iChip: A Package for Analyzing Multi-platform ChIP-chip data with Various Sample Sizes

Qianxing Mo

April 27, 2020

Department of Biostatistics & Bioinformatics
H. Lee Moffitt Cancer Center & Research Institute
12902 Magnolia Drive, Tampa, FL 33612
qianxing.mo@moffitt.org

Contents

1 Introduction **1**

2 Agilent and Affymetrix ChIP-chip Data **2**

3 Example1 — Analyzing the Agilent Promoter Array Data **2**

4 Example2 — Analyzing the Affymetrix Tiling Array Data **8**

5 Tips **13**

6 Parallel Computaton **14**

7 Citing iChip **14**

1 Introduction

This package implements the models proposed by Mo and Liang (2010a, b) for ChIP-chip data analysis. The package can be used to analyze ChIP-chip data from multiple platforms (e.g. Affymetrix, Agilent, and NimbleGen) with various genomic resolutions and various sample sizes. Mo and Liang (2010a,b) proposed Bayesian Hierarchical models to model ChIP-chip data in which the spatial dependency of the data is modeled through ferromagnetic high-order or standard Ising models. Briefly, without loss of generality, the proposed methods let each probe be associated with a binary latent variable $X_i \in (0, 1)$, where i denotes the ID for the probe, and $X_i = 1$ denotes that the probe is an enriched probe, and 0 otherwise. In the first stage, conditioning on the latent variable, the probe enrichment measurements for each state (0 or 1) are modeled by normal distributions. Here, the probe enrichment measurement could be any appropriate measurement for comparison

of IP-enriched and control samples. For example, the measurement could be a log2 ratio of IP-enriched and control samples for a single replicate, or a summary statistic such as t-like statistic or mean difference for multiple replicates. In the second stage, the latent variable is modeled by ferromagnetic Ising models. The Gibbs sampler and Metropolis algorithm are used to simulate from the posterior distributions of the model parameters, and the posterior probabilities for the probes in the enriched state ($X_i = 1$) are used for statistical inference. A probe with a high posterior probability of the enriched state will provide strong evidence that the probe is an enriched probe. For further details, we refer the user to Mo and Liang's papers.

2 Agilent and Affymetrix ChIP-chip Data

A subset of the Oct4 (Boyer et al., 2005) and the p53 (Cawley et al, 2004) data are used for the purpose of illustration. The average genomic resolutions for the Oct4 and p53 data are about 280 bps and 35 bps, respectively. Both the Oct4 and p53 data have been log2 transformed and quantile-normalized. Note iChip software doesn't provide functions for data normalization. The users should normalize their data before using iChip software. For one-color and two-color data, one can use the quantile method (e.g., see the function *normalize.quantiles()* in the **affy** package). For two-color data, one can also use the *loess* method (e.g., see the function *normalizeWithinArrays()* in the **limma** package). The full Oct4 data can be obtained from
`http://jura.wi.mit.edu/young_public/hESregulation/Data_download.html`.
The full p53 data can be obtained from
`http://www.gingeras.org/affy_archive_data/publication/tfbs/`.

3 Example1 — Analyzing the Agilent Promoter Array Data

Let's start analyzing the low resolution Oct4 data. First, we need to calculate the enrichment measurement for each probe. Although the enrichment measurement could be any appropriate measurement for comparison of IP-enriched and control samples, we suggest using the empirical Bayes t-statistic for multiple replicates, which can be easily calculated using the **limma** package (Smyth, 2004). Here, we call the empirical Bayes t-statistic limma t-statistic. For the users who are not familiar with limma t-statistic, we provide a wrapper function **lmtstat** for the calculation.

There are two replicates for the Oct4 data. The enriched DNA was labeled with Cy5 (red) dye and the control DNA was labeled with Cy3 (green) dye.

```
> library(iChip)
> data(oct4)
> head(oct4,n=3L)
```

```
  chr position    green1    green2     red1      red2
1  20    70312  6.969102  6.847819 6.808445  7.063581
2  20    70601  6.625190  6.176981 6.996920  6.391692
3  20    70873 10.334613 11.072903 9.521095 10.785880
```

To use the iChip1 and iChip2 function, the data must be sorted, firstly by chromosome then by genomic position. It may be a good habit to sort the data at the beginning, although function **lmtstat** doesn't require the data to be sorted.

```
> oct4 = oct4[order(oct4[,1],oct4[,2]),]
```

Calculate the enrichment measurements — two-sample limma t-statistics.

```
> oct4lmt = lmtstat(oct4[,5:6],oct4[,3:4])
```

Here, we treat the IP-enriched and control data as independent data although both the IP-enriched and control samples were hybridized to the same array. This is because the quantile-normalization method was applied to the oct4 data. If the data are normalized using *loess* method, the resulting data are in log ratio format (e.g., log2(IP-enriched/control)). In this case, one can calculate the paired limma t-statistics. Suppose a matrix called log2ratio are the loess-normalized data, where each column corresponds to a sample, the paired limma t-statistics can be calculated using **lmtstat**(log2ratio).

Prepare the data for iChip2 function.

