Parallel Tempering with Equi-Energy Moves

Meili Baragatti, Agnès Grimaud, Denys Pommeret

Parallel Tempering with Equi-Energy Moves

Meïli Baragatti[1,2,*], Agnès Grimaud[2], Denys Pommeret[2]

[1] *Ipsogen SA, Luminy Biotech Entreprises, Case 923, Campus de Luminy, 13288 Marseille Cedex 9, France.*

[2] *Institut de Mathématiques de Luminy (IML), CNRS Marseille, case 907, Campus de Luminy, 13288 Marseille Cedex 9, France.*

[*] *baragatt@iml.univ-mrs.fr, baragattimeili@hotmail.com.*

PREPRINT 06-06-2011

Abstract

The Equi-Energy Sampler (EES) introduced by Kou et al. [2006] is based on a population of chains which are updated by local moves and global moves, also called equi-energy jumps. The state space is partitioned into energy rings, and the current state of a chain can jump to a past state of an adjacent chain that has energy level close to its level. This algorithm has been developed to facilitate global moves between different chains, resulting in a good exploration of the state space by the target chain. This method seems to be more efficient than the classical Parallel Tempering (PT) algorithm. However it is difficult to use in combination with a Gibbs sampler and it necessitates increased storage. In this paper we propose an adaptation of this EES that combines PT with the principle of swapping between chains with same levels of energy. This adaptation, that we shall call Parallel Tempering with Equi-Energy Moves (PTEEM), keeps the original idea of the EES method while ensuring good theoretical properties, and practical implementation even if combined with a Gibbs sampler. Performances of the PTEEM algorithm are compared with those of the EES and of the standard PT algorithms in the context of mixture models, and in a problem of identification of gene regulatory binding motifs.

Keywords: equi-energy sampler, parallel tempering, population-based Monte Carlo Markov Chains, algorithm convergence, mixture models, binding sites for transcription factors.

1 Introduction

A common problem in Bayesian statistics is that of generating random variables from a target density π. Many solutions have been proposed in the last two decades, deriving essentially from the Monte Carlo Markov Chains (MCMC) approach introduced by Metropolis et al. [1953] and Hastings [1970]. In classical MCMC methods, a Markov process is built to sample the target probability distribution. But in practice, the Markov process can be easily trapped into a local mode from where it cannot escape in reasonable time (see for instance Liang and Wong [2001]). Many techniques have been proposed to address this waiting time problem, including among others Parallel Tempering (PT) (see Geyer [1991] or Geyer and Thompson [1995]), and more recently Equi-Energy Sampler (EES) (Kou et al. [2006]).

In the PT algorithm, N temperatures are introduced, and N chains are run in parallel, with target distributions being tempered distributions of the target π. Note that the first chain

targets π. Since the tempered distributions becomes flatter as the temperature increases, the chains at high temperatures can move easily between modes. Each iteration of the PT algorithm is decomposed into two types of moves: local moves via classical MCMC algorithms to update the different chains, and global moves allowing swaps between two chains. The use of these swaps enables new modes to be propagated through the different chains, thereby improving mixing. The first chain associated with the target distribution will then be able to escape from local modes. Some improvements of PT have been proposed, like swaps with delayed rejection (see Green and Mira [2001]) which permit to propose a new swap when the first one is not accepted, or like Evolutionary Monte Carlo (Liang and Wong [2001]). However this PT algorithm does not retain information of where chains have been and it does not choose one of the best swaps. This is what is done by the EES proposed by Kou et al. [2006]. Note that a method described in Atchadé and Liu [2006], called multicanonical sampling, is in the same spirit than the EES.

In the EES algorithm, the target density is rewritten in terms of energy function. K temperatures and energy levels are introduced. Then a population of K distributions is considered, each one being a tempered distribution of π truncated by an energy level. This algorithm is mainly based on a new type of move called the equi-energy jump, that aims to explore the state space by moving directly between states with similar energy. The goal is still to improve mixing of the chains. However, to perform these moves, the sampler uses past states of the different chains. All these past states should then be kept in memory. A substantial advantage of this algorithm is that it seems to be very efficient compared to classical MCMC methods and to PT (see Kou et al. [2006]). But an associated drawbacks is the cost of increased storage, all the past being taken into account in equi-energy jumps. In addition some difficulties are encountered to combine EES with a Gibbs sampler. The problem is to sample from the tempered distributions truncated by energy levels. Some algorithms could be used to sample from it, like accept-reject or Approximate Bayesian Computation algorithms, but the computational cost would then be too high in practice. From a theoretical point of view, the EES is not based on a Markov chain, and its theoretical analysis is relatively difficult. Several authors studied its convergence under various assumptions. The proof of the convergence has been discussed in Andrieu et al. [2007, 2008], and Atchadé et al. [2010] showed that the asymptotic variance of the EES can be substantially different than that suggested by Kou et al. [2006]. Hua and Kou [2010] completed the proof of the convergence of the EES in the case of a countable state space, and recently more general convergence results has been established by Fort et al. [2010].

In this paper we develop an adaptation of the PT and EES algorithms, called the Parallel Tempering with Equi-Energy Moves (PTEEM) algorithm. An equi-energy exchange move is proposed, based on the energies of current states of the chains, and not on past states. Compared to PT algorithm, only moves between chains whose states are close in energy are proposed. This focuses computational effort on moves which are likely to be accepted, and hence which allow jumps between modes. This PTEEM algorithm can be easily combined with a Gibbs sampler, and its convergence is ensured. Furthermore, it does not need a large storage. The possible loss or gain of this algorithm compared to EES and PT are evaluated through simulations and real data. The advantage of using a Gibbs sampling is highlighted on a problem of motif sampling already studied by Kou et al. [2006].

The paper is organized as follows: In Section 2, PT and EES algorithms are briefly recalled. In Section 3 the PTEEM algorithm is presented. In Sections 4 and 5, performances of the

PTEEM algorithm are compared with those of the EES and of the standard PT algorithms in the context of mixture models, through simulations and real data. In Section 6, PTEEM and EES algorithms are compared in a challenging problem of identification of gene regulatory binding motifs. Section 7 presents concluding remarks.

