Supplementary Materials for
"A Bayesian Mixture Model for Chromatin Interaction Data"

Liang Niu, Shili Lin

S1. Details of the MCMC algorithms (for MC_DIST and NIP)

We use a MCMC algorithm to generate posterior samples for the MC_DIST model. The MCMC algorithm involves generating random samples from the full conditional distributions of each parameter of MC_DIST. Since the full conditional distributions of λ_{0i}, λ_{1i}, r_0, b and d are not of a known form, we utilize a Metropolis-Hastings algorithm using either a log-normal distribution (for λ_{0i}, λ_{1i}, r_0 and b) or a uniform distribution (for d) as the proposal distribution. Specifically, for each of the parameters λ_{0i}, λ_{1i}, r_0 and b, the new candidate value in each iteration is generated by a log-normal distribution $\ln N(x,\gamma^2)$, where x is the current value and γ is a tuning parameter; for the parameter d, the new candidate value in each iteration is generated by a uniform distribution $U(lb,ub)$, where $lb = max(0,x-\delta)$ and $ub = min(D,x+\delta)$ (remember that $d \sim U(0,D)$, where D is a constant and was set to be $10,000$ in the simulation study), where x is the current value and δ is a tuning parameter. For the simulation study, the chosen tuning parameters of the proposal distributions (γ or δ) and the achieved acceptance rates are shown in Table S1. In summary, all acceptance rates are reasonable.

Table S1: Metropolis-Hastings algorithm tuning parameters and achieved acceptance rates for MC_DIST in the simulation study.

parameter	tuning parameter	acceptance rate (or its range)
λ_{0i}	0.5	(0.45, 0.60)
λ_{1i}	0.5	(0.32, 0.78)
r_0	0.01	0.45
b	0.06	0.34
d	50	0.64

We also use a MCMC algorithm to generate posterior samples for the NIP model. The MCMC algorithm involves generating random samples from the full conditional distributions of each parameter of NIP. Since the full conditional distributions of λ_{0i}, λ_{1i}, r_0 and b are not of a known form, we utilize Metropolis-Hastings algorithms using log-normal distributions as the proposal distributions for those parameters. For the simulation study, the chosen tuning parameters of the proposal log-normal distributions and the achieved acceptance rates are shown in Table S2. Again, all acceptance rates are reasonable.

Table S2: Metropolis-Hastings algorithm tuning parameters and achieved acceptance rates for NIP in the simulation study.

parameter	tuning parameter	acceptance rate (or its range)
λ_{0i}	0.5	(0.52, 0.59)
λ_{1i}	0.5	(0.32, 0.76)
r_0	0.006	0.60
b	0.052	0.38

S2. Convergence diagnostics of the MCMC algorithms (for MC_DIST and NIP) used in the simulation study

To check the convergence of the MCMC algorithm used for MC_DIST in the simulation study, we first checked the trace plots for the model parameters. The trace plots (after a thinning of 100) are shown in Figure S1. Since that w_{1i}, λ_{0i} and λ_{1i} are pair specific, i.e., depend on i, we randomly selected one pair and checked the trace plots of w_{1i}, λ_{0i} and λ_{1i} for the selected pair, The trace plots in the figure indicate that the MCMC algorithm converges well.

We further checked the MCMC convergence using the Gelman and Rubins convergence diagnostic for each parameter. Such diagnostics provide a potential scale reduction factor for each parameter. If a potential scale reduction factor is high (greater than 1.1 or 1.2), then we need to run the chain longer. We used the R package *coda* to obtain the estimates of the above potential scale reduction factors and the estimates are shown in Table S3. They are all less than 1.1.

We also checked the MCMC convergence using the Raftery and Lewis diagnostic for each parameter. Such diagnostic is designed to determine the number of iterations and burn-in needed to ensure the convergence of a parameter. We used the R package *coda* to perform the Raftery and Lewis diagnostics and the results are summarized in Table S4. All burn-in values are less than 400,000 (the burn-in value used in the MCMC algorithm). Also, all numbers of iterations needed to ensure the convergence (the “total” in the table) are less than 2,000,000 (the number