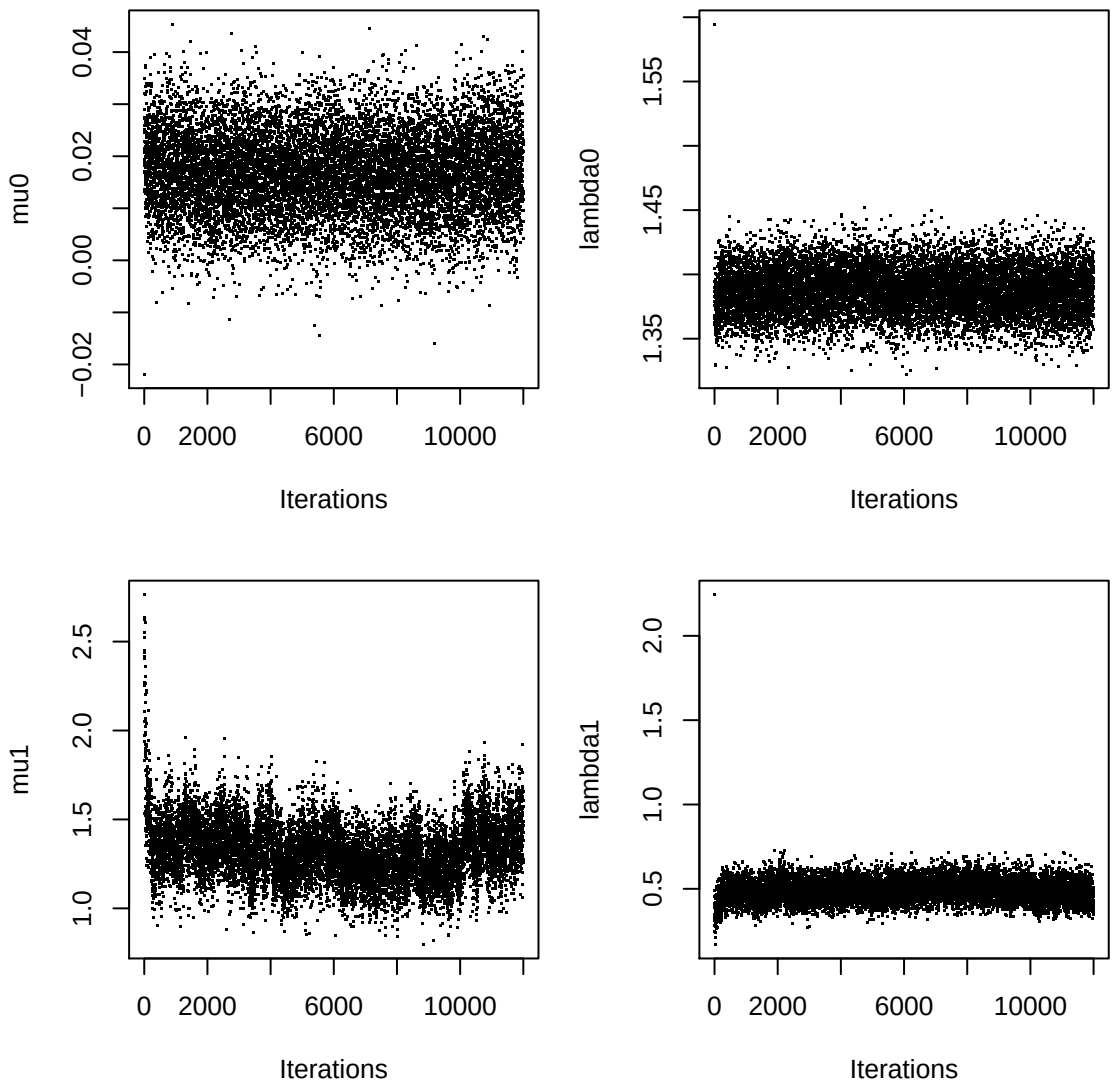
```
> oct4Y = cbind(oct4[,1],oct4lmt)
```

Apply the second-order Ising model to the ChIP-chip data by setting winsize = 2. According to our experience, a balance between high sensitivity and low FDR can be achieved when winsize = 2. The critical value of the second-order Ising model is about 1.0. For low resolution data, the value of beta could be around the critical value. In general, increasing beta value will lead to less enriched regions, which amounts to setting a stringent criterion for detecting enriched regions.

```
> set.seed(777)
> oct4res2 = iChip2(Y=oct4Y,burnin=2000,sampling=10000,winsize=2,
+   sdcut=2,beta=1.25,verbose=FALSE)
```