2 Background on PT and EES algorithms

2.1 PT algorithm

In case of complex or high dimensional problems whose densities of interest contain several modes, classical MCMC methods (like Metropolis-Hastings algorithm or Gibbs sampler for instance) are often trapped into local modes from where they cannot escape in reasonable time. To avoid this problem, the principle of PT is to choose N temperatures $T_1 = 1 < T_2 < \cdots < T_N$, and to run in parallel N associated MCMC chains having different stationary distributions, $\pi_1, \cdots, \pi_N$, where

$$\pi_i \quad \propto \quad \pi^{1/T_i}.$$

The higher the temperature is, the easier the exploration of the state space is for the associated chain. Each iteration of the PT algorithm is decomposed into local and global moves. During local moves, each chain is updated independently of others. In particular, the ith chain is updated using a classical MCMC algorithm with stationary distribution π_i. For a global move, two chains i and j are randomly chosen and a swap of their current states is proposed, and accepted with the following Metropolis-Hastngs ratio:

$$\min\left\{1, \frac{\pi_i(x_j)\pi_j(x_i)}{\pi_i(x_i)\pi_j(x_j)}\right\},$$

where x_i stands for the current state of the ith chain.

2.2 EES algorithm

To use the EES algorithm introduced by Kou et al. [2006], a sequence of K temperatures and $K+1$ energy levels should be chosen: $H_1 < H_2 < \cdots < H_{K+1} = \infty$, where $H_1 \leq \min(h(x))$, and $T_1 = 1 < T_2 < \cdots < T_K$. A population of K distributions with the following densities is considered:

$$\tilde{\pi}_i(x) \quad \propto \quad \exp\{-h_i(x)\}, \qquad \text{where} \qquad h_i(x) = \frac{\max\{h(x), H_i\}}{T_i}.$$

The main difference with the PT algorithm being the energy truncation. This energy truncation is used to flatten the distributions for easier exploration. This method uses energy rings for each chain, an energy ring containing past states of the chain of similar energy levels. The algorithm begins by sampling the Kth chain from a Metropolis-Hastings kernel with stationary distribution $\tilde{\pi}_K$. Once convergence is reached, generated samples are stored in the energy rings of this Kth chain, and the next chain targeting $\tilde{\pi}_{K-1}$ starts. This $(K-1)$th chain will be updated by either (with probability p_{ee}) using a Metropolis-Hastings kernel with stationary distribution $\tilde{\pi}_{K-1}$, or

by proposing to replace the current state of the chain by a past state of the previous chain of similar energy level. This move corresponds to the equi-energy jump, and is based on the energy rings of the previous chain. Once convergence is reached, generated samples are stored in the energy rings of this $(K-1)$th chain, and the next chain targeting $\tilde{\pi}_{K-2}$ starts. The EES algorithm successively steps down the energy and temperature ladder until the target distribution $\pi_1 = \pi$ is reached. Each chain i, with $i < K$, is updated by either a Metropolis-Hastings kernel with stationary distribution $\tilde{\pi}_i$ or by an equi-energy jump. More precisely, an equi-energy jump between two successive chains i and $i-1$ is the following: a state y is chosen from the chain i such that $h(y)$ and $h(x_{i-1})$ belong to energy rings of similar energy. Then y is accepted to be the next state of the $(i-1)$th chain with probability

$$\min\left\{1, \frac{\tilde{\pi}_{i-1}(y)\tilde{\pi}_i(x_{i-1})}{\tilde{\pi}_{i-1}(x_{i-1})\tilde{\pi}_i(y)}\right\},$$

3 PTEEM algorithm

3.1 Description of the algorithm

We introduce a sequence of $d+1$ energy levels $H_1 < H_2 < \cdots < H_{d+1} = \infty$ with $H_1 \leq \min(h(x))$, and a sequence of N temperatures $T_1 = 1 < T_2 < \cdots < T_N$. The algorithm considers a population of N chains associated with probability measures $\pi_i(x) \propto \pi(x)^{1/T_i}$, each π_i being a density with respect to a probability measure λ on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, where $\mathcal{X}$ coincides with the support of the π_i and $\mathcal{B}(\mathcal{X})$ stands for the associated σ-algebra.

Energy rings $D_j, j = 1, \ldots, d$ are constructed as follows: the state space $\mathcal{X}$ is partitioned according to the energy levels: $\mathcal{X} = \bigcup_{j=1}^{d} D_j$, where

$$\begin{aligned} D_j &= \{x \in \mathcal{X}; h(x) \in [H_j, H_{j+1})\}, \quad j = 2, \cdots, d \\ D_1 &= \{x \in \mathcal{X}; h(x) \in (-\infty, H_2)\}. \end{aligned}$$

Compared to the energy rings of the EES method, these rings contain only current states, and there is only one sequence of energy rings for all the chains. By contrast the rings defined by Kou et al. [2006] contain past states, and a sequence of energy rings is constructed for each chain.

Each step of the PTEEM algorithm is decomposed into two types of moves: local moves via classical MCMC algorithms and global moves allowing an exchange between two chains with similar energies.

Local moves Each chain is locally updated, independently of others. In particular, the ith chain is updated using one iteration of a classical MCMC algorithm with stationary distribution π_i. This algorithm could be a Metropolis-Hastings algorithm, a Gibbs sampler, an hybrid MCMC (Robert and Casella [2004]), or a Reversible Jump MCMC (Green [1995], Richardson and Green [1997]).

Global moves At each step, an energy ring D_j containing at least two chains is chosen randomly. Two chains are then chosen uniformly in D_j, say the ith and the kth ones (with $i < k$),

and an exchange move is proposed between the current two states of these chains. The move is from $s = (x_1, \cdots, x_i, \cdots, x_k, \cdots, x_N)$ to $s' = (x_1, \cdots, x_k, \cdots, x_i, \cdots, x_N)$.
The product σ-algebra is written $\mathcal{B}(\mathcal{X})^{\mathcal{N}}$, and the product measure is denoted by λ_N. The probability measure π^* is defined as follows:

$$\pi^*(dx_1, dx_2, \ldots, dx_N) = \prod_{i=1}^{N} \pi_i(x_i)\lambda(dx_i) \qquad \text{on} \qquad (\mathcal{X}^N, \mathcal{B}(\mathcal{X})^N)$$

The probability acceptance for the global move is then given by:

$$\begin{aligned} \rho(s; s') &= \min\left\{1, \frac{\pi^*(s')}{\pi^*(s)}\right\} \\ &= \min\left\{1, \frac{\pi_i(x_k)\pi_k(x_i)}{\pi_i(x_i)\pi_k(x_k)}\right\}. \end{aligned} \tag{1}$$

Note that if the denominator is null, then the numerator is also null and by convention $\rho(s; s')$ is null. The chains are not Markov by themselves, it is the whole stochastic process made of the N chains together that forms a Markov chain on $(\mathcal{X}^N, \mathcal{B}(\mathcal{X})^N)$.