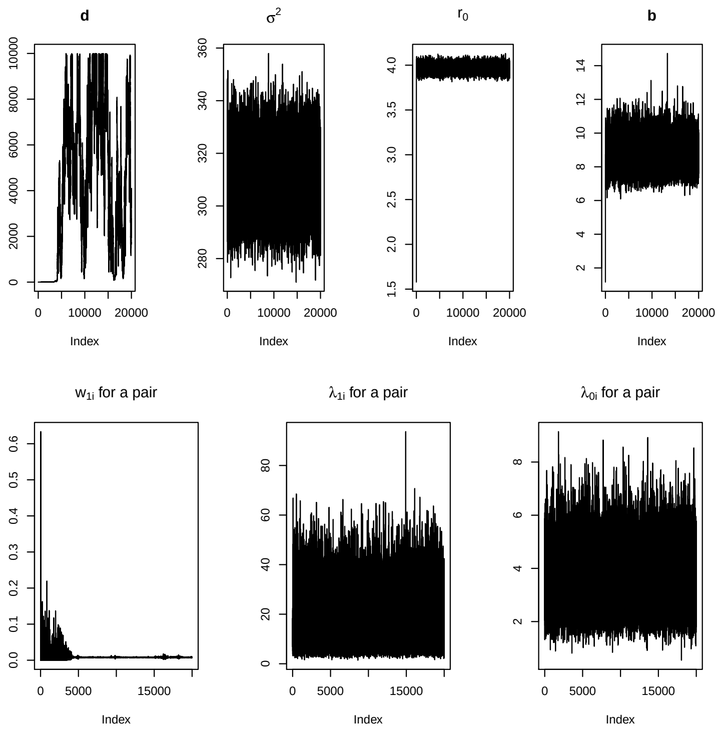

Figure S1: Trace plots (after a thinning of 100) of the parameters in the MCMC algorithm used in the simulation study for MC_DIST.

Table S3: Gelman and Rubins convergence diagnostics for the MCMC algorithm used in the simulation study for MC_DIST.

parameter	potential scale reduction factor estimate
r_0	1
b	1
d	1.04
σ^2	1
w_{1i}	1.02
λ_{0i}	1
λ_{1i}	1

of iterations used in the MCMC algorithm) except for d and w_{1i}. Although the numbers of iterations suggested exceed 2,000,000 for d and w_{1i}, this suggestion should be viewed in balance with the other diagnostic results. Further, since the Raftery and Lewis diagnostics are known to be conservative, we opted not to increase the

Table S4: Raftery and Lewis convergence diagnostics for the MCMC algorithm used in the simulation study for MC_DIST.

parameter	burn-in	total
r_0	200	381100
b	400	470300
d	169900	94788200
σ^2	200	381800
w_{1i}	11200	12643400
λ_{0i}	100	369500
λ_{1i}	100	375500

run length, especially since no problem was detected based on trace plots and the Gelman and Rubin diagnostics.

For NIP model, we also check the convergence of MCMC algorithm by the trace plots (after a thinning of 100), the Gelman and Rubins convergence diagnostic and the Raftery and Lewis diagnostic for each parameter. The trace plots are shown in Figure S2. Note that in this figure, there is no parameter d as no such a parameter is in NIP, and that the parameter w_1 is not pair specific. Again, we randomly selected a pair and checked the trace plots of λ_{0i} and λ_{1i} for the selected pair. The trace plots indicate that the MCMC algorithm converges well.

The Gelman and Rubins convergence diagnostics for parameters in NIP are summarized in Table S5. All estimates of the potential scale reduction factors are 1, which indicate that the MCMC algorithm converges well.

Table S5: Gelman and Rubins convergence diagnostics for the MCMC algorithm used in the simulation study for NIP.

parameter	potential scale reduction factor estimate
r_0	1
b	1
σ^2	1
w_1	1
λ_{0i}	1
λ_{1i}	1

The Gelman and Rubins convergence diagnostics for parameters in NIP are summarized in Table S6. We see that all burn-in values and numbers of iterations needed for convergence are less than 40,000 (the burn-in value used in the MCMC algorithm) and 2,000,000 (the number of iterations used in the MCMC algorithm).

