iSeq: Bayesian Hierarchical Modeling of ChIP-seq Data Through Hidden Ising Models

Qianxing Mo

October 18, 2010

Department of Epidemiology and Biostatistics
Memorial Sloan-Kettering Cancer Center
moq@mskcc.org

Contents

1 Introduction

This package implements the models proposed by Mo (2010) for ChIP-seq data analysis, which are the extensions of the models proposed for ChIP-chip data (Mo and Liang, 2010a,b). The package can be used to analyze the ChIP-seq data with or without controls.

This package processes ChIP-seq data in three steps.

- Build a signal profile for each chromosome. To create the signal profiles, sequence tags are aggregated into non-overlapping bins whose length can be decided by the user (e.g. 50 base pairs (bp)). The number of tags falling each bin is counted. The signal profiles are made of the bin-based tag counts and the corresponding genomic positions on the chromosomes. The bins and their tag counts are ordered along the chromosomes according to their genomic positions.

- Model the signal profiles. It is to model the tag counts on the chromosomes. The models assume that each bin is associated with a binary latent variable $X_i \in (1, -1)$, where i denotes the ID for the bin, and $X_i = 1$ denotes that the bin is an enriched bin, and -1 for a non-enriched bin. In the first stage, conditioning on the latent variable X_i, the bin-based tag counts are modeled by Poisson-Gamma distributions. If $X_i = -1$, the tag count for bin i is assumed to follow a Poisson distribution with parameter λ_0, where $\lambda_0 \sim \text{Gamma}(a_0, b_0)$. If $X_i = 1$, the tag count for bin i is assumed to follow a Poisson distribution with parameter λ_1, where $\lambda_1 \sim \text{Gamma}(a_1, b_1)$. $\text{Gamma}(a, b)$ denotes a gamma distribution with mean a/b and variance a/b^2. 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.

- Call enriched regions. The posterior probabilities for the bins in the enriched state ($X_i = 1$) are used for statistical inference. A bin with a high posterior probability in the enriched state will provide strong evidence that the bin is enriched. Enriched bins are then merged into enriched regions.

For more information, we refer the user to Mo and Liang (2010 a, b) because the manuscript (Mo, 2010c) is still under review. The major difference for modeling ChIP-chip and ChIP-seq data is at the first stage, where normal distributions are used for ChIP-chip data, and Poisson distributions are used for ChIP-seq data.

2 The NRSF ChIP-seq Data

Johnson et al.(2007) carried out genome-wide identification of the binding sites of human neuron-restrictive silencer factor (NRSF) in Jurkat T cells. We use the data of chromosomes 22 and Y to illustrate the proposed method. The NRSF sequence tags (25 bp) were generated by the Illumina/Solexa sequencing platform, and mapped to The human genome May 2004 (hg17). We only use the uniquely mapped tags (up to two mismatches) for the analysis. Note that iSeq package doesn't provide functions to read the raw data generated by the sequencers. However, it should not be difficult to prepare the data in the format as shown in the following.

3 Example — Analyze the NRSF Data

Firstly, let's load the library and check the data. There are three ChIP and control samples, respectively.

```
> library(iSeq)
> data(nrsf)
> names(nrsf)
```

```
[1] "chipFC1592" "chipFC1862" "chipFC2002" "mockFC1592" "mockFC1862"
[6] "mockFC2002"
```

3.1 Build the signal profiles using the mergetag function

The following two steps just merge the ChIP and control samples, respectively.

```
> chip = rbind(nrsf$chipFC1592, nrsf$chipFC1862, nrsf$chipFC2002)
> mock = rbind(nrsf$mockFC1592, nrsf$mockFC1862, nrsf$mockFC2002)
> print(chip[1:3, ])
```

```
      chr position strand
38  chr22 48039379      F
104 chr22 28163725      R
180 chr22 38814016      R
```

```
> print(mock[1:3, ])
```

```
     chr position strand
15 chr22 35510328      F
57 chr22 25441949      F
90 chr22 31761090      F
```

We suggest building the signal profiles using a 50 bp window. Use a small window size (e.g. 25 - 75 bp) in order to precisely infer the true binding sites. If the sequenced DNA fragments are around 200 bp, it is expected that an enriched region is made of 4 bins. However, if the window size is too small, it may lose power to detect enriched regions.

```
> tagct = mergetag(chip = chip, control = mock, winsize = 50)
> print(tagct[1:3, ])
```

```
    chr   gstart     gend adjct ipct1 ipct2 conct1 conct2
1 chr22 14433408 14433408     1     0     1      0      0
2 chr22 14436138 14436138     1     1     0      0      0
3 chr22 14436262 14436262     1     0     1      0      0
```