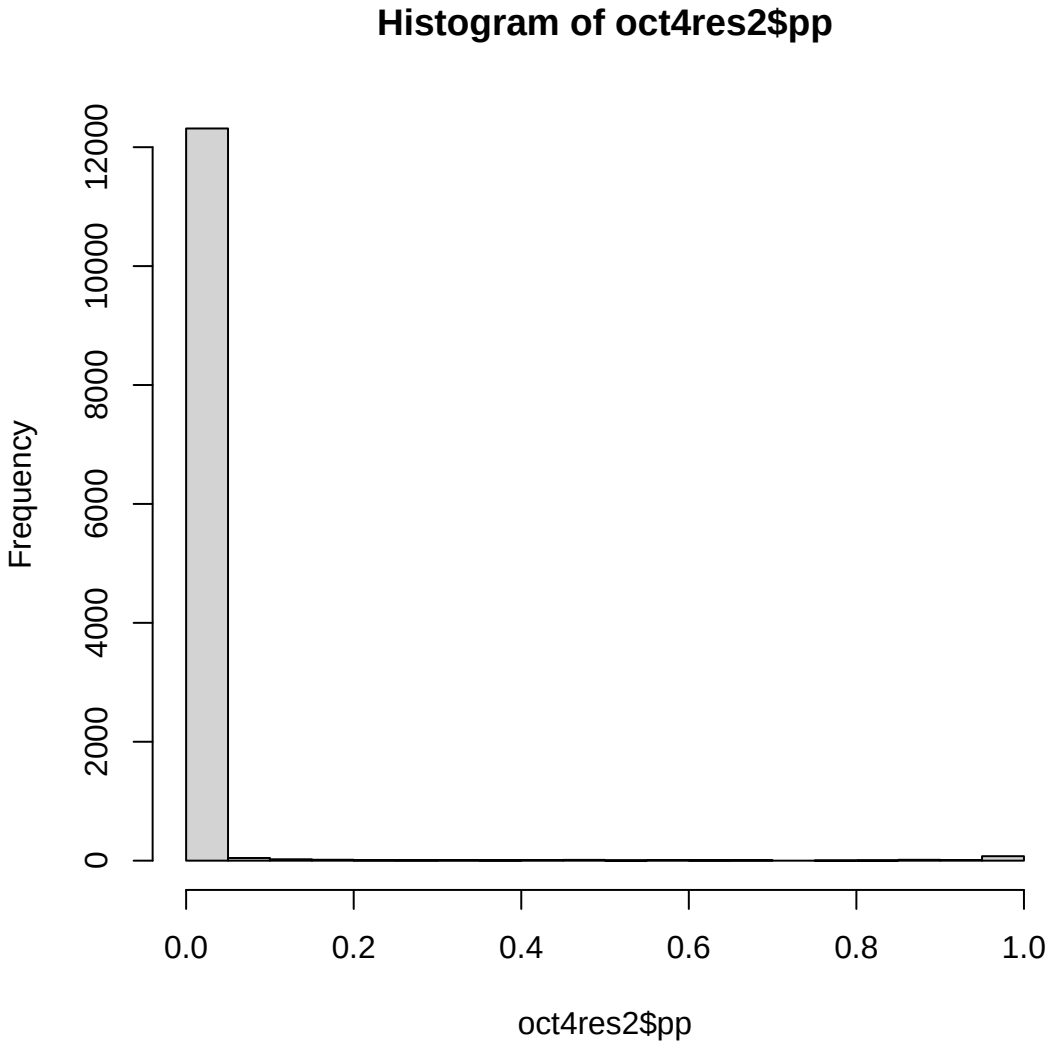
Plot the model parameters to see whether they converge. In general, the MCMC chains have converged when the parameters fluctuate around the modes of their distributions. If there is an obvious trend(e.g. continuous increase or decrease), one should increase the number of iterations in the burn-in phase. If this doesn't work, one can adjust the parameter **beta** to see how it affects the results.

```
> par(mfrow=c(2,2), mar=c(4.1, 4.1, 2.0, 1.0))
> plot(oct4res2$mu0, pch=".", xlab="Iterations", ylab="mu0")
> plot(oct4res2$lambda0, pch=".", xlab="Iterations", ylab="lambda0")
> plot(oct4res2$mu1, pch=".", xlab="Iterations", ylab="mu1")
> plot(oct4res2$lambda1, pch=".", xlab="Iterations", ylab="lambda1")
```



The histogram of the posterior probabilities should be dichotomized, either 0 or 1. For transcription factor binding site studies, the histogram should be dominated by 0.

```
> hist(oct4res2$pp)
```



Call the enriched regions detected by iChip2 using a posterior probability (pp) cutoff of 0.9.

```
> reg1 = enrichreg(pos=oct4[,1:2],enrich=oct4lmt,pp=oct4res2$pp,
+   cutoff=0.9,method="ppcut",maxgap=500)
> print(reg1)
```

	chr	gstart	gend	rstart	rend	peakpos	meanpp	maxpp	nprobe
1	20	3944132	3946241	1415	1427	3946061	0.96	1	13
2	20	20291072	20291658	3293	3295	20291658	0.96	1	3
3	20	20292352	20294499	3296	3304	20293941	1.00	1	9
4	20	21441187	21450238	3441	3477	21445231	0.99	1	37
5	20	22519126	22519690	3545	3547	22519406	1.00	1	3
6	20	28137489	28138889	4307	4312	28137489	1.00	1	6

```
7  20 34633143 34633770   6132 6134 34633506   0.98     1      3
8  20 44034352 44034352   8313 8313 44034352   1.00     1      1
9  20 54633181 54635934   9703 9713 54633459   0.99     1     11
```

Call the enriched regions detected by iChip2 using a FDR cutoff of 0.01. The FDR is calculated using a direct posterior probability approach (Newton et al., 2004).

```
> reg2 = enrichreg(pos=oct4[,1:2],enrich=oct4lmt,pp=oct4res2$pp,
+   cutoff=0.01,method="fdrcut",maxgap=500)
> print(reg2)

  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  20  3944132  3946241   1415 1427  3946061   0.96     1     13
2  20 20291344 20291658   3294 3295 20291658   0.98     1      2
3  20 20292352 20294499   3296 3304 20293941   1.00     1      9
4  20 21441187 21449717   3441 3475 21445231   1.00     1     35
5  20 22519126 22519690   3545 3547 22519406   1.00     1      3
6  20 28137489 28138889   4307 4312 28137489   1.00     1      6
7  20 34633143 34633770   6132 6134 34633506   0.98     1      3
8  20 44034352 44034352   8313 8313 44034352   1.00     1      1
9  20 54633181 54635934   9703 9713 54633459   0.99     1     11
```

BED file can be easily made using the output from function **enrichreg**, which can be used for motif discovery and visualized in the UCSC genome browser. For example,

```
> bed1 = data.frame(chr=paste("chr",reg2[,1],sep=""),reg2[,2:3])
> print(bed1[1:2,])

    chr   gstart     gend
1 chr20  3944132  3946241
2 chr20 20291344 20291658
```

Alternatively, one may create a BED file using the peak position of the enriched regions. For example,