Remark 3.1 *It is of interest to compare the total number of local and global moves required in PTEEM and EES algorithms. Let us denote by B the size of the burn-in period, by R the number of iterations necessary to initialize energy rings within EES, and by M the sample size of the chains (after the burn-in period). We have:*

- *For EES, the total number of local moves is equal to*

$$K(B+R) + (M-R) + (1-p_{ee})\left(\frac{(K-1)K}{2}(B+R) + (K-1)(M-R)\right),$$

  and the total number of proposed global moves is equal to

$$p_{ee}\left(\frac{(K-1)K}{2}(B+R) + (K-1)(M-R)\right),$$

  where K denotes the number of chains in EES.

- *For PTEEM, the total number of local moves is $NM + NB$,*
  and the total number of global moves is $M + B$,
  where N stands for the number of chains in PTEEM.

In terms of computational cost, we should take into account that in some problems the local algorithms used by EES and PTEEM can be different (see Section 6), and thus can have different computational costs. In terms of storage, to obtain the $(i+1)$th iteration of the target chain, EES uses $KR + i + (1-p_{ee})\left(\frac{(K-1)KR}{2} + (K-1)i\right)$ values in memory to choose an element in an energy ring, whereas PTEEM necessitates only N values. Notice that CPU time to compute one iteration increases within EES as the simulations go along, while it is constant within PTEEM algorithm.

3.2 Some theoretical results

In this section standard sufficient conditions ensuring convergence of the PTEEM algorithm are given. Denote by S the Markov chain on $(\mathcal{X}^N, \mathcal{B}(\mathcal{X})^N)$ obtained by the PTEEM algorithm, a state of S is written s. The transition kernel associated with an iteration of PTEEM is written P, and P^k is the k-step transition kernel. They are defined on $\mathcal{X}^N \times \mathcal{B}(\mathcal{X})^N$. The transition kernel associated with the local move of the ith chain is written PL_i, and is defined on $\mathcal{X} \times \mathcal{B}(\mathcal{X})$. The transition kernel associated with the whole N local moves of an iteration of PTEEM is written PL, and is defined on $\mathcal{X}^N \times \mathcal{B}(\mathcal{X})^N$. The transition kernel associated with the equi-energy move is written PE, and is defined on $\mathcal{X}^N \times \mathcal{B}(\mathcal{X})^N$. Writing

$$\begin{aligned} s &= (x_1, \ldots, x_i, \ldots, x_k, \ldots, x_N) \\ s' &= (x'_1, \ldots, x'_i, \ldots, x'_k, \ldots, x'_N), \end{aligned}$$

we have

$$\begin{aligned} PL(s, s') &= \prod_{i=1}^{N} PL_i(x_i, x'_i) \\ P(s, s') &= (PE * PL)(s, s') = \int_{\mathcal{X}^N} PE(\tilde{s}, s') PL(s, \tilde{s}) d\tilde{s} \end{aligned}$$

Write $q(s, s')$ the auxiliary distribution to propose s' from s in an equi-energy move, and $q_i(x_i, x'_i)$ the auxiliary distribution to propose x'_i from x_i in a local move of the ith chain. The total variation norm for a measure μ on $(\mathcal{X}^N, \mathcal{B}(\mathcal{X}^{\mathcal{N}})$ is defined by:

$$\|\mu\|_{TV} = \sup_{A \in \mathcal{B}(\mathcal{X}^{\mathcal{N}})} |\mu(A)|.$$

Proposition 3.1 *If the transition kernels associated with the local moves are reversible with stationary distributions π_i, $i = 1, \cdots, N$, aperiodic and strongly λ-irreducible, then the chain S is strongly λ_N-irreducible and we have for π^*-almost all $s \in \mathcal{X}^N$*

$$\lim_{n \to \infty} \|P^n(s, .) - \pi^*\|_{TV} = 0.$$

Therefore π^* is the stationary distribution of S and the chain associated with $T_1 = 1$ provides samples corresponding to $\pi_1 = \pi$, which is the target distribution.

Proof. See Appendix B.1.

Remark 3.2 *In Proposition 3.1, the transition kernels of the local moves are assumed to be aperiodic. We can relax this hypothesis. In fact, it is sufficient that only one of the N transition kernel is aperiodic to have P aperiodic.*
However, all transition kernels should be irreducible to have P irreducible.

This proposition has minimal assumptions, which are usually not difficult to verify, especially for classical MCMC algorithms like Metropolis-Hastings algorithms or Gibbs samplers. However, it is possible to have a null set of states from which convergence does not occur. The following lemma and proposition have stronger assumptions that ensure convergence from all starting points.

Lemma 3.1 *Assume that the transition kernels associated with the local moves are reversible with stationary distributions π_i, $i = 1, \cdots, N$, aperiodic and strongly λ-irreducible, and assume the positivity of the density π^* on $\mathcal{X}^N$ ($\forall s \in \mathcal{X}^N, \pi^*(s) > 0$). Then the chain S is strongly λ_N-irreducible, positive and Harris-recurrent.*

Proof. See Appendix B.2.

The following proposition is a consequence of Lemma 3.1.

Proposition 3.2 *Assume that the transition kernels associated with the local moves are reversible with stationary distributions π_i, $i = 1, \cdots, N$, aperiodic and strongly λ-irreducible, and assume the positivity of the density π^* on $\mathcal{X}^N$ ($\forall s \in \mathcal{X}^N, \pi^*(s) > 0$). Then we have for all $s \in \mathcal{X}^N$*

$$\lim_{n \to \infty} \|P^n(s, .) - \pi^*\|_{TV} \quad = \quad 0.$$

Proof: Using Lemma 3.1 and Proposition 3.1, S is a Markov chain π^*-irreducible, aperiodic, with stationnary distribution π^* and Harris-recurrent. The result follows from Theorem 1 of Tierney [1994]. □

3.3 Choice of energy ladder and temperatures

Following our experience we suggest a simple way to calibrate energy ladder and temperatures.