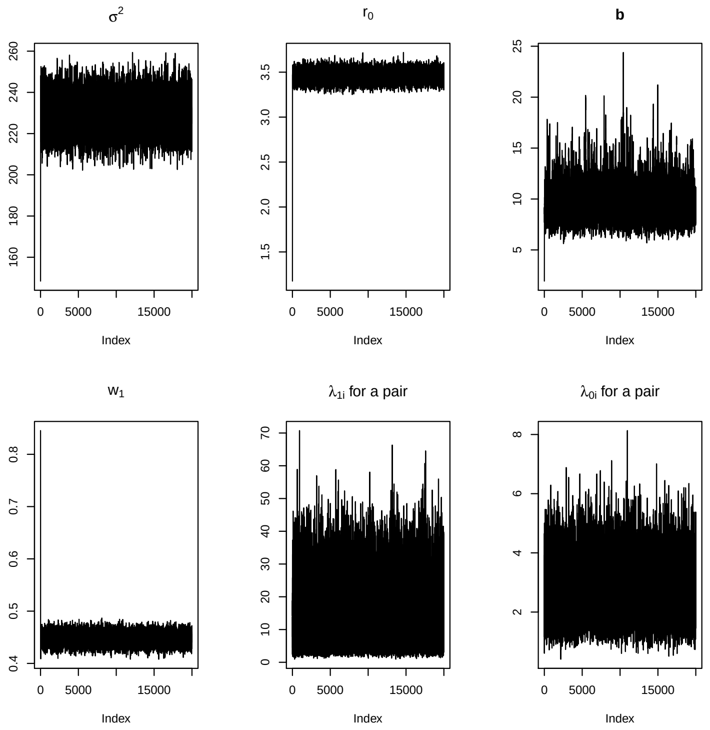

Figure S2: Trace plots (after a thinning of 100) of the parameters in the MCMC algorithm used in the simulation study for NIP.

Table S6: Raftery and Lewis convergence diagnostics for the MCMC algorithm used in the simulation study for NIP.

parameter	burn-in	total
r_0	600	930200
b	800	1060200
σ^2	200	383800
w_1	200	399400
λ_{0i}	100	375500
λ_{1i}	100	391300

S3. Analysis on posterior distributions

To investigate the properties of MC_DIST model, we analyzed the marginal posterior distribution of each parameter of the MC_DIST model in the simulation study. Such distributions were estimated from the posterior samples obtained from the

MCMC algorithm for MC_DIST, using the kernel density estimation.

The estimated marginal posterior distributions of d, σ^2, r_0 and b are shown in Figure S3. The marginal posterior distribution of r_0 is concentrated over the narrow interval $(3.8, 4.1)$; the marginal posterior distribution of b is concentrated over the interval $(7, 11)$. To see whether these two distributions are reasonable, we pooled the posterior samples for λ_{0i} (for all pairs) obtained from the MCMC algorithm, and fitted a gamma distribution to the pooled samples. The kernel density estimation for the pooled samples and the fitted gamma distribution $\Gamma(8.63, 2.18)$ are shown as the black curve and the red dashed curve respectively in Figure S4. From the figure, we see that the gamma distribution fits perfectly to the samples and the shape parameter (b) 8.63 falls in the interval $(7, 11)$, and the rate parameter ($\frac{b}{r_0}$) 2.18 is equal to $8.63/3.96$ where 3.96 falls in the interval $(3.8, 4.1)$. The above observation show that the model assumption that λ_{0i} follows a gamma distribution is consistent with the analysis results and that the marginal posterior distributions of r_0 and b are reasonable. Furthermore, The observation that the marginal posterior distribution of r_0 is concentrated over $(3.8, 4.1)$ is supported by the data analysis

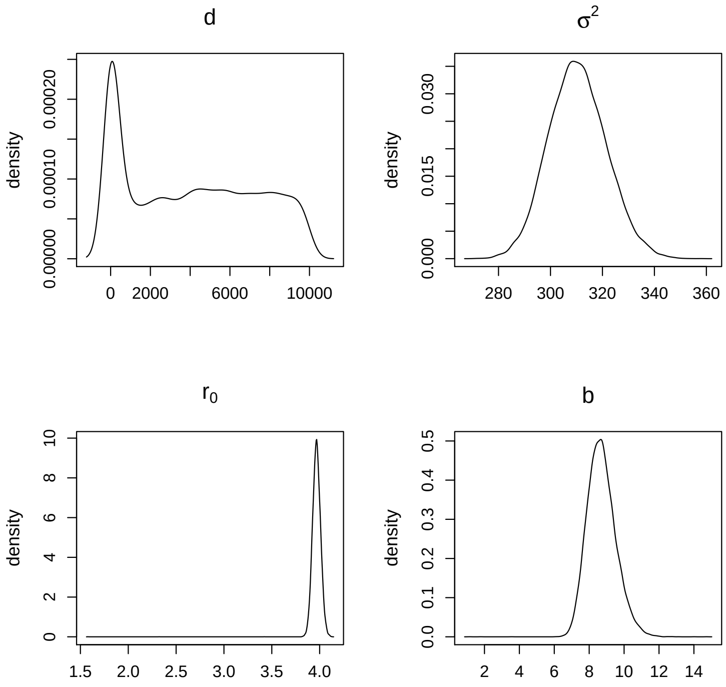

Figure S3: Estimated posterior distributions of d, σ^2, r_0 and b, in the simulation study.

