c[1,]=flat;
      em shape shift(c,q,data); plot classes(c); print(q)}
      }
      prob = rowSums(p[,2,])
      shift = apply(p,c(1,3),sum)  %*% (1:S)
Read peaks and filter based on probability value into different classes:
      peak <- read.table("huvec_peaks.sga", header=F, sep="\t")</pre>
      good <- peak[which(prob>=0.5),]
      bad <- peak[which(prob<=0.5),]</pre>
      peak[,3] <- round(peak[,3]+10*(shift-6))
      good shifted <- peak[which(prob>=0.5),] # good peaks after
      shifting
Write output as sga files.
      write.table(good, "chipPart good.sga", sep="\t", guote=F,
      row.names=F, col.names=F)
      write.table(bad, "chipPart bad.sga", sep="\t", quote=F,
      row.names=F, col.names=F)
      write.table(good shifted, "chipPart good shifted.sga", sep="\t",
      quote=F, row.names=F, col.names=F)
```

Read the bed file and sort based on p-value generated by ENCODE consortium. For fair comparison, the threshold for partitioning with the original p-values is chosen such as to match the numbers of good and bad peak obtained with probabilistic partitioning.

```
pval <- read.table("ctcf_peaks.bed", header=F, sep="\t")
spval <- pval[order(pval[,8], decreasing=T),]
goodpval <- spval[1:nrow(good),]
badpval <- spval[nrow(good)+1:nrow(pval),]
write.table(goodpval, "pval_good.bed", sep="\t", quote=F,
row.names=F, col.names=F)
write.table(badpval, "pval_bad.bed", sep="\t", quote=F,
row.names=F, col.names=F)</pre>
```

Use ChIP-Convert to convert the output of partitioning into following formats.

• bed to sga for p-value classes

Following piece of experiment needs to be carried out on SSA web-server. Extract CTCF motif enrichment (for all the sga files produced in previous step) using Oprof

- OProf Parameters: -499 to 500; Window size: 50, Window shift: 1; Search mode: bidirectional; Cut-off: p-val: 1.0000e-05
- PWM form library: JASPAR CORE 2014 vertebrates; Motif: CTCF MA0139.1 (length=19)

Save the graph output as: good_part.txt, bad_part.txt, good_shifted_part.txt, high_pvalue.txt, low_pvalue.txt and ctcf_all.txt (for all the sites)

Plot the results (as shown in above figure). Show plotting script

Now move back into R for plotting the results. This will produce the figure as shown above.

```
## read files that were the output/ graph text output from Oprof
and plot results:
a <- read.table("good part.txt", header=T, sep="")</pre>
b <- read.table("bad part.txt", header=T, sep="")</pre>
c <- read.table("good shifted part.txt", header=T, sep="")</pre>
d <- read.table("high_pvalue.txt", header=T, sep="")</pre>
e <- read.table("low_pvalue.txt", header=T, sep="")
all <- read.table("ctcf_all.txt", header=T, sep="")</pre>
par(mar=c(0,2.75,1,1), mfrow = c(1,2), oma=c(4,0,2,0))
plot(c[,1],c[,2],type="1",lwd=2, col="red", cex.axis=1, xlab="",
ylab="", yaxt="n", ylim=c(0,25));
points(c[,1],all[,2],type="1",lwd=2, col="blue", lty=4)
points(c[,1],b[,2],type="1",lwd=2, lty=3)
plot(d[,1],d[,2],type="l",lwd=2, col="red", cex.axis=1, xlab="",
ylab="", ylim=c(0,25)); title(ylab="Frequency (%)", cex.lab=1,
mqp=c(-2.2,2,1))
points(c[,1],all[,2],type="1",lwd=2, col="blue", lty=4)
points(c[,1],e[,2],type="1",lwd=2, lty=3)
legend("topright",c("All(63,904)","Good(51,136)","Bad(12,768)"),c
ol=c("blue","red","black"), lwd=c(1,1,1), cex=1, box.col=NA,
bty="n", lty=c(4,1,3), seg.len=2)
```

Part III: Nucleosome organization around transcription factor binding sites in GM12878 cells. For this exercise, the datasets come from the following sources: ENCODE Uniform, GSE32970, GSE35586, GSE29611

EPDNew

Here are the needed R functions em_api.R. Download this file and store it in the directory you will work in. The code below contains calls to these functions.

In this exercice, we will investigate the nucleosome organization around CTCF and ETS1 binding sites in GM12878 cells. CTCF is a factor which is known to show nicely phased nucleosomes arrays on each side of its binding sites. ETS1 is a transcription factor which is mostly found near transcription start sites (TSS) but which does not behave as CTCF regarding nucleosome organization. We will first extract CTCF and ETS1 peaks from peak lists which have the corresponding factor motif in their neighbourhood using

FindM. Then, we will measure the density of MNase and DNAseI reads around these peaks by generating matrices of tag counts using ChIP-Extract. We will import these data in R and perform a probabilistic partitioning to try to identify several nucleosomes organization classes around transcription factor binding sites and isolate classes of interest. Finally, we will extract one class of interest and go back to ChIP-Cor for additional analyses.