Energy ladder The levels $H_1, H_2, \ldots, H_d$ are associated with d energy rings, the first one including states having an energy value lower than H_2 and including only few states having an energy value lower than H_1, and the last one including states having an energy value higher than H_d. Once the values H_1 and H_d are chosen, the other energy levels can be set to be evenly spaced on a logarithmic scale

$$ln(H_i) = ln(H_1) + i\frac{ln(H_d) - ln(H_1)}{d-1}.$$

To choose H_1 and H_d we use one or few runs of a classical MCMC algorithm with target density π. We take for H_d the energy associated with a state with high enough finite energy compared to other states. Concerning H_1, we take the energy corresponding to an observed mode. In practice, we can take for H_d the energy associated with a state after few iterations of the algorithm, and for H_1 the energy associated with a state after a burn-in period.

Remark 3.3 *Concerning H_1, if the modes of the distribution of interest are known, we just have to take H_1 slightly lower than the energy of the highest mode.*

Temperatures The distribution associated with the highest temperature should be sufficiently flattened so that the associated chain can move freely from one mode to another. After choosing a T_N value we just have to check that the associated chain moves easily. T_1 is obviously equal to 1, and is associated with the chain of interest. Once T_1 and T_N are fixed, the other temperatures can be chosen by evenly spacing them on a logarithmic scale, by evenly spacing their inveres, or by evenly spacing their inverses geometrically (see for instance Kou et al. [2006], Nagata and Watanabe [2008] or Neal [1996]).

Checking that the choices of temperatures and energy ladder are relevant It is necessary to check on a run of PTEEM that the choices of temperatures and energy ladder are relevant. The chain 1 should have almost all its states in the first energy ring, the last chain should have almost all its states in the last energy ring, and between them the states of the different chains should be well distributed in the rings. The distribution in the rings can be considered as correct if there is no "energy gap" between adjacent chains, and if for each chain equi-energy moves are performed with several other chains. If poor mixing is observed between chains then it is necessary to adjust the temperatures or the energy levels, adding new temperatures for instance or proposing a new calibration. Following Atchadé et al. [2010], we can try to adjust the temperatures so that the proportion of accepted equi-energy exchange moves is approximately 0.234. This problem of calibration is illustrated in Table 1.

4 Example of simulations using local Metropolis-Hastings moves

To compare the three algorithms (PT, EES and PTEEM) when the local move is a Metropolis-Hastings algorithm, we consider sampling from a two-dimensional normal mixture model taken from Liang and Wong [2001] and used as an illustration by Kou et al. [2006]. Let

$$f(x) = \sum_{i=1}^{20} \frac{w_i}{\sigma_i \sqrt{2\pi}} exp\Big(- \frac{1}{2\sigma_i^2}(x-\mu_i)'(x-\mu_i)\Big),$$

where $\sigma_1 = \ldots = \sigma_{20} = 0.1$, $w_1 = \ldots = w_{20} = 0.05$, and the 20 mean vectors

$$(\mu_1, \ldots, \mu_{20}) = \begin{pmatrix} 2.18 & 8.67 & 4.24 & 8.41 & 3.93 & 3.25 & 1.70 & 4.59 & 6.91 & 6.87 \\ 5.76 & 9.59 & 8.48 & 1.68 & 8.82 & 3.47 & 0.50 & 5.60 & 5.81 & 5.40 \\ 5.41 & 2.70 & 4.98 & 1.14 & 8.33 & 4.93 & 1.83 & 2.26 & 5.54 & 1.69 \\ 2.65 & 7.88 & 3.70 & 2.39 & 9.50 & 1.50 & 0.09 & 0.31 & 6.86 & 8.11 \end{pmatrix}.$$

The different local modes are quite far from each other (most of them are more than 15 standard deviations from the nearest ones), hence this mixture distribution is quite challenging for sampling. In addition, the initial states of the different chains were drawn from a uniform distribution on $[0,1]^2$, a region far from the local modes.
Each algorithm was run 100 times. For each run, the PT and PTEEM algorithms were run for 2500 iterations after a burn-in period of 2500 iterations. Similarly, for each chain of the EES the burn-in period was of 2500 iterations, and for the first chain (the target chain) 2500 iterations were simulated after this burn-in period and the period to construct the rings, which was of 500

iterations. As in Kou et al. [2006], the Metropolis-Hastings proposal was a bivariate Gaussian $X_{n+1}^{(i)} \sim \mathcal{N}_2(X_n^{(i)}, \tau_i^2 I_2)$, with $\tau_i = 0.25\sqrt{T_i}$. Unlike them, the step size τ_i was not tuned later in the algorithms such that the acceptance ratio is in the range (0.22,0.32). Indeed, we would like to compare algorithms as simple as possible. For the EES, we took the same number of chains, the same energy levels, the same temperatures and the same equi-energy jump probability than Kou et al. [2006] ($K = 5$, $H = (0.2, 2, 6.3, 20, 63.2)$, $T = (1, 2.8, 7.7, 21.6, 60)$, $p_{ee} = 0.1$). For the PT and PTEEM algorithms, $N = 20$ chains were taken, with temperatures between 1 and 60 evenly spaced on a logarithmic scale. As in Kou et al. [2006], the PT algorithm used a swap between neighboring temperature chains for the exchange operation, but only one swap was proposed at each iteration, to make it comparable with the PTEEM. For the PTEEM, the same 5 groups of energy than for the EES were taken.

Mean acceptances rates for the local Metropolis-Hastings moves and for the exchange moves between chains for the three algorithms are given in Table 2. In comparison Kou et al. [2006] obtained results slightly different probably because the step size τ_i was tuned in their EES.