See the help file for function 'mergetag' for the meanings of the tagct columns.
The user can quickly get the 'enriched' regions without calculating statistical confidence using function 'peakreg'. For example, if we claim that a bin with adjusted tag count > 10 is an enriched bin, the 'enriched' regions can be obtained by using the following code. Let's use the chromosome 22 data as an example.

```
> tagct22 = tagct[tagct[, 1] == "chr22", ]
> reg0 = peakreg(chrpos = tagct22[, 1:3], count = (tagct22[, 5:6] -
+     tagct22[, 7:8]), pp = (tagct22[, 4] > 10), cutoff = 0, method = "ppcut",
+     maxgap = 300)
> print(dim(reg0))
```

```
[1] 68 11
```

```
> print(reg0[1:3, ])
```

```
    chr   gstart     gend rstart rend  peakpos cp meanpp ct1 ct2 ct12
1 chr22 15913767 15913863    670  671 15913813  1      1   8  21   29
2 chr22 15975306 15975356    851  851 15975331  0      1   8   4   12
3 chr22 15975753 15976055    858  863 15975957  1      1 301 314  615
```

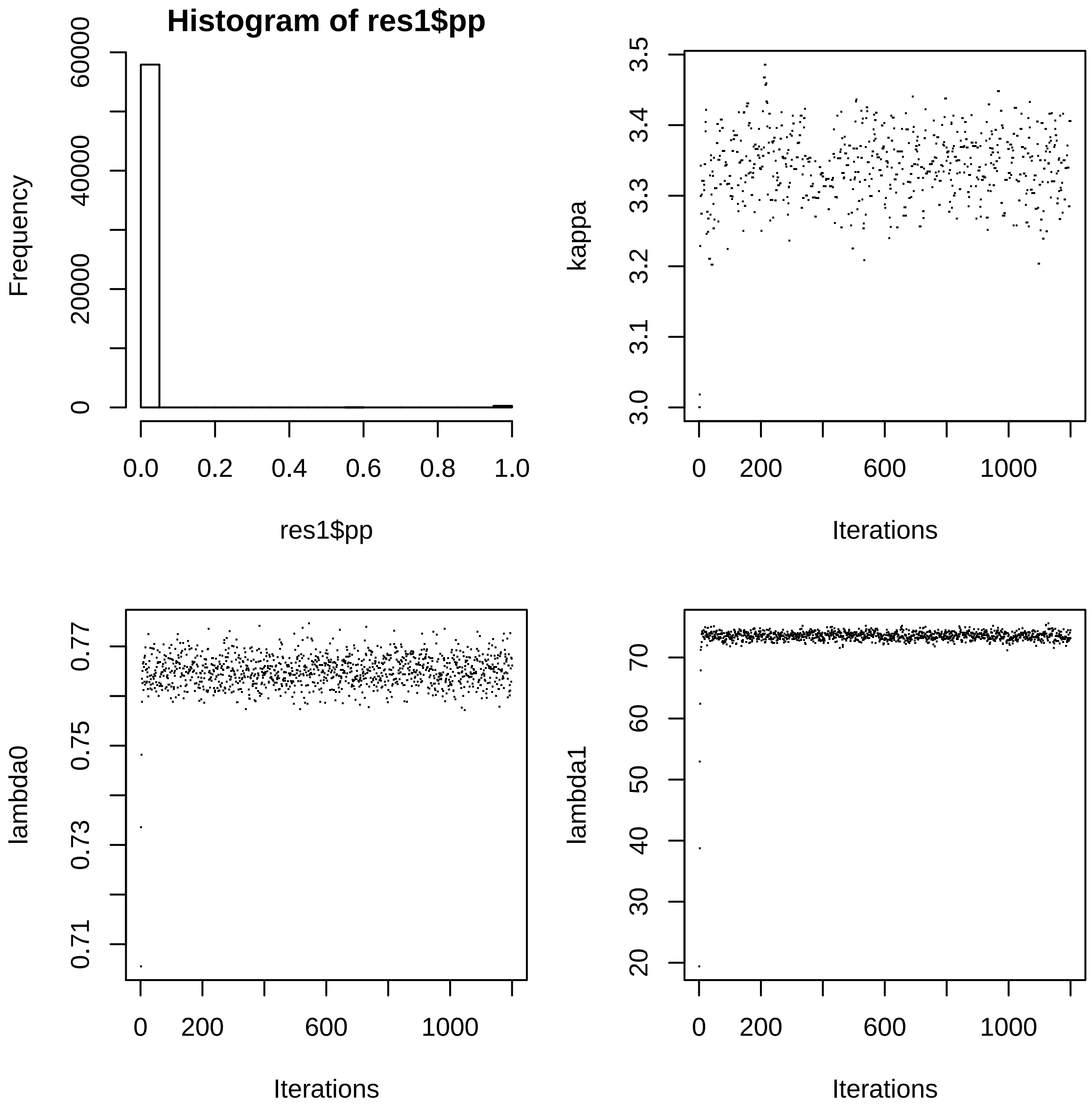
3.2 Model the signal profiles using the iSeq1 function