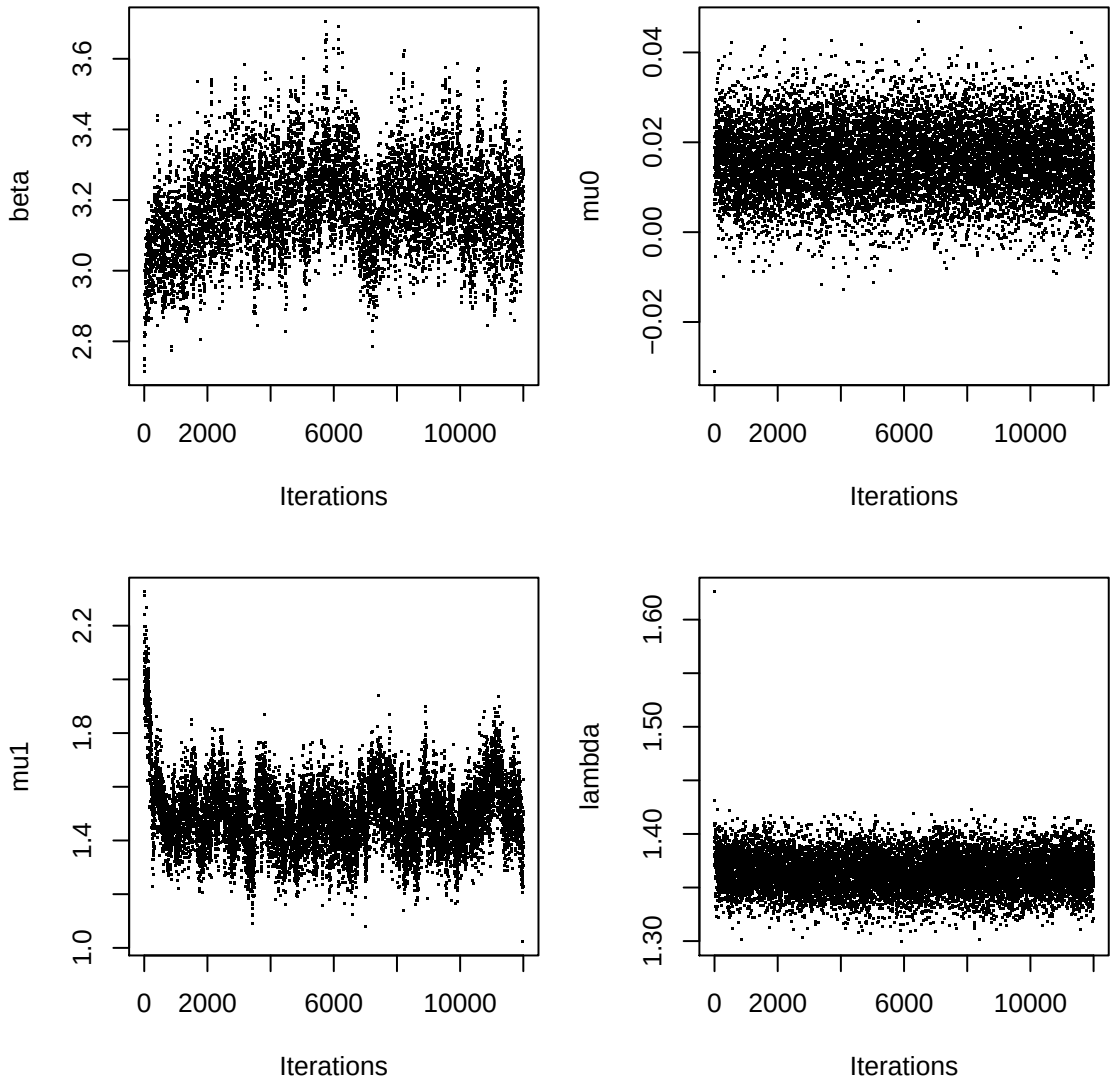
```
> bed2 = data.frame(chr=paste("chr",reg2[,1],sep=""),gstart=reg2[,6]-100,
+   gend=reg2[,6]+100)
> print(bed2[1:2,])

    chr   gstart     gend
1 chr20  3945961  3946161
2 chr20 20291558 20291758
```

Model the oct4 data using the first-order Ising model.

```
> oct4res1 =iChip1(enrich=oct4lmt,burnin=2000,sampling=10000,sdcut=2,beta0=3,
+   minbeta=0,maxbeta=10,normsd=0.1,verbose=FALSE)
```

Plot the model parameters to see whether they converge.

```
> par(mfrow=c(2,2), mar=c(4.1, 4.1, 2.0, 1.0))
> plot(oct4res1$beta, pch=".", xlab="Iterations", ylab="beta")
> plot(oct4res1$mu0, pch=".", xlab="Iterations", ylab="mu0")
> plot(oct4res1$mu1, pch=".", xlab="Iterations", ylab="mu1")
> plot(oct4res1$lambda, pch=".", xlab="Iterations", ylab="lambda")
```



Call the enriched regions detected by iChip1.

```
> enrichreg(pos=oct4[,1:2],enrich=oct4lmt,pp=oct4res1$pp,cutoff=0.9,
+           method="ppcut",maxgap=500)
```

```
  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  20  3944132  3946241   1415 1427  3946061   1.00     1     13
```

```
2  20 20291072 20291658   3293 3295 20291658   0.96     1      3
3  20 20292352 20294805   3296 3305 20293941   0.99     1     10
4  20 21441187 21449717   3441 3475 21445231   0.99     1     35
5  20 22519126 22519690   3545 3547 22519406   1.00     1      3
6  20 28137489 28138889   4307 4312 28137489   0.99     1      6
7  20 34632735 34633770   6131 6134 34633506   0.97     1      4
8  20 54633181 54636203   9703 9714 54633459   0.99     1     12
```

```
> enrichreg(pos=oct4[,1:2],enrich=oct4lmt,pp=oct4res1$pp,cutoff=0.01,
+           method="fdrcut",maxgap=500)
```

```
  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  20  3944132  3946241   1415 1427  3946061   1.00     1     13
2  20 20291072 20291658   3293 3295 20291658   0.96     1      3
3  20 20292352 20294805   3296 3305 20293941   0.99     1     10
4  20 21441187 21449717   3441 3475 21445231   0.99     1     35
5  20 22519126 22519690   3545 3547 22519406   1.00     1      3
6  20 28137489 28138889   4307 4312 28137489   0.99     1      6
7  20 34632735 34633770   6131 6134 34633506   0.97     1      4
8  20 54633181 54636203   9703 9714 54633459   0.99     1     12
```