To compare the ability of each algorithm to explore the distribution space, we considered for each run of each algorithm the number and frequency of visited modes by the target chain, as well as the estimations of the mean vector $(E(X_1), E(X_2))$ and of the second moments $(E(X_1^2), E(X_2^2))$ using the samples generated from the target chain. Table 3 contains these estimations. Concerning the estimations of the mean vector and of the second moments, the EES and PTEEM estimates were more accurate than those of the PT, with smaller mean squared errors. Moreover, it appeared that the PTEEM estimates were slightly more accurate than those of the EES. Concerning the number of visited modes, good results were obtained by the EES and PTEEM algorithms compared to the PT. The results are reported in Table 4. The mean number of visited modes by the PT on the 100 runs was 14.31, compared to 19.92 for the EES and 19.98 for the PTEEM. Then, as in Kou et al. [2006], we counted in each of the 100 runs for the three algorithms how many times the target chain visited each mode in the last 2500 iterations. The absolute frequency error is given by $err_i = |\hat{f}_i - 0.05|$, where $\hat{f}_i$ is the sample frequency of the ith mode being visited ($i = 1, \ldots, 20$). The median and the maximum of err_i over the 100 runs was calculated. To compare the three algorithms the ratios of these values between PT and EES, between PT and PTEEM and between EES and PTEEM were calculated for each mode. All these ratios are presented in Table 5. As denoted in Kou et al. [2006], EES seemed to be more efficient than PT: the mean of the ratios $R_{med(PT/EES)}$ over the 20 modes was 2.42, and the mean of the ratios $R_{max(PT/EES)}$ over the 20 modes was 2.92. As expected, PTEEM gave better results than PT: the mean of $R_{med(PT/PTEEM)}$ was 2.52, and the mean of $R_{max(PT/PTEEM)}$ was 3.07. Besides, we noticed a slight improvement of PTEEM compared to EES: 1.05 for the mean of $R_{med(EES/PTEEM)}$, and 1.13 for the mean of $R_{max(EES/PTEEM)}$.

Figures 1 and 2 show the last 2500 iterations after burn-in for the chains 1, 7, 14 and 20 obtained by one run of the PT algorithm, and by one run of the PTEEM algorithm. Figure 3 shows the simulations after a burn-in period for chains 1 to 5 obtained by a run of EES. The first chains of the PTEEM and EES visited all the modes of the target density whereas the first chain of PT did not visit all of them. Notice that chains with the highest temperatures of the PT algorithm visited all the modes, and these chains for the EES kept in memory lots of iterations.

Table 6 presents the repartition of accepted equi-energy moves for chains 1, 10 and 20,

with other possible chains within a run of the PTEEM algorithm. As expected, the closer the temperatures of chains were, the more often the equi-energy moves were accepted. Note that equi-energy moves had been proposed and accepted for all possible pairs of chains, including for pairs of chains with very different temperatures.

As in Kou et al. [2006], it appeared that the EES algorithm gave better results than the classical PT. Besides the PTEEM algorithm gave results comparable to those of the EES, and even slightly better.

5 Example of estimation using local Gibbs samplers moves

We consider estimation of model parameters in case of a mixture model with known number of components. The classical algorithm used for this kind of problem is a Gibbs sampler. However, some difficulties are encountered to combine EES with a Gibbs sampler. Therefore, we compared only performances of PT and PTEEM algorithms, using the well-known example of the Galaxy dataset (see for instance Richardson and Green [1997]).
We consider independent observations $y_1, \cdots, y_n$ from k mixture components

$$y_i \sim \sum_{j=1}^{k} w_j f(.|\mu_j, \sigma_j^2), \qquad i = 1, \ldots, n,$$

with k fixed and known and where $f(.|\mu_j, \sigma_j^2)$ denotes the density of the Gaussian distribution $\mathcal{N}(\mu_j, \sigma_j^2)$. The sizes of the k groups are proportional to $w_1, w_2, \ldots, w_k$, which are the weights of the components. The parameters to be estimated are the means μ_j, the variances σ_j^2, and the weights w_j, for $j = 1, \ldots, k$. The label of the component from which each observation is drawn is unknown, and a label vector c which is a latent allocation vector is introduced as follows: $c_i = j$ if the observation y_i is drawn from the jth component. The variables c_i are supposed independent with distributions

$$p(c_i = j) = w_j, \qquad j = 1, \ldots, k.$$

Write $y = (y_i)_{i=1,\ldots,n}$, $\mu = (\mu_j)_{j=1,\ldots,k}$, $\sigma^2 = (\sigma_j^2)_{j=1,\ldots,k}$, $w = (w_j)_{j=1,\ldots,k}$ and $c = (c_i)_{i=1,\ldots,n}$. The μ_j and σ_j^{-2} are supposed to be independent with the following priors:

$$\mu_j \sim \mathcal{N}(\xi, \kappa^{-1}), \qquad \sigma_j^{-2} \sim \Gamma(\alpha, \beta) \qquad \text{and} \qquad \beta \sim \Gamma(g, h), \tag{2}$$

where β and h are rate parameters. The prior on w is taken as a symmetric Dirichlet distribution

$$w \sim D(\delta, \delta, \ldots, \delta).$$

The parameters δ, ξ, κ, α, g and h are supposed to be fixed. Let us denote by $m_j = \sum_{i=1}^{n} \mathbb{1}_{c_i=j}$ the number of observations labeled by j. The joint posterior density, the full conditional distributions and the formula of the acceptance rate for the equi-energy move are given in Appendix A.
In this example, the estimates of the parameters obtained after labeling were quite good and

similar for the PT and PTEEM algorithms. They were even comparable to those obtained with a classical Gibbs sampler. The major difference between these three algorithms was the ability to explore the parameter space: the Gibbs sampler found one mode of the mixture posterior and usually was staying only on this mode, while the PT and PTEEM algorithms succeeded to jump from one mode to another. Consequently, we focused on the label-switching phenomenon (see Jasra et al. [2005]), and not on the estimation of the parameters.

The data consist of the velocities of 82 distant galaxies diverging from our own. We fix the number of components to $k = 6$, and we took for the fixed parameters in (2): $\alpha = 3$, $\xi = 20$, $\delta = 1$, $\kappa = 1/R^2$, $g = 0.2$ and $h = 10/R^2$, where $R = 10$. The algorithms PT and PTEEM were run 100 times, each run consisting of 10000 iterations after a burn-in period of 2000 iterations. We used 20 chains and 5 energy rings. As in the previous example, the PT algorithm used a swap between neighboring temperature chains for the exchange operation, and only one swap was proposed at each iteration. Concerning the energy ladder, after a run of a classical Gibbs sampler with target density π, we chose $H_1 = 180$ and $H_5 = 260$. Four energy rings were obtained with levels evenly spaced between H_1 and H_5 on a logarithmic scale, the fifth ring containing all states having an energy value higher than H_5. The levels obtained were 180, 197.3, 216.3, 237.2 and 260. We chose $N = 20$ temperatures between 1 and 4, with their inverses evenly spaced. Table 7 shows for several chains the distributions of states in the energy rings.