Function iSeq1 implements a fully Bayesian hidden Ising model in which the latent variable X is modeled by the standard 1D Ising model. The columns 1-4 of tagct are the signal profiles for modeling. Note that the column 4 is the adjusted tag counts of the ChIP samples. For one sample analysis, it is the total tag counts for the forward and reverse chains.

```
> set.seed(777)
> res1 = iSeq1(Y = tagct22[, 1:4], gap = 300, burnin = 200, sampling = 1000,
+     ctcut = 3, a0 = 1, b0 = 1, a1 = 5, b1 = 1, k0 = 3, mink = 0,
+     maxk = 10, normsd = 0.1, 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), the user should increase the number of iterations in the burn-in and/or sampling phases. If the chains do not mix well, the user can adjust the argument normsd to see how it affects the results.

```
> par(mfrow = c(2, 2), mar = c(4.1, 4.1, 2, 1))
> hist(res1$pp)
> plot(res1$kappa, pch = ".", xlab = "Iterations", ylab = "kappa")
> plot(res1$lambda0, pch = ".", xlab = "Iterations", ylab = "lambda0")
> plot(res1$lambda1, pch = ".", xlab = "Iterations", ylab = "lambda1")
```



From the trace plots, we see the chains converge quite fast.

3.3 Call the enriched regions detected by iSeq1 using the peakreg function

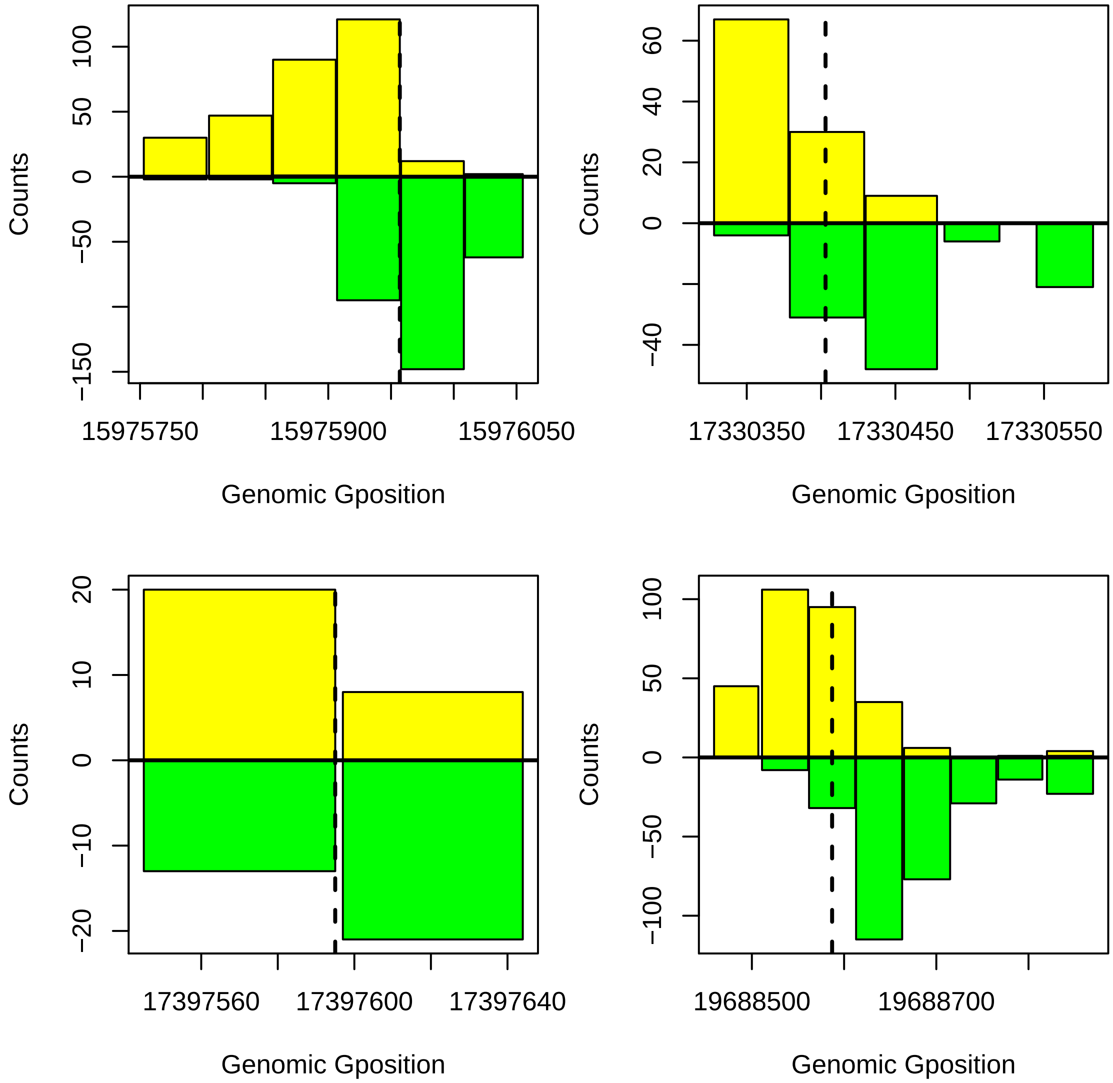
Call the enriched regions detected by iSeq1 using 0.5 posterior probability (pp) cutoff. Note the argument count is the net tag counts for the two sample analysis. The net tag counts are not truncated at zero, but this doesn't matter because it is just used for inferring the center of the enriched region, which is usually the true binding site. The user can also use the ChIP tag counts only. The results are little different. See the help file of peakreg for details.