4 Example2 — Analyzing the Affymetrix Tiling Array Data

Now, let's analyze the high resolution p53 data.

```
> data(p53)
> head(p53,n=3L)
```

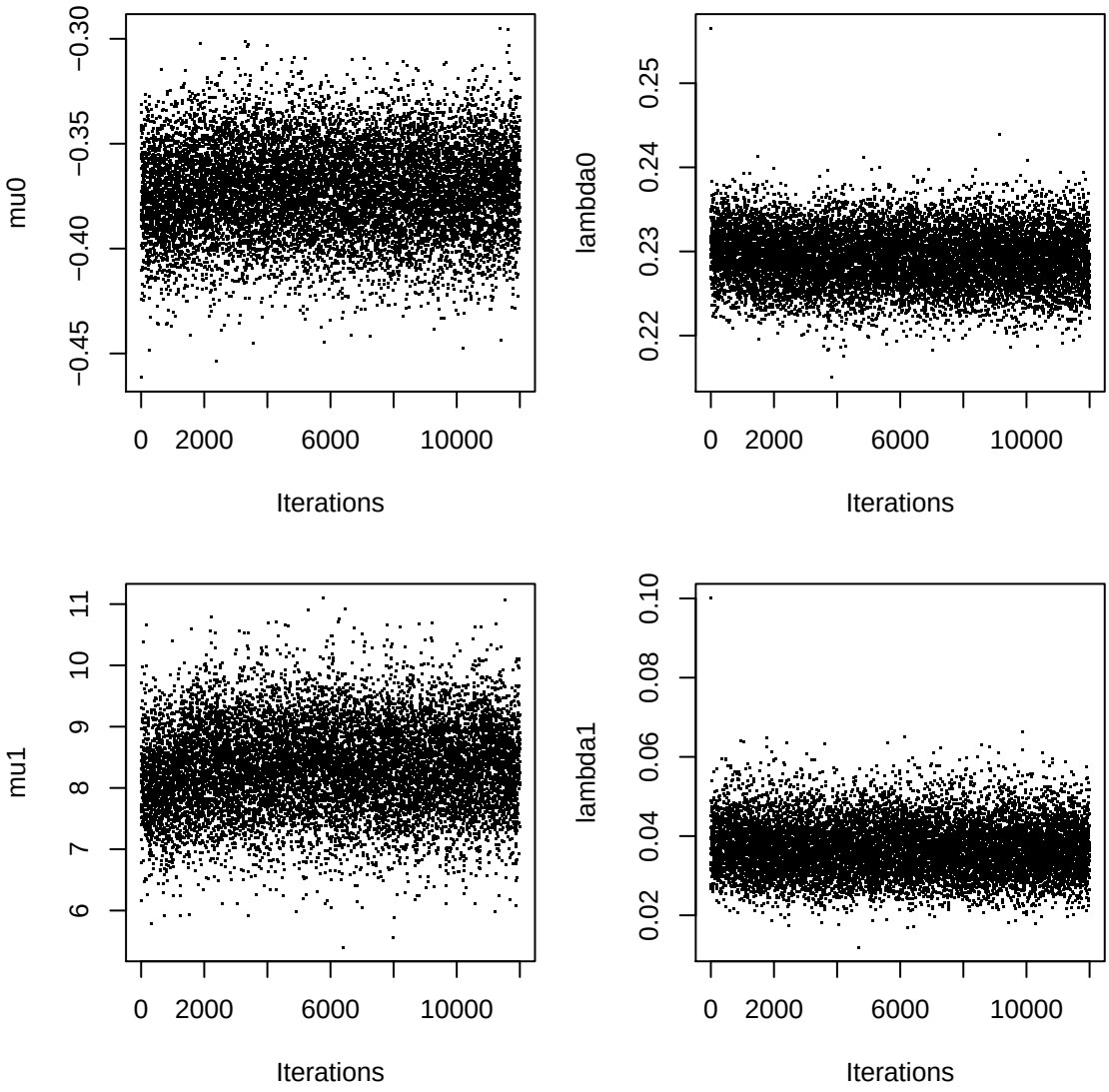
```
       chr position      CON      CON       CON      CON      CON      CON
783581  22 27980300 9.576077 10.90728  9.585894 11.14744 10.23070 11.01191
783582  22 27980347 9.941713 10.66333 10.031774 10.86761 10.23361 10.81113
783583  22 27980372 9.932955 10.63290  9.995038 10.42966 10.02872 10.39427
             IP       IP       IP        IP        IP       IP
783581  9.70315 10.64889 10.53961  9.376407  9.869731 10.89024
783582 10.32478 10.39944 10.39757 10.602358 10.544956 10.24796
783583 10.27690 10.28804 10.27836 10.244013  9.961192 10.18641
```

```
> # sort the p53 data by chromosome and genomic position
> p53 = p53[order(p53[,1],p53[,2]),]
> p53lmt = lmtstat(p53[,9:14],p53[,3:8])
> p53Y = cbind(p53[,1],p53lmt)
```

For high resolution data, beta could be set to a relatively large value (e.g. 2–4). In general, increasing beta value will lead to less enriched regions, which amounts to setting a stringent criterion for detecting enriched regions.