Clearly, the mixture posterior has $k! = 720$ symmetric modes and, in theory, for a very high number of iterations, the chain of interest should have visited all modes, with equal frequencies. When the chain goes from one mode to another, there is the so-called label-switching phenomenon (see Jasra et al. [2005]). Such a phenomenon is a useful convergence diagnostic to check if the chain of interest has explored all possible labelings of the parameters. To compare PT and PTEEM algorithms we considered for each run of each algorithm both the number and the frequency of visited modes by the target chain. Table 8 shows that on 100 runs of PTEEM the target chain visited more modes than on 100 runs of PT. Hence the label-switching phenomenon seems to occur more often during a run of PTEEM than during a run of PT. We also counted in each of the 100 runs for the two algorithms how many times the target chain visited each mode in the last 10000 iterations. The absolute frequency error is given by $err_i = |\hat{f}_i - 1/6!|$, where $\hat{f}_i$ is the sample frequency of the ith mode being visited ($i = 1, \ldots, 6!$). We then calculated the mean and median of this absolute frequency error over the 100 runs and the 6! modes. Absolute frequency errors were slightly lower for PTEEM with a mean (resp. a median) of 0.119% (resp. 0.099%), compared to 0.126% (resp. 0.099%) for PT.

We studied further the equi-energy moves of the algorithm PTEEM. In Table 9 it appears that exchange moves were more frequent between chains with similar temperatures. The mean acceptance rates of the equi-energy moves for PTEEM and of the exchange moves for PT were of 49% and 61% respectively. Note that we could code the PT algorithm so that exchange moves can be proposed between any two chains and not only between adjacent chains. But in this case the mean acceptance rate of an exchange move would be much lower. In comparison the PTEEM algorithm has the advantage to propose exchanges moves between chains not necessarily adjacent, but more relevant in terms of energy levels.

6 A complex problem: discovery of gene regulatory binding motifs

6.1 Model and data

The discovery of binding motifs in order to understand gene regulation is an important topic in biology. Indeed, a first step to understand gene expression is to know which are the corresponding binding sites of a common transcription factor (BSTF). The identification of these BSTF is a major computational problem, often studied these last twenty years (see for instance Stormo and Hartzell [1989], Lawrence and Reilly [1990], Lawrence et al. [1993], Liu et al. [1995] or Jensen et al. [2004]).

The data often consist of several homologous DNA sequences, and finding the BSTF is equivalent to identifying the starting positions of these sites in the sequences. Denote by S the set of M sequences, each one containing zero, one or more BSTF. Each sequence is made of four nucleotides: A, C, G or T. The BSTF are assumed to be of known length w. The length of the mth sequence is L_m, hence the number of possible starting positions for BSTF is denoted by $L_m^* = L_m - (w-1)$. The total number of motif sites is unknown and is denoted by $|A|$. As this number is unknown, the M sequences (without their $w-1$ last nucleotides) are considered as one long sequence of length $L^* = \sum_{m=1}^{M} L_m^*$. This long sequence contains $|A|$ BSTF. To identify the most promising positions for the BSTF, we introduce a missing vector $A = (a_1, a_2 \ldots, a_{L^*})$, where $a_i = 1$ if the ith position of the long sequence is the starting point of a BSTF, and $a_i = 0$ otherwise. Given A, the set S can be written as the union of two disjoint subsets: $S(A) \bigcup S(A^C)$, where $S(A)$ contains the aligned motifs of the identified BSTF, $S(A^C)$ representing the background sequence. Two different models are used for these two subsets. Concerning the background sequence, the simplest model is a product multinomial model (see Liu et al. [1995]), but it has been shown that a Markov model is biologically more relevant and improves the results obtained (see Jensen et al. [2004]). However, it makes the motif discovery more difficult, as repeated patterns are local modes for the algorithms. Following Kou et al. [2006] we used a Markov model of order one based on the following transition matrix

$$\theta_0 = \begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix}$$

where $\alpha = 0.12$. The parameter θ_0 is assumed to be known (in practice it can be easily well estimated from the data). Concerning $S(A)$, it can be seen as a matrix of dimensions $|A| \times w$, with the BSFT in rows. The kth column contains the nucleotides in kth position of the $|A|$ sites. Let $C = (C_1, ..., C_w)$ be a count vector, where $C_k = (C_{kA}, C_{kC}, C_{kG}, C_{kT})$ is the vector of the nucleotides counts in position k of all the sites. The common pattern of the BSTF is modeled by a product multinomial distribution of parameter $\Theta = (\theta_1, ..., \theta_w)$ where $\theta_k = (\theta_{kA}, \theta_{kC}, \theta_{kG}, \theta_{kT})$ is a probability vector for the preference of the nucleotide types in position k. According to the model, each vector C_k has a multinomial distribution with parameter θ_k independent of the

other columns. For this example we used

$$\Theta = \begin{pmatrix} 0.6 & 0.1 & 0 & 0.6 & 0,1 & 0 & 0.3 & 0 & 0.2 & 0 & 0.5 & 0 \\ 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.25 & 0.7 \\ 0 & 0.2 & 0 & 0.1 & 0.8 & 0.7 & 0 & 0.9 & 0 & 0 & 0.25 & 0.2 \\ 0.4 & 0.7 & 0.2 & 0.3 & 0,1 & 0.3 & 0.7 & 0.1 & 0.6 & 0.5 & 0 & 0.1 \end{pmatrix}.$$

The corresponding WebLogo (Crooks et al. [2004]) is given in Figure 6.1.

To complete the model, conjugate prior distributions are considered. The distribution of Θ is a product of Dirichlet with parameters $(\beta_1, \ldots, \beta_w)$:

$$\pi(\Theta) \propto \prod_{k=1}^{w} \theta_k^{\beta_k - 1}, \qquad \text{where} \qquad \theta_k^{\beta_k} = \prod_{j=\{A,C,G,T\}} \theta_{kj}^{\beta_{kj}}.$$

The prior probability of a component a_i of A is denoted by p_0, which is the "site abundance" parameter:

$$\pi(A \mid p_0) = p_0^{|A|}(1 - p_0)^{L^* - |A|}.$$

Finally, this parameter p_0 is assumed to follow a beta distribution $Be(a, b)$.

From θ_0 and Θ, we generated $M = 10$ background sequences of length 200, and $|A| = 20$ BSTF of length $w = 12$. Two BSTF were introduced in each of the ten sequences, hence we obtained 10 sequences of length 224.