```
> reg1 = peakreg(chrpos = tagct22[, 1:3], count = (tagct22[, 5:6] -
+     tagct22[, 7:8]), pp = res1$pp, cutoff = 0.5, method = "ppcut",
+     maxgap = 300)
> print(dim(reg1))

[1] 54 11

> print(reg1[1:3, ])

    chr   gstart     gend rstart rend  peakpos cp meanpp ct1 ct2 ct12
1 chr22 15975753 15976055    858  863 15975957  1    1.0 301 314  615
2 chr22 17330328 17330583   2724 2728 17330403  1    0.8 106 110  216
3 chr22 17397545 17397644   2855 2856 17397595  1    1.0  28  34   62
```

Note, the 5' positions of the sequence tags are used as the genomic positions for the NRSF data. To infer the actual binding sites, one may add 13 bp to the peak position (reg1$peakpos + 13) because the tags' length is 25 bp. If the middle positions of the sequence tags are used as the genomic positions, the user doesn't need to do the adjustment.

Plot some enriched regions. The dash lines indicate the region center, usually the true binding sites.

```
> par(mfrow = c(2, 2), mar = c(4.1, 4.1, 2, 1))
> for (i in 1:4) {
+     ID = (reg1[i, 4]):(reg1[i, 5])
+     plotreg(tagct22[ID, 2:3], tagct22[ID, 5:6], tagct22[ID, 7:8],
+         peak = reg1[i, 6])
+ }
```



Call the enriched regions using 0.05 FDR cutoff. The FDR is calculated using a direct posterior probability approach (Newton et al., 2004).

```
> reg2 = peakreg(tagct22[, 1:3], tagct22[, 5:6] - tagct22[, 7:8],
+     res1$pp, 0.05, method = "fdrcut", maxgap = 300)
> print(dim(reg2))
```

```
[1] 56 11
```

```
> print(reg2[1:3, ])
```

```
    chr   gstart     gend rstart rend  peakpos cp meanpp ct1 ct2 ct12
1 chr22 15975753 15976055    858  863 15975957  1    1.0 301 314  615
2 chr22 17330328 17330583   2724 2728 17330403  1    0.8 106 110  216
3 chr22 17397545 17397644   2855 2856 17397595  1    1.0  28  34   62
```

The columns 1-3 of the enriched regions (e.g. reg2[,1:3]) can be used to extract the sequences from the UCSC genome browser. Alternatively, one may create a BED file using the peak position of the enriched regions. For example,