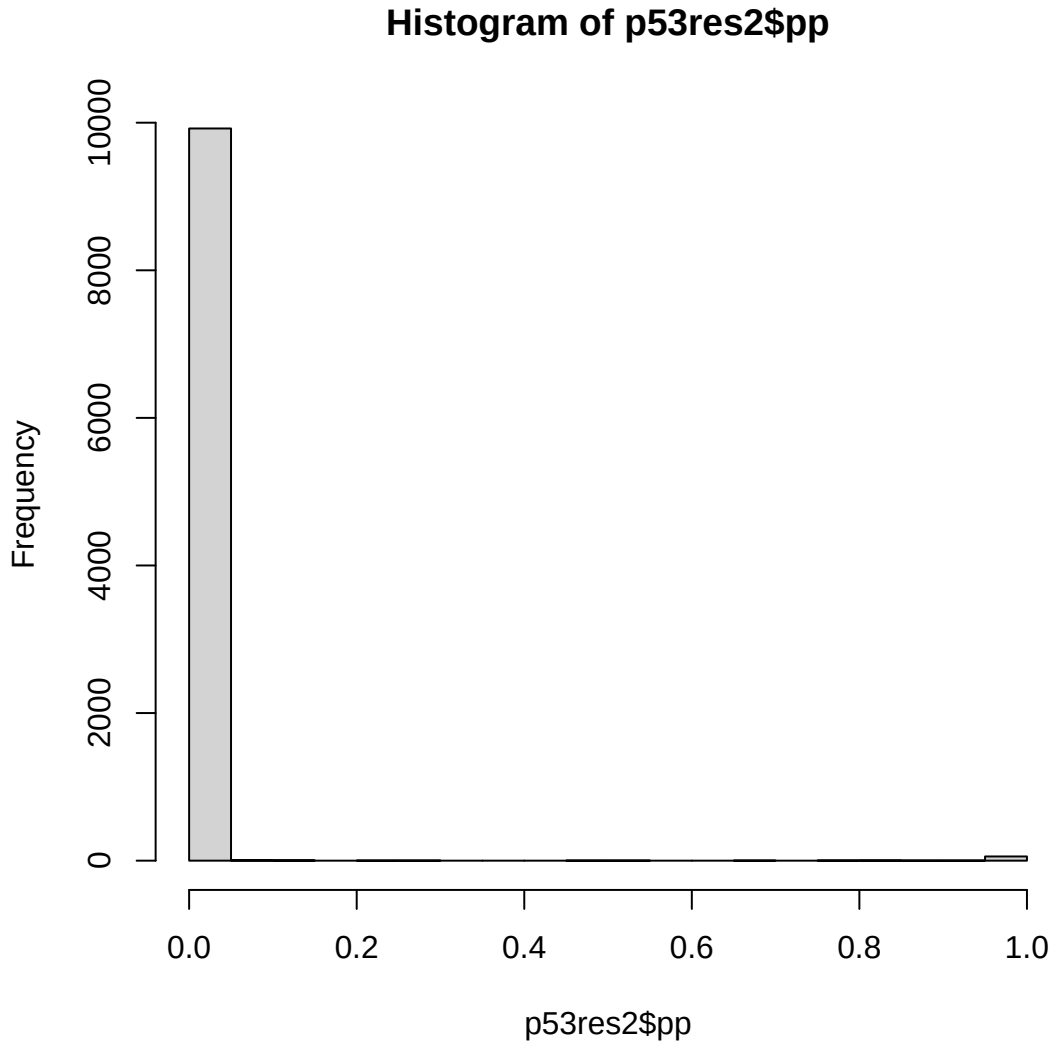
```
> p53res2 = iChip2(Y=p53Y,burnin=2000,sampling=10000,winsize=2,sdcut=2,beta=2.5)
```

```
> par(mfrow=c(2,2), mar=c(4.1, 4.1, 2.0, 1.0))
> plot(p53res2$mu0, pch=".", xlab="Iterations", ylab="mu0")
> plot(p53res2$lambda0, pch=".", xlab="Iterations", ylab="lambda0")
> plot(p53res2$mu1, pch=".", xlab="Iterations", ylab="mu1")
> plot(p53res2$lambda1, pch=".", xlab="Iterations", ylab="lambda1")
```



The histogram of the posterior probabilities should be dichotomized, either 0 or 1. For transcription factor binding site studies, the histogram should be dominated by 0.

```
> hist(p53res2$pp)
```