6.2 Classical approach: the Gibbs sampler

To solve the challenging problem of identifying BSTF, bayesian approaches using Gibbs samplers were developed by Lawrence et al. [1993], Liu et al. [1995], or Liu [1994]. The missing vector A giving the starting positions of the BSTF is of interest, hence the aim is to build a Markov chain having the posterior distribution of A as stationnary distribution.

Following Kou et al. [2006], in order to obtain the posterior of interest $\pi(A \mid S)$, the collapsing technique of Liu [1994] is used to integrate out the unknown parameters Θ and p_0 in the joint posterior distribution. As a consequence these parameters are not updated at each iteration and the computation time is reduced. Moreover, the use of this technique facilitates the convergence of the Markov chain, as noted by Liu [1994] and van Dyk and Park [2008]. The posterior of interest is given by

$$\pi(A \mid S) \quad \propto \quad \frac{1}{\pi(S(A)|A, \theta_0)} \frac{\Gamma(|A| + a)\Gamma(L^* - |A| + b)}{\Gamma(L^* + a + b)} \prod_{k=1}^{w} \frac{\Gamma(C_k + \beta_k)}{\Gamma(|A| + |\beta_k|)}, \tag{3}$$

with $\Gamma(C_k + \beta_k) = \prod_{j=\{A,C,G,T\}} \Gamma(C_{kj} + \beta_{kj})$. Using (3), a predictive update version of the Gibbs sampler has been proposed by Liu et al. [1995]. They suggest to update each component of A independently of the others using the following predictive update formula:

$$\frac{p(a_i = 1|A_{[-i]}, S)}{p(a_i = 0|A_{[-i]}, S)} \quad = \quad \frac{1}{\pi(S(a_i) \mid A, \theta_0)} \times \frac{|A_{[-i]}| + a}{L^* - A_{[-i]} - 1 + b} \times \prod_{k=1}^{w} \left(\frac{C_{k(-i)} + \beta_k}{|A_{[-i]}| + |\beta_k|} \right)^{C_{k(i)}}, \tag{4}$$

where the following notations are used: $A_{[-i]}$ represents the vector A without the ith component, $S(a_i)$ represents the sites of A starting in position i, $C_{k[-i]} = (C_{k[-i]A}, C_{k[-i]C}, C_{k[-i]G}, C_{k[-i]T})$ is the vector of the nucleotides counts in position k of all the sites, excluding the site starting in position i, $C_{k(i)} = (C_{k(i)A}, C_{k(i)C}, C_{k(i)G}, C_{k(i)T})$ is the vector of the nucleotides counts in position k of the site starting in position i (this vector contains three 0 and one 1). We have $C_k = C_{k[-i]} + C_{k(i)}$.

6.3 EES algorithm

The algorithms resulting from the Gibbs sampling approach, such as BioProspector (Liu et al. [2001]) or AlignACE (Roth et al. [1998]), are often trapped into local modes and true motif patterns are not found. Therefore Kou et al. [2006] proposed to use the EES algorithm, which seems to improve the global BSTF search. In this algorithm, K chains are used and the lth chain has the following target distribution

$$\tilde{\pi}_l(A) \propto exp\Big(- \frac{h(A) \vee H_l}{T_l}\Big), \qquad \text{with} \qquad h(A) = -log(\pi(A \mid S)).$$

For the target chain Kou et al. [2006] used a Gibbs sampler to generate the vector A. For the other chains, given the current sample A, they first estimated the common pattern by a frequency counting. Then they built a new vector according to the Bayes rule, which is accepted according to a Metropolis-Hasting move (see Kou et al. [2006] for more details).

6.4 PTEEM algorithm

In this algorithm, N chains are used and the lth chain has the following target distribution

$$\pi_l(A \mid S) = \pi(A \mid S)^{\frac{1}{T_l}},$$

and are locally updated by Gibbs samplers. Concerning the first chain, the updating of each component of A is done using the predictive update formula (4). Concerning the lth chain ($l > 1$), the predictive update formula to be used is the following:

$$\frac{p(a_i = 1|A_{[-i]}, S)}{p(a_i = 0|A_{[-i]}, S)} = \left(\frac{1}{\pi(S(a_i) \mid A, \theta_0)} \times \frac{|A_{[-i]}| + a}{L^* - A_{[-i]} - 1 + b} \times \prod_{k=1}^{w} \left(\frac{C_{k[-i]} + \beta_k}{|A_{[-i]}| + |\beta_k|} \right)^{C_{k(i)}} \right)^{\frac{1}{T_l}} \quad (5)$$

Concerning a proposed equi-energy move between two chains l_1 and l_2 of current states A_{l_1} and A_{l_2}, the acceptance probability is given by:

$$\rho = \min\{1, \frac{\pi_{l_1}(A_{l_2} \mid S)\pi_{l_2}(A_{l_1} \mid S)}{\pi_{l_1}(A_{l_1} \mid S)\pi_{l_2}(A_{l_2} \mid S)}\},$$

with

$$\begin{aligned}\frac{\pi_{l_1}(A_{l_2} \mid S)\pi_{l_2}(A_{l_1} \mid S)}{\pi_{l_1}(A_{l_1} \mid S)\pi_{l_2}(A_{l_2} \mid S)} &= \Big(\frac{\pi(A_{l_2} \mid S)}{\pi(A_{l_1} \mid S)}\Big)^{\frac{1}{T_{l_1}}-\frac{1}{T_{l_2}}} \\ &= \left[\frac{\pi(S(A_{l_1}) \mid A_{l_1}, \theta_0)}{\pi(S(A_{l_2}) \mid A_{l_2}, \theta_0)} \times \frac{\mathcal{B}(|A_{l_2}| + a, L^* - |A_{l_2}| + b)}{\mathcal{B}(|A_{l_1}| + a, L^* - |A_{l_1}| + b)}\right. \\ &\quad \left. \times \prod_{k=1}^{w} \frac{\Gamma(C_{l_2 k} + \beta_k)}{\Gamma(C_{l_1 k} + \beta_k)} \times \prod_{k=1}^{w} \frac{\Gamma(|A_{l_1}| + |\beta_k|)}{\Gamma(|A_{l_2}| + |\beta_k|)}\right]^{\frac{1}{T_{l_1}}-\frac{1}{T_{l_2}}}.\end{aligned}$$

6.5 Results

The algorithms EES and PTEEM as explained above were run 10 times each, on the data presented in 6.1. For each run of PTEEM, $N = 15$ chains were used, with a burn-in of 200 iterations and a post-burn-in of 800 iterations, resulting in 15000 local moves and 1000 proposed global moves. For each run of EES, $K = 9$ chains were used, with $p_{ee} = 0.1$, a burn-in of 200 iterations and a post-burn-in of 800 iterations, among which 100 iterations were used to construct energy rings. It results in 18160 local moves and approximately 1640 global moves.