1. First, let's extract CTCF and ETS1 peaks which have CTCF and ETS1 binding motif nearby, respectively. To do this, go the FindM and fill the parameters as follows:

	SSA Input Data		Signal Description	
Select available Data Sets		e PWMs from Library		
Genome:	H. sapiens (Feb 2009/hg19)	Motif Library:	JASPAR CORE 2014 vertebrates	\$
Data Type :	ENCODE ChIP-seq-peak	Motif:	CTCF MA0139.1 (length=19)	÷
Serles :	Uniform TFBS from UCSC	O Custom Weight Matrix		
Sample :	GM12878 CTCF None Broad peaks	Format	- Please select a format -	
Sequence input via server-resident F	PS file(s)			
O Upload Sequence Set in Fasta Forma	at	PWM text		
		Clear Text		
Upload custom Data 🕖				
	Sequence Range			
 Entire sequence range (for uploaded sequence) 	uences only)			
5' border: -59 3' border: 60		Cut-off O abso	blute Opercentage Op-value	
Sea	uence Selection and Search Criteria	value 0.0001		
Search mode: O forward @ bidirectional Sequence Selection mode : best matches		Bg base composition	0.29, 0.21, 0.21, 0.29	
		Consensus seg		
		Mismatches 0		
		Name (max 20 chars)	MA0139.1 CTCF	
		Reference position	10	
			Sequence Output Format	
		Retrieve sequences in : O FASTA 🛛 🙂 FPS		
		Sequence extraction range around matches (FASTA only) :		
		5'border: 99 3'border: 100		
		Example	e	Submit Reset

SSA Input Data	Signal Description			
8 Select available Data Sets	PWMs from Library			
Genome: H. sapiens (Feb 2009/hg19)	Motif Library: JASPAR CORE 2014 vertebrates			
Data Typo : ENCODE ChIP-seq-peak	Motif: Ets1 MA0098.2 (length=15)			
Series : Uniform TFBS from UCSC	O Custom Weight Matrix			
Sample : - GM12878 ETS1 None HudsonAlpha ; 🗘	Format – Please select a format – 🛟			
O Sequence input via server-resident FPS file(s)				
O Upload Sequence Set in Fasta Format	PWM text			
	Clear Text			
O Upload custom Data Ø				
Sequence Range				
Entire sequence range (for uploaded sequences only)				
5' border: -59 3' border: 60	Cut-off O absolute O percentage O p-value			
Sequence Selection and Search Criteria	value 0.0001			
	Bg base composition 0.29, 0.21, 0.21, 0.29			
Search mode: O forward 🐵 bidirectional	O Consensus seq			
Sequence Selection mode : best matches				
	Mismatches			
	Name (max 20 chars) MA0098.2 Ets1			
	Reference position 8			
	Sequence Output Format			
	Retrieve sequences in : O FASTA 🛛 🔒 FPS			
	Sequence extraction range around matches (FASTA only) :			
	5' border: -99 3' border: 100			
	Example Submit Reset			

From the result page you will need to do two things. First save the results on your hard drive under sga format and name the files 'findm_CTCF.sga' and 'findm_ETS1.sga' (in case you can retrieve these files here : findm_CTCF.sga findm_ETS1.sga). Second, redirect the results toward ChIP-Extract by clicking on the corresponding button under 'Useful links for Downstream Analysis'.

 For each FindM result, run ChIP-extracts (4 runs total) as shown under and save ChIP-Extract results on your hard drive by naming your files 'chipextract_CTCF_MNase.mat','chipextract_CTCF_DNaseI.mat', 'chipextract_ETS1_MNase.mat', 'chipextract_ETS1_DNaseI.mat'. ChIP-Extract Analysis Module

Feature Correlation Tool v1.5.2					
Ch	IP-Seq Input Data (Reference Feature)	ChIP-Seq Input Data (Target Feature)			
O Select available Data Sets ()		8 Select available Data Sets 0			
		Genome : H. sapiens (Feb 2009/hg19)			
Filename :	hourly/Findm_23226_15602.sga	Data Type : ENCODE DNase FAIRE etc.			
Experiment :	Custom SGA	Series () : GSE35586, Nucleosome Position by MNas 💲			
Feature 🕖 :		Sample : Nucleosome Gm12878 rep1			
O Upload custom Data 🕖		Server-resident SGA Files by Filename	Server-resident SGA Files by Filename		
	Additional Input Data Options	O Upload custom Data 🕖	O Upload custom Data 🛈		
Strand (): O + O - O any	oriented	Additional Input Data Options			
Centering 🕖 :		Strand (): 0 + 0 - 0 any			
Repeat Masker 🕖		Centering () : 70			
	Analysis Parameters	Repeat Masker ()			
🖲 Range 🛈		Heatmap Options			
Beginning : -999	End : 1000	Heatmap with ordering Ø			
Minday Width (1)	Histogram Paraméters	Example			
		Submit Reset			
Count Cut-off U : 1					
raw Count density globa	al				
ChIP-Extract Analysis Mo	dula				
Chin -Extract Analysis mo	auto				
Feature Correlation Tool v1.5.2					
Chi	P-Seq Input Data (Reference Feature)	ChIP-Seq Input Data (Target Feature)			
O Select available Data Sets ()		Select available Data Sets ()			
		Genome : H. sapiens (Feb 2009/hg19)			
Server-resident SGA Files by Filer	haudu/Godm 8400 20260 and	Data Type : ENCODE DNase FAIRE etc.			