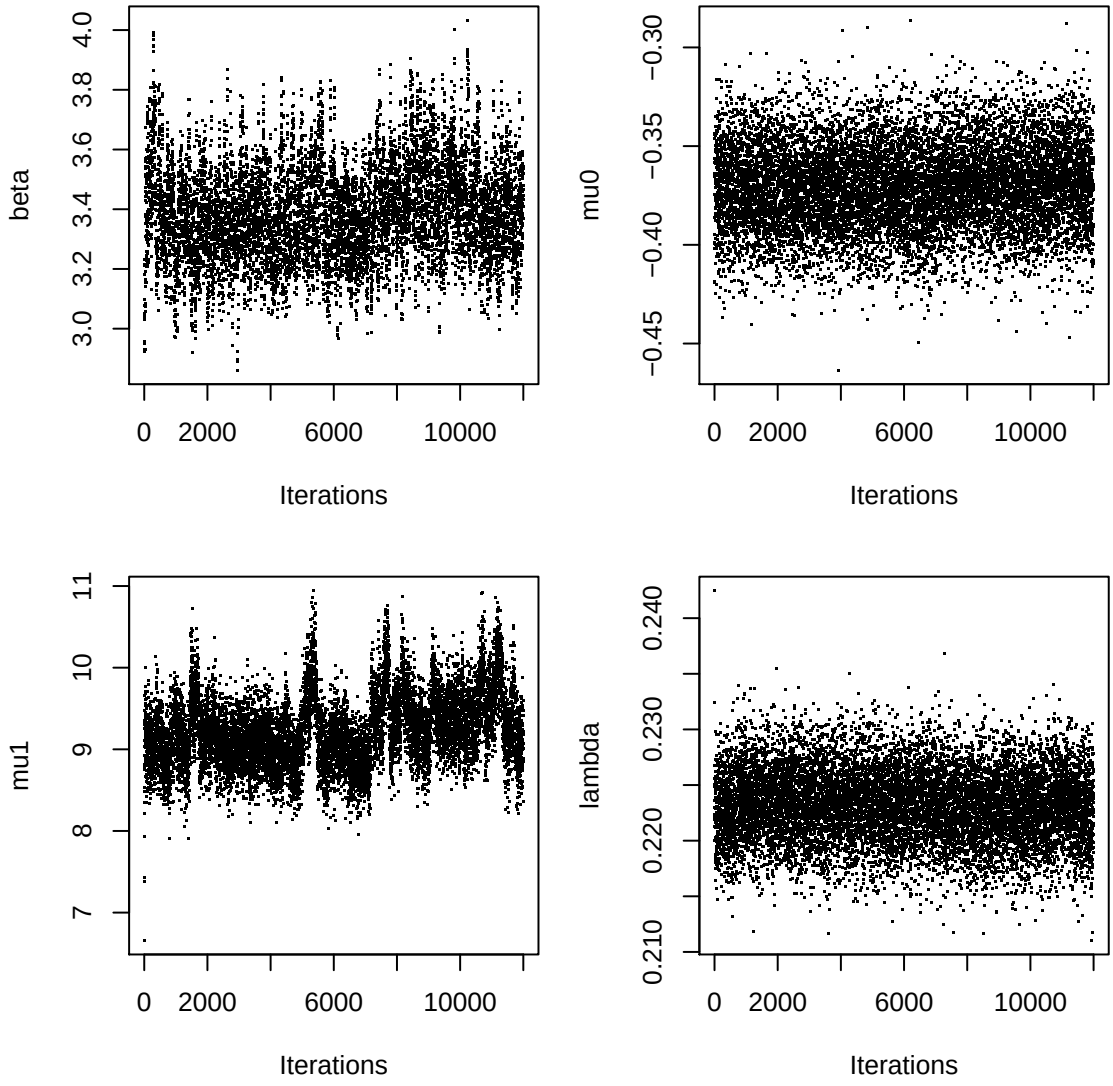



```
> enrichreg(pos=p53[,1:2],enrich=p53lmt,pp=p53res2$pp,cutoff=0.9,
+           method="ppcut",maxgap=500)

  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  22 28211540 28211540   2991 2991 28211540   0.96  0.96      1
2  22 28269526 28270158   3705 3724 28269751   1.00  1.00     20
3  22 28345939 28346341   4831 4842 28346226   1.00  1.00     12
4  22 28380484 28380676   5440 5446 28380676   1.00  1.00      7
5  22 28775272 28775790   9878 9895 28775550   1.00  1.00     18

> enrichreg(pos=p53[,1:2],enrich=p53lmt,pp=p53res2$pp,cutoff=0.01,
+           method="fdrcut",maxgap=500)
```

```
  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  22 28211540 28211540   2991 2991 28211540   0.96  0.96      1
2  22 28269497 28270158   3704 3724 28269751   0.99  1.00     21
3  22 28345939 28346341   4831 4842 28346226   1.00  1.00     12
4  22 28380484 28380725   5440 5447 28380676   0.98  1.00      8
5  22 28775272 28775790   9878 9895 28775550   1.00  1.00     18
```

Model the p53 data using the first-order Ising model.

```
> p53res1 =iChip1(enrich=p53lmt,burnin=2000,sampling=10000,sdcut=2,beta0=3,
+   minbeta=0,maxbeta=10,normsd=0.1,verbose=FALSE)

> par(mfrow=c(2,2), mar=c(4.1, 4.1, 2.0, 1.0))
> plot(p53res1$beta, pch=".", xlab="Iterations", ylab="beta")
> plot(p53res1$mu0, pch=".", xlab="Iterations", ylab="mu0")
> plot(p53res1$mu1, pch=".", xlab="Iterations", ylab="mu1")
> plot(p53res1$lambda, pch=".", xlab="Iterations", ylab="lambda")
```




```
> enrichreg(pos=p53[,1:2],enrich=p53lmt,pp=p53res1$pp,cutoff=0.9,
+           method="ppcut",maxgap=500)
```

	chr	gstart	gend	rstart	rend	peakpos	meanpp	maxpp	nprobe
1	22	28211540	28211540	2991	2991	28211540	1.00	1	1
2	22	28269526	28270058	3705	3721	28269751	0.97	1	17
3	22	28345939	28346341	4831	4842	28346226	1.00	1	12
4	22	28380574	28380676	5443	5446	28380676	1.00	1	4
5	22	28775272	28775790	9878	9895	28775550	0.99	1	18

```
> enrichreg(pos=p53[,1:2],enrich=p53lmt,pp=p53res1$pp,cutoff=0.01,
+           method="fdrcut",maxgap=500)
```

```
  chr   gstart     gend rstart rend  peakpos meanpp maxpp nprobe
1  22 28211540 28211540   2991 2991 28211540   1.00     1      1
2  22 28269526 28270058   3705 3721 28269751   0.97     1     17
3  22 28345939 28346341   4831 4842 28346226   1.00     1     12
4  22 28380546 28380676   5442 5446 28380676   0.97     1      5
5  22 28775272 28775790   9878 9895 28775550   0.99     1     18
```