```
> bed = data.frame(chr = reg2[, 1], gstart = reg2[, 6] - 100, gend = reg2[,
+     6] + 100)
```

3.4 Model the signal profiles using the iSeq2 function

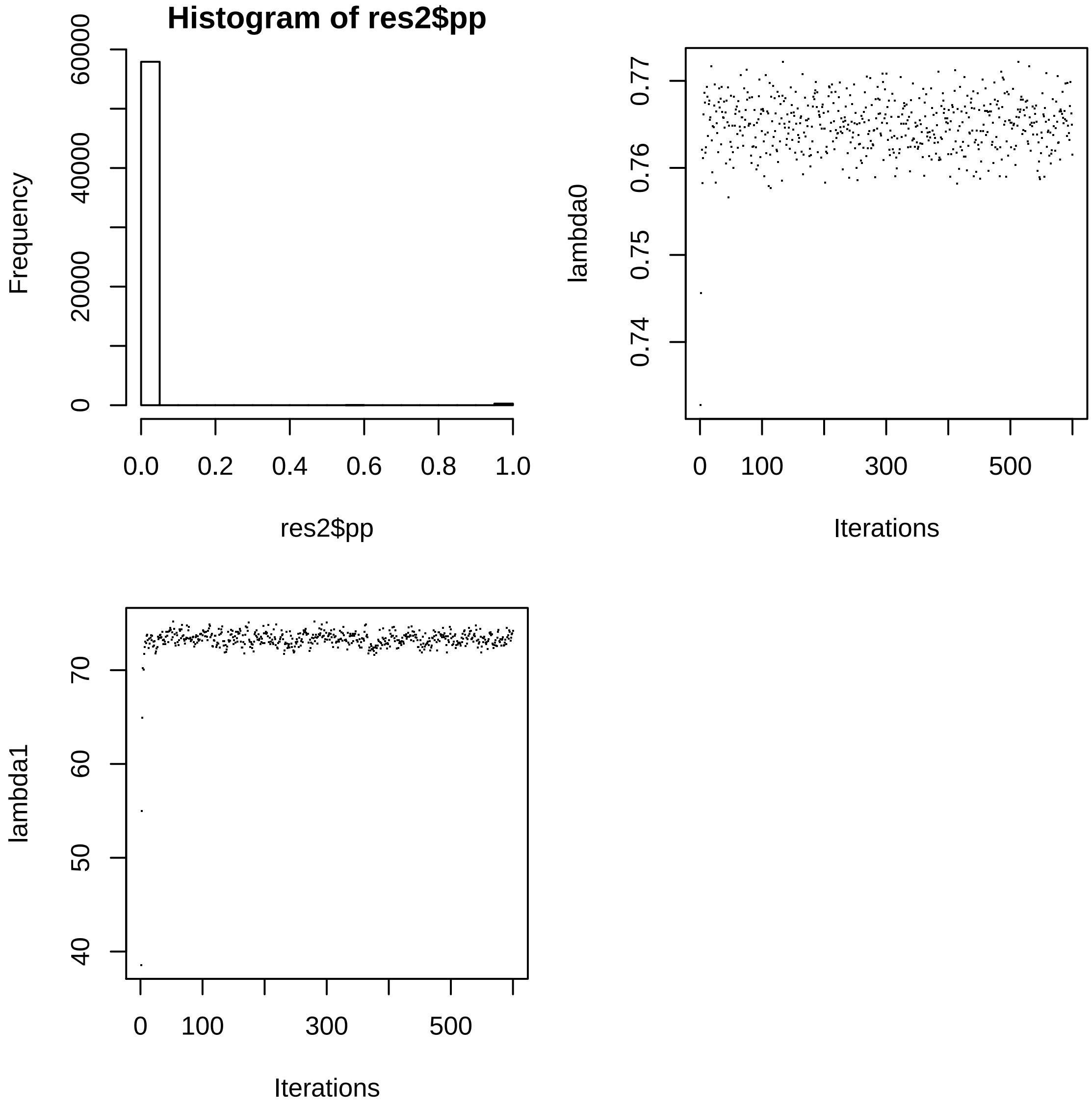
The latent variable X can be modeled through a high-order Ising model using function iSeq2. The interaction parameter k for the high-order (or the standard/first-order) Ising model is fixed and set by the user in iSeq2. To apply the second-order Ising model to ChIP-seq data, the user can let winsize = 2. If set winsize = 1, it will be the standard Ising model. To use a high-order Ising model, according to our experience, a balance between high sensitivity and low FDR can be achieved when winsize = 2. The critical value for the second-order Ising model is about 1.0. In general, increasing the value of k will lead to less enriched regions, which amounts to setting a stringent criterion for detecting enriched regions.

Model the NRSF data using the second-order Ising model.

```
> res2 = iSeq2(Y = tagct22[, 1:4], gap = 300, burnin = 100, sampling = 500,
+     winsize = 2, ctcut = 5, a0 = 1, b0 = 1, a1 = 5, b1 = 1, k = 1,
+     verbose = FALSE)
```

Plot the model parameters to see whether they converge. If the chains do not mix well, one can adjust the parameter **k** and/or normsd to see how it affects the results.