Filename .	Curber 5C4	Series 🕖 : Boyle 2008, Open Chromatin by DNasel H 🤤			
Experiment .	Custom sua	Sample : DNasel HS - GM12878 - Rep1			
reature 👽 :					
O Upload custom Data 🕖		O Server-resident SGA Files by Filename			
	Additional Input Data Options	Upload custom Data 🕖			
Strand (): O + O - O any	e oriented	Additional Input Data Options			
Centering 🕖 :		Strand 10 : O + O - 😕 any			
Repeat Masker 10		Centering ():			
	Analysis Parameters	Bepeat Masker ()			
Range ①		Heatmap Options			
Beginning : -999	End : 1000	Heatmap with ordering Ø			
	Histogram Parameters	Example			
Window Width 🛈 : 10			in the little based		
Count Cut-off 0 : 1		Submic, Reset	i soomit keset		
Normalization () :					
- raw Count density O glob	uai				

3. Now open a R session in the folder where you have saved all the files and load the ChIP-Extract data. R code:

load needed functions
source("em_api.R")
load data
CTCF.MNase = as.matrix(read.table("chipextract_CTCF_MNase.mat"))
CTCF.DNaseI = as.matrix(read.table("chipextract_CTCF_DNaseI.mat"))

```
ETS1.MNase = as.matrix(read.table("chipextract_ETS1_MNase.mat"))
ETS1.DNaseI = as.matrix(read.table("chipextract_ETS1_DNaseI.mat"))
```

4. To verify whether CTCF has nicely phased nucleosomes arrays on each sides of its binding sites, and that ETS1 does not, you can draw aggregation plots. An aggregation plot is the type of plot that ChIP-cor returns you. ChIP-Extract data can be used to draw exactly the same plots as you would have had with ChIP-Cor. R code:

```
dist = seq(from=-990, to=990, by=10)
xlab = "Dist to peak"
ylab = "Norm. signal"
```

CTCF

plot(dist, colSums(CTCF.MNase) / max(colSums(CTCF.MNase)), ylim=c(0,1), type='l', lwd=2, main="CTCF", xlab=xlab, ylab=ylab) lines(dist, colSums(CTCF.DNaseI) / max(colSums(CTCF.DNaseI)), ylim=c(0,1), lwd=2, lty=2)

ETS1

plot(dist, colSums(ETS1.MNase) / max(colSums(ETS1.MNase)), ylim=c(0,1), type='l', lwd=2, main="ETS1", xlab=xlab, ylab=ylab) lines(dist, colSums(ETS1.DNaseI) / max(colSums(ETS1.DNaseI)), ylim=c(0,1), lwd=2, lty=2)

overlap

plot(dist, colSums(CTCF.MNase) / max(colSums(CTCF.MNase)), ylim=c(0,1), type='l', lwd=2, main="CTCF", xlab=xlab, ylab=ylab) lines(dist, colSums(CTCF.DNasel) / max(colSums(CTCF.DNasel)), lwd=2, lty=2) lines(dist, colSums(ETS1.MNase) / max(colSums(ETS1.MNase)), lwd=2, lty=1, col='red') lines(dist, colSums(ETS1.DNasel) / max(colSums(ETS1.DNasel)), lwd=2, lty=2, col='red') legend("topright", legend=c("CTCF Mnase", "CTCF DNasel", "ETS1 Mnase", "ETS1 DNasel"), col=c("black", "black", "red", "red"), lwd=c(2,2,2,2), lty=c(1,2,1,2))

```
xlab = ylab = NULL
```

On the plots, a few things are visible. First, CTCF has a strong perdiodic MNase signal around its binding sites. This reflects the presence of well positioned nuclesomes (the peaks are ~140bp wide, a nucleosome core particule covers ~146bp). This is not the case for ETS1 albeit a weak periodic signal is visible. Second, in both cases, there is a clear drop of MNase signal at the binding site. This makes sense if we consider that a DNA binding factor needs the DNA to be accessible in order to bind it. Third, there is a DNaseI hypersensitivity at the level of CTCF and ETS1 binding sites. This reflects a more accessible chromatin structure at the level of the binding site and coroborate what is visible at the nucleosome level. If you look more carefully, you will see a decrease of DNaseI signal exactly on the binding site, which reflects the presence of the factor on the

DNA (which impedes DNA accessibility and its digestion by DNaseI). This mark is called a footprint.