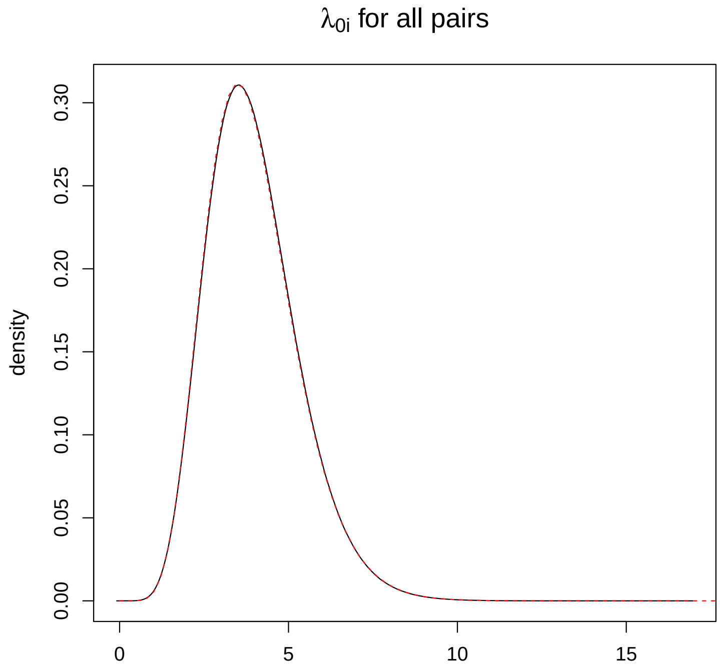

Figure S4: Estimated posterior distribution of λ_{0i} for all pairs in the simulation study. The black curve represents the estimated kernel density and the red dashed curve represents the density function of the fitted gamma distribution $\Gamma(8.63, 2.18)$.

result that the mean count for those pairs that were classified by the MC_DIST as false pairs is 3.90. Thus, the model assumption $\lambda_{0i} \sim \Gamma(b, \frac{b}{r_0})$ is reasonable for the simulated data. The marginal posterior distribution of σ^2 is a normal-like distribution with mean at about 315. To see whether such a distribution is consistent with the data, we calculated different percentiles of a folded normal distribution $N(0,351) \cdot I(\cdot > 0)$, and found that the theoretical percentiles are consistent with the data analysis result. In particular, the 10th, 20th, 30th, 40th theoretical percentiles are 2.23, 4.50, 6.84 and 9.31 respectively; and the 10th, 20th, 30th, 40th empirical percentiles are 2.91, 5.54, 7.99 and 10.41, respectively, where the empirical percentiles were calculated based on the posterior samples of $\{\lambda_{1i} - \lambda_{0i} | 1 \leq i \leq n\}$. The above observation shows that the marginal posterior distributions of σ^2 is reasonable. The marginal posterior distribution of d is more spread out, this is because that d is a hyperparameter in very top level of the hierarchy (see Figure 1 in the main text) and thus the marginal posterior distribution is not affected much by the data.

We also randomly selected three pairs ($i = 17$, 18 and 70) and estimated the marginal posterior distributions of w_{1i}, λ_{1i} and λ_{0i} by the kernel density estimation using the posterior samples obtained from the MCMC algorithm. The count of the three selected pairs are 3, 5, 13 and the rmcd's of the three selected pairs are 0.05, 0.02 and 33.00, respectively. The three rmcd's are about 20th, 40th and 96th percentiles of all the rmcd's, respectively. The first two pairs are false and the last one is a true pair and the MC_DIST classified all pairs correctly. The estimated distributions are shown in Figure S5. We see that the posterior distributions are consistent with the data. In particular, the marginal posterior distribution of λ_{0i} for each pair is concentrated around the corresponding count of the pair.