```
> par(mfrow = c(2, 2), mar = c(4.1, 4.1, 2, 1))
> hist(res2$pp)
> plot(res2$lambda0, pch = ".", xlab = "Iterations", ylab = "lambda0")
> plot(res2$lambda1, pch = ".", xlab = "Iterations", ylab = "lambda1")
```



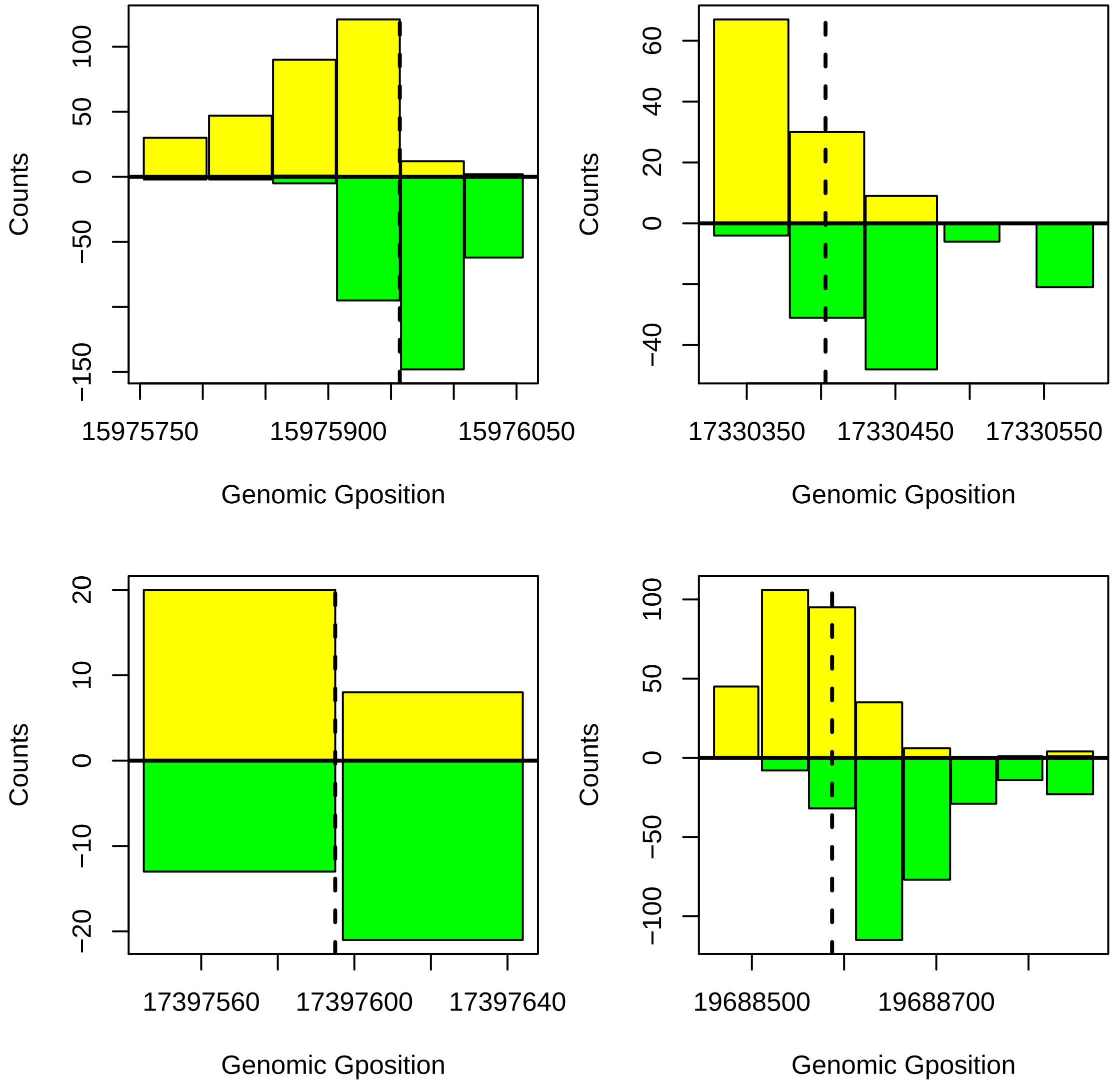
3.5 Call the enriched regions detected by iSeq2 using the peakreg function

```
> reg2 = peakreg(tagct22[, 1:3], tagct22[, 5:6], res2$pp, 0.5,
+     method = "ppcut", maxgap = 300)
> print(dim(reg2))