5. Now, let's run the EM partitioning to classify MNase signal and highlight whether different types of nucleosome organization are present in the data. We will allow the algorithm to shift and flip the data.

Nucleosome signal is perdiodic by nature. Allowing the vectors to be phased by shifting them will increase the resolution of the signal. For instance, if the nucleosomes are not always organized the same around the binding site, this can create interference and mess up the nucleosome signal. Hopefully, this can be fixed by allowing shifting. Additionally, allowing algorithm to flip the vector might help resolving symetrical patterns which may originate from the overlap of two asymetrical anti-sense patterns. R code:

All the functions are defined in em_api.R and are commented if needed

EM parameters n.class = 5 n.iter = 5 n.shift = 15 seeding = "random"

plotting parameters
lty = c(1,1)
lwd = c(2,2)
col = c("black", "red")

remove rows containing no MNase counts CTCF.filter = which(apply(CTCF.MNase, 1, sum) == 0) ETS1.filter = which(apply(ETS1.MNase, 1, sum) == 0) CTCF.MNase = CTCF.MNase[-CTCF.filter,] CTCF.DNasel = CTCF.DNasel[-CTCF.filter,] ETS1.MNase = ETS1.MNase[-ETS1.filter,] ETS1.DNasel = ETS1.DNasel[-ETS1.filter,]

em classification with flip and shift of MNase signal set.seed(1) CTCF.FLIP.SHIFT.prob = em.shape.shift.flip(CTCF.MNase, k=n.class, iter=n.iter, shift=n.shift, seeding=seeding) set.seed(1) ETS1.FLIP.SHIFT.prob = em.shape.shift.flip(ETS1.MNase, k=n.class, iter=n.iter, shift=n.shift, seeding=seeding)

plots the results, overlap DNaseI data (flipped and shifted accordingly to MNase)
two figures should appear in your working directory
em.plot.class.profile(list(CTCF.MNase, CTCF.DNaseI), CTCF.FLIP.SHIFT.prob[[1]],
"CTCF", sprintf("CTCF_%dclass_flip_%dshift_random.png", n.class, n.shift), n.iter,
n.shift, flip=T, dist, lty, lwd, col)
em.plot.class.profile(list(ETS1.MNase, ETS1.DNaseI), ETS1.FLIP.SHIFT.prob[[1]],
"ETS1", sprintf("ETS1_%dclass_flip_%dshift_random.png", n.class, n.shift), n.iter,
n.shift, flip=T, dist, lty, lwd, col)

dumps the probability arrays to hard drive, two rds files should appear in your working directory

em.write.prob(CTCF.FLIP.SHIFT.prob[[1]], file=sprintf("CTCF_%dclass_flip_%dshift_random.rds", n.class, n.shift)) em.write.prob(ETS1.FLIP.SHIFT.prob[[1]], file=sprintf("ETS1_%dclass_flip_%dshift_random.rds", n.class, n.shift)) CTCF.FLIP.SHIFT.prob = ETS1.FLIP.SHIFT.prob = NULL











The result figures shows a few interesting things.

Regarding CTCF, a strong periodic MNase signal is still visible. Some classes are asymetric, suggesting that the CTCF sites may have an orientation. Moreover, the different classes show pretty different patterns, supporting different type of nucleosome organization around CTCF binding sites. Interestingly, the DNaseI footprint is still visible which indicates that the data were not shifted too much. This suggests that the nucleosomes are organized with respect to the binding sites (or a feature really close).

Regarding ETS1, we now see nucleosomes clearly visible. However, in contrast to CTCF, DNaseI footprint dissapeared and the DNaseI hypersentitivty region is now wider. This reflects the fact that the MNase data vectors have been shifted. This means that the first nucleosome is not a constant distance from the binding site. This suggests that the nucleosomes are not organized with respect to the binding sites. Additionally, unlike what the aggregation plot was showing, the nucleosome organization around ETS1 binding sites is not anymore symetrical. For instance, class 1 and 3 have a nucleosome array on one side of the binding site only.