5 Tips

What happens when there is no enriched region? Suppose the data are just random noises.

```
> randomY = cbind(p53[,1],rnorm(10000,0,1))
> randomres2 = iChip2(Y=randomY,burnin=1000,sampling=5000,winsize=2,
+   sdcut=2,beta=2.5,verbose=FALSE)

Warning: all probes are in the same state at the last MCMC iteration.
 NO enriched region is found!

> table(randomres2$pp)

    1
10000
```

In this case, all the probes are only in one state. Since there is no enriched probe, the mean and variance become $-\infty$ or ∞. In the MCMC simulations, we relabel the outputs according to the constraint $\mu_0 < \mu_1$, where μ_0 and μ_1 are the population means for the non-enriched and enriched probes, respectively (For details, see Mo and Liang, 2010a). That is, when $\mu_0 > \mu_1$, μ_0 will be treated as the population mean of the enriched probes. As a result, no matter $\mu_1 = \infty$ or $\mu_1 = -\infty$, the posterior probabilities of the probes are all 1s or close to 1. Therefore, when this happens, it means there is no enriched region.

In addition, for the studies of transcription factor binding sites, if the posterior probabilities are not dichotomized and dominated by 0, the Ising model is not in the super-paramagnetic phase. Only the super-paramagnetic phase reflects the binding events on the chromosomes. Therefore, the user should increase the value of beta to let the phase transition occur so that the Ising model reach the super-paramagnetic phase.

The probes' states reported by iChip1 are also in the same state if there is no enriched region.

```
> randomres1 =iChip1(enrich=randomY[,2],burnin=1000,sampling=5000,sdcut=2,
+   beta0=3,minbeta=0,maxbeta=10,normsd=0.1,verbose=FALSE)

Warning: all probes are in the same state at the last MCMC iteration.
 NO enriched region is found!

> table(randomres1$pp)
```

```
    0
10000
```

Although the above two examples only show the analysis for the data on a single chromosome, one can use iChip2 and iChip1 functions to analyze data with multiple chromosomes. Although a probe may not be physically close to its adjacent probes (e.g., the last probe of a chromosome and the first probe of the next chromosome, and the probes in the same chromosome that are adjacent but separated by a long genomic distance), in practice, it should be acceptable to consider the interactions between these adjacent and boundary probes. There are two reasons for this. First, the number of these probes is quite small, compared to all the probes in the tiling arrays. Second, these boundary probes have a very high probability of being non-enriched, thus it should be reasonable to consider the interactions between them. If we let these boundary and adjacent probes interact with each other, it has little effect on the results, but significantly simply the algorithms for modeling ChIP-chip data. In addition, it should be noted that when the data are very noisy, the posterior mean of beta will be relatively small(e.g., around 1) when the iChip1 method is used, and the posterior probabilities are not dichotomized and dominated by 0. In this case, the user should increase the value of parameter **minbeta** or use the **iChip2** method for modeling.

6 Parallel Computaton

If the total number of probes is relatively small (e.g., a half million), one may analyze the data in a single run. If the total number of probes are large (e.g., several millions), one may perform parallel computation. For example, one can model the data chromosome by chromosome and run many jobs simultaneously. The following is an example of parallel computation using the snowfall package (`http://cran.r-project.org/web/packages/snowfall/`).

```
library(snow)
library(snowfall)
dataList = list(oct4t=oct4lmt,p53t=p53lmt)
sfInit(parallel=TRUE,cpus=2,type="SOCK")
res=sfLapply(dataList,iChip1,burnin = 2000, sampling = 10000, sdcut = 2,
beta0 = 3,minbeta = 0, maxbeta = 10, normsd = 0.1, verbose = FALSE)
```

In addition, Mo (2012) has applied Ising models to analysis of ChIP-seq data, and provided an R script named 'iSeq.R' that can be used as a command line program in Unix/Linux environment. If the user is interested in it, the user may modify the script to automate ChIP-chip data analyis. The R script iSeq.R is available at
`https://sites.google.com/site/quincymobio/teaching-materials`.

7 Citing iChip

If you use iChip2 function, please cite Mo and Liang (2010a). If you use iChip1 function, please cite Mo and Liang (2010b).

```
> sessionInfo()

R version 4.0.0 (2020-04-24)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Mojave 10.14.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] iChip_1.42.0

loaded via a namespace (and not attached):
[1] compiler_4.0.0 limma_3.44.0   tools_4.0.0
```

References

Mo, Q., Liang, F. (2010a). Bayesian Modeling of ChIP-chip data through a high-order Ising Model. *Biometrics* 66(4), 1284-1294.

Mo, Q., Liang, F. (2010b). A hidden Ising model for ChIP-chip data analysis. *Bioinformatics* **26(6)**, 777-783.

Mo, Q. (2012). A fully Bayesian hidden Ising model for ChIP-seq data analysis. *Biostatistics* **13(1)**, 113-28.

Boyer, L.A., Lee, T.I., Cole, M.F., et al. (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell* **122**, 947-956.

Cawley, S., Bekiranov, S., Ng, H.H., et al. (2004). Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell* **116**, 499-509.

Newton, M., Noueiry, A., Sarkar, D., Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics* **5** , 155-176.

Smyth, G. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology,* **3**, Iss. 1, Article 3.