[1] 56 11

> print(reg2[1:3, ])

    chr   gstart     gend rstart rend  peakpos cp meanpp ct1 ct2 ct12
1 chr22 15975753 15976055    858  863 15975957  1    1.0 302 314  616
```

```
2 chr22 17330328 17330583   2724 2728 17330403  1    0.8 106 110  216
3 chr22 17397545 17397644   2855 2856 17397595  1    1.0  28  34   62
```

Plot some enriched regions detected by iSeq2.

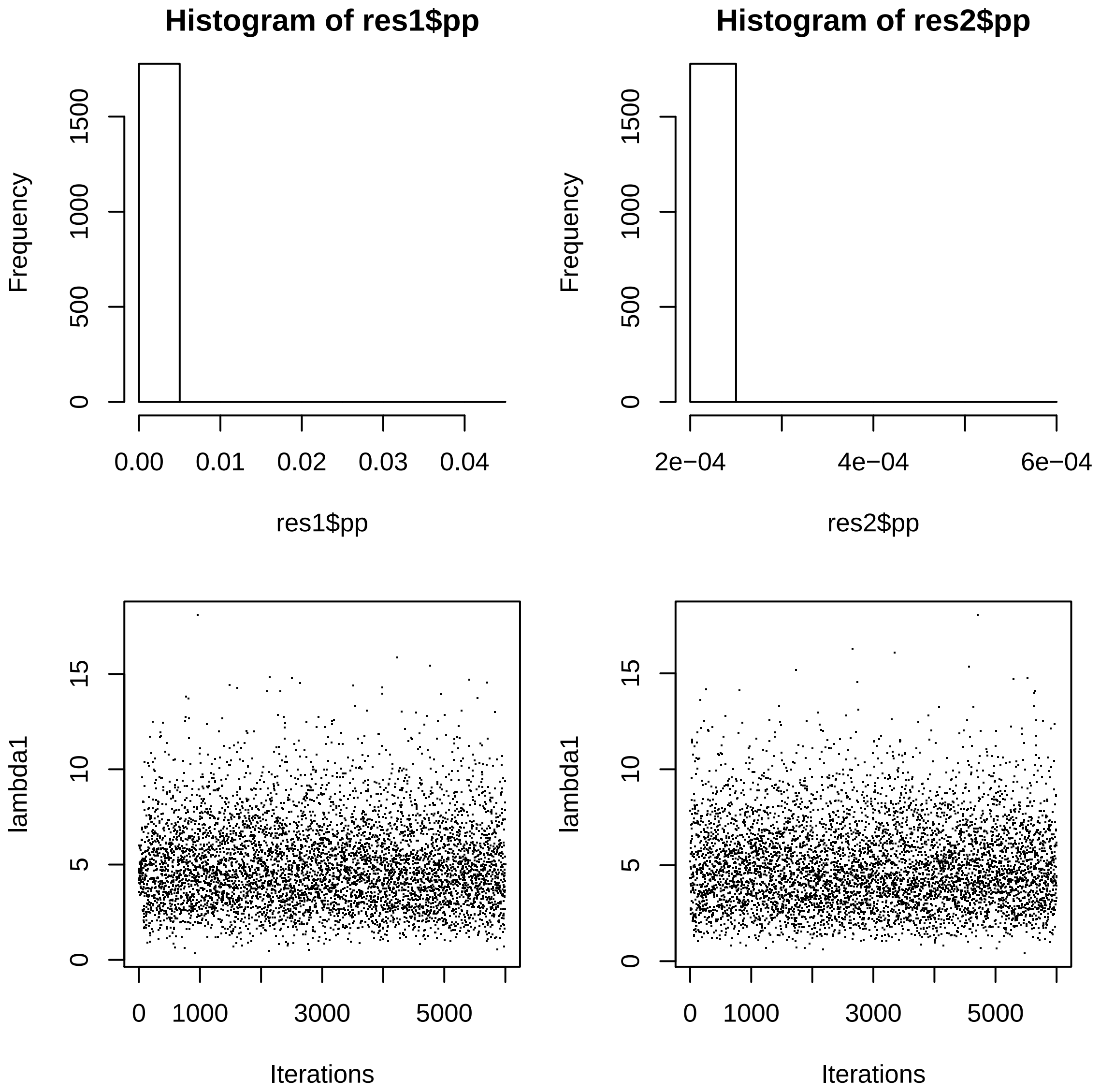
```
> par(mfrow = c(2, 2), mar = c(4.1, 4.1, 2, 1))
> for (i in 1:4) {
+     ID = (reg2[i, 4]):(reg2[i, 5])
+     plotreg(tagct22[ID, 2:3], tagct22[ID, 5:6], tagct22[ID, 7:8],
+         peak = reg2[i, 6])
+ }
```



4 Tips

Finally, let's analyze the chromosome Y data. Intuitively, it seems that there is no binding region on chromosome Y.