6. This algorithm performs a probabilistic (or soft) classification. This means that each binding site is assigned to all classes at the same time, with different probabilities. Nonetheless, it is still possible to extract all binding sites which have been assigned to a class, with a given minimal probability. Interestingly, it is also possible to update the original binding site position to take into account the shift and flip performed by the algorithm. In case, you can download the here : CTCF 5class flip 15shift random.rds classification result file ETS1_5class_flip_15shift_random.rds We know that ETS1 is a factor that mostly bind near TSS. Class 1 shows a profile which is compatible with this. The nucleosome array might reflect the nucleosomes downstream the TSS with the commonly described nucleosome depleted region near the TSS. Let's extract the binding sites which have been classified has belonging to class 1 with an overall probability of 0.9 or higher. R code:

bin.size = 10 # 10bp, value used in ChIP-Extract

reload probability array
ETS1.FLIP.SHIFT.prob = em.read.prob("ETS1_5class_flip_15shift_random.rds")
load original ETS1 peak file
ETS1.sga = read.table("findm_ETS1.sga", stringsAsFactors=F)

remove TFBS which were not containing MNase counts in the ChIP-Extract matrix, as above ETS1.sga = ETS1.sga[-ETS1.filter,]

extract TFBS coordinates and save them as SGA, this should make an sga file appear in your working directory em.extract.unoriented.sga.class.max(ETS1.sga, ETS1.FLIP.SHIFT.prob, sprintf("findm_ETS1_EM_class%d.sga", class), bin.size=bin.size, shift=n.shift, flip=T, class=class, threshold=prob) prob = class = bin.size = NULL ETS1.MNase = CTCF.MNase = ETS1.DNaseI = CTCF.DNaseI = NULL 7. Now, go on ChIP-Cor, upload the coordinates extracted from class 1 and run ChIP-cor 3x as indicated under and save the results in the files named as follows : 'findm_ETS1_EM_class1_TSS.dat', 'findm_ETS1_EM_class1_MNase.dat', 'findm_ETS1_EM_class1_H3K4me3.dat' ChIP-Cor Analysis Module

ChIP-Seq Input Data (Reference Feature)		ChIP-Seq Input Data (Target Featu
iect available Data Sets 🕖	Select available Data S	ets 🕖
rver-resident SCA Files by Filename	Genome :	H. sapiens (Feb 2009/hg19)
	Data Type :	Genome Annotation
oad custom Data 🤨	Serles 🕖 :	EPDnew, the Human Curated Promoter D
SGA 🛟	Sample :	TSS from hg19 EPDnew rel 003
a FILE (gzlp or zlp formats are also accepted): rcourir findm ETS1 EM class1.soa Clear File		
om a URL:	Server-resident SGA Fi	les by Filename
	O Upload custom Data Ø	1
rt Input 🚺 : Off 🖲 on (For SGA only)		Additional Input Data Options
Unknown	Strand 🚺 : O + O -	e any
	Centering () :	
iomes () H. saprens (Feb 2009 GRCh37/hg19)	Bepeat Masker ()	
Additional Input Data Options		Example
nd 🕖 : O + O - O any 🖲 oriented		
lering ():		Submit Reset
peat Masker 🧿		
Analysis Parameters		
nge O		
nning: -999 End: 1000		
Histogram Parameters		
Sunt Cut-off 🙂 : 1		
alization 🔮 : w 🤨 count density 🔷 global		
Cor Analysis Module		
Cor Analysis Module e Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature)		ChIP-Seq Input Data (Target Featu
Cor Analysis Module e Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) iect available Data Sets 0	 Select available Data S 	ChiP-Seq Input Data (Target Featu
Cor Analysis Module e Correlation Tool v1.5.2 ChiP-Seq input Data (Reference Feature) lect available Data Sets 0 rver-resident SGA Files by Filename	Select available Data S Genome :	ChIP-Seq Input Data (Target Featu ots () (H.sapiers (Feb 2009/hg19)
-Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) elect available Data Sets 0 erver-resident SGA Files by Filename	Select available Deta S Genome : Data Type : Data Vype :	ChIP-Seq Input Data (Target Featu ots 0 H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc.
Cor Analysis Module e Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) lect available Data Sets 0 river-resident SGA Files by Filename toad custom Data 0	Select available Data S Genome : Data Type : Series Ø : Series i	ChIP-Seq Input Data (Target Featu ets 0 H. sapiens (Feb 2009/hg19) ENCODE DNass FAIRE etc. CSE35556, Nucleocome Position by MNas Con1278 be income for the same f
Cor Analysis Module e Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) tect available Data Sets () rver-resident SGA Files by Filename load custom Data () SGA () a FILE (gzip or zip formats are also accepted):	Select available Data S Genome : Data Type : Series 0 : Sample :	ChIP-Seq Input Data (Target Featu ets 0 H. sapiers (Feb 2009/hg19) ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by MNas Gm12878 Nucleosome rep1
Cor Analysis Module e Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) tect available Data Sets () river-resident SGA Files by Filename Moda custom Data () SGA () a FILE (gzip or zip formats are also accepted): record:	Solect available Data S Genome : Data Type : Sarles 0 : Sample : Server-resident SGA Fi	ChIP-Seq Input Data (Target Featu ets • H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by MNas Gm12878 Nucleosome rep1
-Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets arver-resident SGA Files by Filename Poload custom Data SGA SGA T A a FLE (gcl or zjp formals are also accepted): arcourt findm_CT31_BM_class1.saga Clear File om a URL:	Select available Data S Genome : Data Type : Sarles ① : Sarles ① : Sample : Server-resident SGA Fil	ChIP-Seq Input Data (Target Featu ets H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by MNas Gm12878 Nucleosome rep1
Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets () rever-resident SGA Files by Filename Noad custom Data () SGA () A FILE (grip or zip formats are also accepted): voormin Inform_ISTL_EM_class1.seg Clear File om a URL: Imput () : of () on (For SGA only)	Select available Data S Genome : Data Type : Series () : Sample : Server-resident SGA FI Upload custom Data ()	ChIP-Seq Input Data (Target Featu eta ChIP-Seq Input Data (Target Featu H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc. C5235586, Nucleosome Position by MNas Cm12878 Nucleosome rep1
-Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) elect available Data Sets erver-resident SGA Flies by Filename pload custom Data	Select available Data S Genome : Data Type : Saries ① : Saries ① : Sample : Upload custom Data ①	ChIP-Seq Input Data (Target Featu ots 7 H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc. C523556, Nucleosome Position by MNas Gm12878 Nucleosome rep1 les by Filename Additional Input Data Options
P-Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) refeat available Data Sets referer-readent SGA Files by Filename reford custom Data SGA SGA	Select available Data S Genome : Data Type : Saries 0 : Sample : Upload custom Data 0 Strand 0 : +	ChIP-Seq Input Data (Target Featu ets 1) H. sapiens (Feb 2009/hg19) ENCODE DNass FAIRE etc. CSE35586, Nucleosome Position by MNas Can12878 Nucleosome rep1 les by Filenamo Additional Input Data Options
Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets rver-resident SGA Files by Filename Noad custom Data SGA SGA	Select available Data S Genome : Data Type : Saries ① : Sample : Upload custom Data ① Strand ① : ● + ● • Centering ① : ⑦ •	ChIP-Seq Input Data (Target Featu eta 1) H. sapiens (Feb 2009/hg19) ENCODE DNase FARE etc. C6235586, Nucleosome Position by MNas Cm12878 Nucleosome rep1 Les by Filename Additional Input Data Options e any
Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets rver-resident SGA Files by Filename Nead custom Data SGA SGA ta FiLE (gzlp or zip formats are also accepted): rcouric tinput tinput	Select available Data S Genome : Data Type : Series 0 : Sample : Server-resident SGA FI Upload custom Data 0 Strand 0 : • + • • Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Featu eta H. sapiero (Feb 2009/hg19) ENCODE DNase FARE etc. CSE3558, Nucleosome Position by MNas Gm12878 Nucleosome rep1 ites by Filename Additional Input Data Options any
-Cor Analysis Module re Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets arver-resident SGA Files by Filename pload custom Data SGA c SGA c s s s s s s s s s s s s s s s s s s	Select available Data S Genome : Data Type : Saries 0 : Sample : Server-resident SGA Fi Upload custom Data 0 Strand 0 : • + • • Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Featu ets ChIP-Seq Input Data (Target Featu ets ChiP-Seq Input Data (Target Featu EKCODE DNase FAIRE etc. CGE35586, Nucleosome Position by MNas Cm12878 Nucleosome rep1 ets by Fliename Additional Input Data Options e any Example
Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) Nect available Data Sets rever-resident SGA Files by Filename Noad custom Data SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) input SGA ChiP-Seq Input Data (Reference Feature) Additional Input Data Options md ChiP-Seq Input SGA ChiP-S	Soloct available Data S Genome : Data Type : Saries 0 : Sample : Server-resident SGA Fi Upload custom Data 0 Strand 0 : • + • • Centering 0 : [70 Repeat Masker 0	ChIP-Seq Input Data (Target Featu ets Core 2009/hg19) ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by Milas Gm12878 Nucleosome rep1 ets by Filename Additional Input Data Options e any Example Submit Reset
Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) Nect available Data Sets arver-resident SGA Files by Filename pload custom Data SGA	Select available Data S Genome : Data Type : Series 0 : Sample : Series 0 : Sample : Upload custom Data 0 Strand 0 : • + • • Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Features of ChIP-Seq Input Data (Target Features (Feb 2009/hg19) ENCODE DNase FARE etc. C6235566, Nucleosome Position by MNas Con12878 Nucleosome rep1 les by Filename Additional Input Data Options e any Example Submit Reset
Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) Sect available Data Sets chiP-Seq Input Data (Reference Feature) Sect available Data Sets sever-resident SGA Files by Filename pload custom Data sca chiP-Seq Input Data (Reference Feature) section of section sca chiP-Seq Input Data (Reference Feature) section section sca chiP-Seq Input Data (Reference Feature) section sca chiP-Seq Input Data (Reference Feature) section sca chiP-Seq Input Data (Reference Feature) sca chiP-Seq Input Data (Reference Feature) sca sca chiP-Seq Input Data Options sca sca sca sca sca sca sca s	 Select available Data S Genome : Data Type : Series Ø : Sample : Server-resident SGA FI Upload custom Data Ø Strand Ø : + + • • Centering Ø : 70 Repeat Masker Ø 	ChIP-Seq Input Data (Target Features to ChIP-Seq Input Data (Target Features (H. sapiens (Feb 2009/hg19) ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by MNas Con12878 Nucleosome rep1 Ites by Filename Additional Input Data Options e any Example Submit Reset
-Cor Analysis Module re Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) elect available Data Sels erver-resident SGA Flies by Filename pload custom Data SGA SGA SGA SGA SGA	Select available Data S Genome : Data Type : Series 0 : Sample : Upload custom Data 0 Strand 0 : + + Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Featu eta () H. sapiens (Feb 2009/hg19) ENCODE DNase FARE etc. C625358, Nucleosome Position by MNas Cm128/8 Nucleosome rep1 Les by Filename Additional Input Data Options e any Example Submit Reset
-Cor Analysis Module re Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) Neet available Data Sets rever-resident SGA Files by Filename Noad custom Data SGA SGA Analysis Parameters repa Manalysis Parameters repa Page Innug : 999 End: 1000	Select available Data S Genome : Data Type : Series 0 : Sample : Upload custom Data 0 Strand 0 : + + • Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Featu ets Collection (Collection) ENCODE DNase FARE etc. COSE3586, Nucleosome Position by MNas Con12878 Nucleosome rep1 Les by Filename Additional Input Data Options any Example Submit Reset
-Cor Analysis Module re Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) elect available Data Sets arver-resident SGA Files by Filename pload custom Data SGA C SGA C n a FLE (gzip or zip formats are also accepted): arcont In fom [ST_EML_dast.taga Clear File Tom a URL: Thput C SGA	Select available Data S Genome : Data Type : Saries 0 : Sample : Upload custom Data 0 Strand 0 : • + • • Centering 0 : 70 Repeat Masker 0	ChiP-Seq Input Data (Target Featu ets ChiP-Seq Input Data (Target Featu ets ChiP-Seq Input Data (Target Featu ExcoDE DNase FAIRE etc. CSE35586, Nucleosome Position by Milas Cm12878 Nucleosome rep1
Cor Analysis Module e Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) lect available Data Sets * rver-resident SGA Files by Filename load custom Data * a FLE (gcip or zip formats are also accepted): crourie finding.ESS_EM_dast.sqa Clear File m a URL: input * : ort * on (For SGA only) ertiment : Urknown urs * :	Solect available Data S Genome : Data Type : Sarles 0 : Sample : Upload custom Data 0 Strand 0 : • + • • Centering 0 : 70 Repeat Masker 0	ChIP-Seq Input Data (Target Features tes ChIP-Seq Input Data (Target Features ENCODE DNase FARE etc. CSE3556, Nucleosome Position by MNas Can12878 Nucleosome rep1 Les by Filename Additional Input Data Options any Example Submit Reset
Cor Analysis Module Correlation Tool v1.5.2 ChiP-Seq Input Data (Reference Feature) sct available Data Sets * re-reaident SGA Files by Filename ad custom Data * FILE (g2p or zip formats are also accepted): additional figure (SGA *) FILE (g2p or zip formats are also accepted): additional figure (SGA *) put * : oft * on (For SGA only) timent: Unknown * re * : Additional Input Data Options d * : + • any * oriented ring * : Analysis Parameters p * pot * Histogram Parameters p * : Histogram Parameters p * : Parameters * p * : Parameters * : Param	Select available Data S Genome : Data Type : Saries 0 : Sample : Server-resident SGA Fil Upload custom Data 0 Strand 0 : + + • • • • • • • • • • • • • • • • •	ChIP-Seq Input Data (Target Features tes ChIP-Seq Input Data (Target Features ENCODE DNase FAIRE etc. CSE35586, Nucleosome Position by MNas Carri2878 Nucleosome rep1 Les by Filename Additional Input Data Options any Example Submit Reset
Cor Analysis Module e Correlation Tool v1.5.2 ChIP-Seq Input Data (Reference Feature) iect available Data Sets 0 rver-resident SGA Files by Filename load custom Data 0 SGA :: a FLE (gcip or zip formats are albo accepted): counting. Infone_ETS1_EM_class1.sea Clear File am a URL: imput 0 : oft 0 on (For SGA only) oriment : Unknown ure 0 : Additional Input Data Options Additional Input Data Options nd 0 : + - any 0 oriented tering 0 : peat Masker 0 Analysis Parameters ngo 0 Histogram Parameters ndow Woldh 0 : 10 un Cut-off 0 1 matization 0 :	Select available Data S Genome : Data Type : Saries 0 : Satample : Server-resident SGA FI Upload custom Data (Strand 0 : + Centering 0 20 Repeat Masker ()	ChIP-Seq Input Data (Target Featu eta () H. sapiens (Feb 2009/hg19) ENCODE DNase FARE etc. CSE3558, Nucleosome Position by MNas Cm12878 Nucleosome rep1 Ice by Filename Additional Input Data Options e any Example Submit Reset