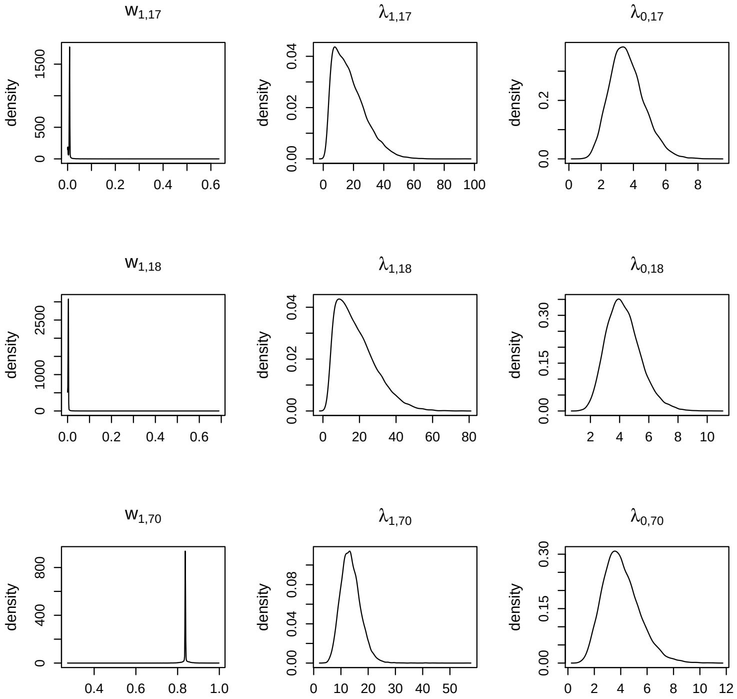

Figure S5: Estimated posterior distributions of w_{1i}, λ_{1i} and λ_{0i}, for three randomly selected pairs in the simulation study.

Base on the above analysis on the marginal posterior distributions, we see that the marginal posterior distributions are all reasonable and the MC_DIST model assumptions are consistent with the simulated data.

S4. Summary of 100 additional simulations

To investigate how reproducible the results from MC_DIST are across multiple simulations, we replicated the simulation in section 3.1 100 times to obtain 100 simulated data sets. The 184 loci used in each simulation are simulation-specific. That is, in each simulation we randomly selected 46 ERα binding sites in MCF7 cell line (two on each human chromosome), 46 gene transcription start sites (two on each human chromosome), and 92 non-specific loci (not a ERα binding site or a gene transcription start site, four on each human chromosome). The MC_DIST results on those data sets are summarized in Figure S6, in which we plotted the box plot of the type I error rates and the box plot of power for those 100 simulation studies. We can see that the type I error rates and the powers are consistent across the 100 simulation studies.

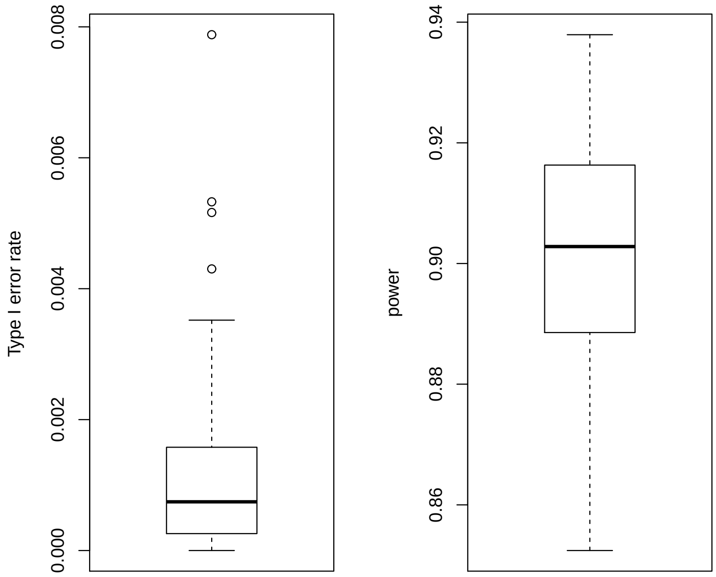

Figure S6: Summary of the MC_DIST results on 100 simulated data sets.

S5. Investigation of variants of MC_DIST

To investigate the sensitivity and stability of MC_DIST, we modified MC_DIST by assuming an alternative prior distribution on λ_{1i}, i.e., $N(\lambda_{0i} + m, \sigma^2) \cdot I(\lambda_{1i} > \lambda_{0i})$ with $m > 0$ pre-selected, and applied such a MC_DIST variant on the simulated data set. To choose different m values, we ran MC_DIST on the simulated data set and calculated the 10th, 20th, 30th and 40th percentiles of the posterior samples of $\{\lambda_{1i} - \lambda_{0i} | 1 \leq i \leq n\}$, and used these four values as the m values. The results of those MC_DIST variants, together with the result for MC_DIST, are summarized in Table S7. We can see that the MC_DIST variants performed almost the same as MC_DIST.

Table S7: MC_DIST variants.

MC_DIST variant	Type I error	Power
MC_DIST	0.0003	0.876
10th percentile	0.0003	0.882
20th percentile	0.0003	0.882
30th percentile	0	0.878
40th percentile	0	0.872