```
> tagctY = tagct[tagct[, 1] == "chrY", ]
> print(table(tagctY[, 4]))

   0    1    2    3    4    5    6    7    9
 434 1106  149   61   17    6    5    1    1

> res1 = iSeq1(Y = tagctY[, 1:4], gap = 300, burnin = 1000, sampling = 5000,
+     ctcut = 3, a0 = 1, b0 = 1, a1 = 5, b1 = 1, k0 = 3, mink = 0,
+     maxk = 10, normsd = 0.5, verbose = FALSE)

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

> res2 = iSeq2(Y = tagctY[, 1:4], gap = 300, burnin = 1000, sampling = 5000,
+     winsize = 2, ctcut = 3, a0 = 1, b0 = 1, a1 = 5, b1 = 1, k = 3,
+     verbose = FALSE)

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

> par(mfcol = c(2, 2), mar = c(4.1, 4.1, 2, 1))
> hist(res1$pp)
> plot(res1$lambda1, pch = ".", xlab = "Iterations", ylab = "lambda1")
> hist(res2$pp)
> plot(res2$lambda1, pch = ".", xlab = "Iterations", ylab = "lambda1")
```



In this case, all the bins are only in the non-enriched state and the posterior probabilities are zero or close to zero. In some cases, if the λ_1 is small (e.g. $\approx a_1/b_1$, the mean value of the gamma prior), it suggests that there is no enriched region on that chromosome. The user may not claim any enriched regions found on that chromosome, even if some posterior probabilities are high. For the studies of transcription factor binding sites, if the posterior probabilities are not dichotomized or not dominated by 0, it suggests that the Ising model is not in the super-paramagnetic phase. Only the super-paramagnetic phase reflects the binding events on the chromosomes. Therefore, if the user use iSeq2, the user should increase the value of k to let the phase transition occur so that the Ising model reaches the super-paramagnetic phase.

The iSeq methods take into account the spatial dependency, the global and local distributions of the sequence tags. If the signal profiles have very sparse signals and some bins have very large (adjusted) tag counts (e.g., the NRSF data), the estimated λ_1 will be relatively large, which makes the bins with relatively small counts (e.g., tens) have low posterior probabilities. In this case, the user can truncate the (adjusted) tag counts at certain value (e.g., if count $>$ 100, set count = 100) to increase the power to detect the regions with small tag counts. For the NRSF data, if the adjusted tag counts are truncated at 100, more enriched regions can be detected.

5 Parallel Computation

To speed up the analysis, the user can do parallel computation easily. The user needs to install the snow and snowfall packages. The following is an example.

```
library(snow)
library(snowfall)
dataList = list(chr22=tagct22,chrY=tagctY)
sfInit(parallel=TRUE,cpus=2,type=”SOCK”)
res=sfLapply(dataList,iSeq1,gap=300,burnin=100,sampling=200,ctcut=3,a0=1,b0=1,a1=1,b1=1,
k0=3,mink=0,maxk=10,normsd=0.1,verbose=FALSE)
```

6 Citing iSeq

The iSeq1 method is described in “Mo, Q. (2010). A fully Bayesian hidden Ising model for ChIP-seq data analysis (Submitted).”. The other functions are described in “Mo, Q. (2010). iSeq - A flexible and powerful R/Bioconductor package for analyzing ChIP-seq data (Submitted).“

```
> sessionInfo()

R version 2.12.0 (2010-10-15)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=C              LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

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

other attached packages:
[1] iSeq_1.0.0
```

```
loaded via a namespace (and not attached):
[1] tools_2.12.0
```

References

Mo, Q., Liang, F. (2010a). Bayesian Modeling of ChIP-chip data through a high-order Ising Model. *Biometrics*, 2010 Jan 29 [Epub ahead of print]. DOI: 10.1111/j.1541-0420.2009.01379.x

Mo, Q., Liang, F. (2010b). A hidden Ising model for ChIP-chip data analysis. *Bioinformatics* **26(6)**, 777-783. doi:10.1093/bioinformatics/btq032

Mo, Q. (2010). A fully Bayesian hidden Ising model for ChIP-seq data analysis. (Submitted).

Mo, Q. (2010). iSeq - A flexible and powerful R/Bioconductor package for analyzing ChIP-seq data.(Submitted).

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

Johnson, D.S., Mortazavi, A., Myers, R.M., Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science* **316**, 1497-1502.