ChIP-Cor Analysis Modul	le			
Feature Correlation Tool v1.5.2				
ChIP-Seq Input Data (Reference Feature)		ChIP-Seq Input Data (Target Feature)		
O Select available Data Sets ()		Select available Data Sets ()		
O Server-resident SGA Files by File	ename	Genome :	H. sapiens (Feb 2009/hg19)	
,		Data Type :	ENCODE ChIP-seq 🗘	
Upload custom Data Ø		Series 🕜 :	GSE29611, Histone Modifications by ChIP 🐥	
to a Fil F (asia a sia tamata an sia	SGA ‡	Sample :	GM12878 H3K4me3	
from a FILE (gzip or zip formats are also accepted): Parcouric findm_ETS1_EM_class1.sga Clear File or from a URL:		Server-resident SGA Files by Filename		
		O Upload custom Data 🕖		
Sort Input 10 : Off 9 on (For SGA only)			Additional Input Data Options	
Experiment :	Unknown	Strand () · · · ·		
Feature 🕖 :			any	
Genomes 🕖	H. sapiens (Feb 2009 GRCh37/hg19) 💲			
	Additional Input Data Options	Repeat Masker 🖤		
Strand 10 : O + O - O any 😕 oriented			Example	
Centering 🕖 :			Submit Reset	
Repeat Masker 0				
Analysis Parameters				
Range 10				
Beginning : -999	End : 1000			
	Histogram Parameters			
🖲 Window Width 🛈 : 10				
Count Cut-off : 1				
Normalization 0:				
🔿 raw 🛛 🖲 count density 🔷 global				

8. Go back to your R session and draw an aggregation plots with all the ChIP-Cor data. R code:

```
# load ChIP-Cor data
ETS1.TSS = read.table("findm_ETS1_EM_class1_TSS.dat")
ETS1.MNase = read.table("findm_ETS1_EM_class1_MNase.dat")
ETS1.H3K3me3 = read.table("findm_ETS1_EM_class1_H3K4me3.dat")
# plot chipcor data
xlab = "Dist to peak"
ylab = "Norm. signal"
plot(ETS1.MNase[,1], ETS1.MNase[,2] / max(ETS1.MNase[,2]), ylim=c(0,1), type='l',
lwd=2, main="ETS1 class 1", xlab=xlab, ylab=ylab)
lines(ETS1.H3K3me3[,1], ETS1.H3K3me3[,2] / max(ETS1.H3K3me3[,2]),
ylim=c(0,1), lwd=2, lty=1, col="red")
lines(ETS1.TSS[,1], ETS1.TSS[,2] / max(ETS1.TSS[,2]), ylim=c(0,1), lwd=2, lty=1,
col="blue")
legend("topright", legend=c("MNase", "H3K4me3", "TSS"), lwd=c(2,2,2), lty=c(1,1,1),
col=c("black", "red", "blue"))
```



The peak of TSS density in the nucleosome depleted region together with the H3K4me3 labelling suggest that the binding sites which have been assigned to this class are indeed in promoters regions. However, the exact orientation of these promoters is hard to estimate since H3K4me3 labelling is pretty symetrical and that we can see nucleosomes on both sides of the TSS density peak.

Now, you might want to pursue this analysis by extracting other classes coordinates or to rerun an entire analysis on other type of data such as histone marks.

Part IV: H3K4me3 around human promoters. Dataset in this example is from GEO series GSE29611 and EPDNew.

We will use ChIP-Extract Analysis Module to generate a tag count matrix in defined bins around promoters. Select appropriate target feature and then click submit.

- o Reference feature: hg19; Genome Annotation; EPDNew; rel 003
- Target feature: hg19; ENCODE ChIP-seq; GSE29611; GM12878 H3K4me3
- Analysis parameters: -1000 to 1000; Window width:10; Count cut-off:10

Download the Ref SGA File and Table (TEXT) and save as promoters.sga and h3k4me3_promoters.txt respectively.

Perform iterative shape based partitioning with shifting and write class output as sga files. R script

```
Read data:
    data = as.matrix(read.table("h3k4me3_promoters.txt"))
Define classes and shifts:
    K=4; S=3; N=dim(data)[1]; L=dim(data)[2]-S+1; ITER=5
```

```
Iterative shape based partitioning with shifting
      mean shift=floor(S/2)+1
      c = colMeans(data[,mean_shift:(mean_shift+L-1)])
      flat = matrix(data=mean(data), nrow=1, ncol=L)
      q=q0 = dnorm(1:S,mean shift,1)/sum(dnorm(1:S,mean shift,1))
      for (m in 1:(K-1)) {
      c = rbind(flat, c)
      q = rbind(q0/m,q); q/sum(q)
      for(i in 1:ITER) {
      reg shift(q); em shape shift(c,q,data); plot classes(c);
      print(q)}
      }
Extract class labels
      class=apply(apply(p,c(1,2),sum),1,order)[4,]
Write output sga files for different classes
      peak <- read.table("promoters.sga", header=F, sep="\t")</pre>
      write.table(peak[which(class==1),], "h3k4me3_class1.sga",
      sep="\t", quote=F, row.names=F, col.names=F)
      write.table(peak[which(class==2),], "h3k4me3_class2.sga",
      sep="\t", quote=F, row.names=F, col.names=F)
      write.table(peak[which(class==3),], "h3k4me3_class3.sga",
      sep="\t", quote=F, row.names=F, col.names=F)
      write.table(peak[which(class==4),], "h3k4me3_class4.sga",
      sep="\t", quote=F, row.names=F, col.names=F)
```

Now you may want to explore these four classes in-terms of associated features, such as PolII, conservation, or check the enrichment for various core promoter motifs (TATA, CCAAT-box, etc).