Calibration
Concerning PTEEM, 5 energy rings were used, with energy levels regularly spaced on a logarithmic scale between 10 and 100, giving levels 10, 17, 78, 31.62, 56.23 and 100. The temperatures have their inverses regularly spaced between 1 and 1/1.3, giving $T_{min} = 1$ and $T_{max} = 1.3$. Concerning EES, 9 energy rings were used, with energy levels regularly spaced on a logarithmic scale between 10 and 100. The temperatures used are the following: 1, 1.001, 1.002, 1.005, 1.01, 1.02, 1.06, 1.1 and 1.3. This choice has been made in order to permit equi-energy jumps between chains. For instance, fixing $T_1 = 1$ and $T_2 = 1.1$, no jumps would have been possible between the first and the second chains, because the energies of chains associated with these temperatures are too different.

Local and global moves
Concerning the 10 runs of PTEEM, 55.71% of the proposed equi-energy moves were accepted, allowing a good mixing of the chains. The first chain exchanged states relatively easily with chains of lower orders, and it exchanged states even with chains 10 or 11. Concerning the 10 runs of EES, the last 8 chains were locally updated by Metropolis-Hastings algorithms: approximately 15% of new states proposed for the second to fifth chains were accepted, but only 2.9% were accepted for chain 8 and 0.7% for chain 9. The proposed equi-energy jumps were mainly accepted (86% in mean), but it is noticeable that very few jumps were proposed between the first and the second chain (26 jumps in mean during the 1000 iterations). Indeed, the states of these two chains often had energies quite different. As an example, the second chain never obtained a state in the first energy ring.

Identification of the BSTF
Results of the 10 runs of PTEEM were quite similar, as opposition to the 10 runs of EES. Figure

6.5 represents two boxplots representing empirical posterior probabilities $P(a_i = 1 \mid S), i = 1, \ldots, L^*$ obtained during a run of PTEEM, and during a run of EES. Hence these boxplots represent the posterior probabilities of each possible position to be the starting point of a BSTF. Only the positions associated to high posterior probabilities are relevant. On the boxplots of figure 6.5 for instance, we decided to keep only the positions with posterior probabilities higher than 0.8. Concerning the 10 runs of PTEEM, they identified 16 sites among 20. Among them, 15 were identified with exactly the true starting positions, and 1 was identified with 3 other positions (positions 877, 880 and 883 were kept, the true one being 880). Concerning the 10 runs of EES, they identified in mean 15.6 sites among 20. Among them, 9.6 were identified with exactly the true starting positions, and 6 were identified with phase-shifted positions or several positions. For example, a site has been identified by positions 1784 and 1791 while the true one was 1784, and another has been identified by position 27 while the true one was 26. Notice that 5 EES runsamong 10 obtained similar results as the PTEEM runs.

Conclusion on these results

Results obtained by EES could be improved with a better calibration. In particular, using more chains would improve the results (with a supplementary computational cost). However, the low number of jumps proposed between chains 1 and 2 is noticeable. It could be due to temperatures too far from each others, but as we used $T_1 = 1$ and $T_2 = 1.001$, t it should be due to the Metropolis-Hastings algorithm used to update the second chain. This algorithm could have difficulties to propose relevant states. Indeed, it is not easy to find a good proposal law for new states, and maybe the method proposed by Kou et al. [2006] is not the best possible. The difficulties encountered to calibrate the EES and the associated Metropolis-Hastings algorithms are a disadvantage to the use of this algorithm. In comparison, the calibration of PTEEM is much easier. Indeed, the Gibbs sampler does not need to be calibrated, and if temperatures and energy levels are well chosen, the number of accepted equi-energy moves between chains is sufficiently large to allow good mixing of the chains. We did not encounter difficulties to calibrate these parameters, suggestions of 3.3 giving good results.

Concerning the results obtained on this challenging example, those obtained with PTEEM were slightly better than those obtained with EES. The mixing of the chain of interest was more efficient in PTEEM. Hence, PTEEM identified exactly most of the true starting positions of BSTF, while EES tended to identified BSTF with several positions or phase-shifted positions. That means that EES was most often trapped in local phase-shift modes.

Note that this phase-shift problem is encountered by most of the methods used to identify BSTF, and solutions have been proposed, see Liu [1994] or Lawrence et al. [1993]. Implementation of these solutions in the algorithms PTEEM or EES is absolutely possible. Similarly, improvements can be carried out to these algorithms to allow BSTF of unknown length, several motifs of BSTF, or BSTF made of several non contiguous blocks, see Jensen et al. [2004] and Liu et al. [2001]. However, it would be easier to implement these improvements in a Gibbs sampler, and hence in PTEEM, than in EES.

7 Discussion

In this paper a new algorithm combining Parallel Tempering and Equi-Energy Sampler was proposed. Thanks to relevant equi-energy moves, the proposed PTEEM algorithm allows a good exploration of the parameter space and good mixing of the generated Markov chains, while ensuring the reversibility of the exchange moves. Therefore the generated Markov process theoretically converges to π^*, and the first chain generates samples corresponding to the distribution of interest π.

Compared to PT, this new algorithm has the same theoretical properties, while outperforming it. The drawback is that an energy ladder is needed, but we explained a simple and practical way to obtain a relevant ladder, which proved to be efficient.

Compared to EES, this new algorithm has the advantage to be based on Monte Carlo Markov chains theory, which is quite simple to use and to understand, even for non-experimented users. Less storage is also needed, since all iterations from the past are not kept in memory. Another substantial advantage of the PTEEM is that it can be coupled with a Gibbs sampler. Indeed, because of the use of an energy truncation, the EES can not be easily coupled with a Gibbs sampler to locally update the chains. In the cases where a Gibbs sampler can be used, our feeling is that this new algorithm can give results at least equivalent to those obtained with EES. On our examples, PTEEM gave results as good as those obtained with an EES.

A direction for future research is to investigate further the theoretical properties of the PTEEM algorithm, by comparing convergence rates of PTEEM and PT algorithms for instance. An adaptive PTEEM algorithm to finely tune the temperatures and/or the energy levels during a run would also be of interest.

References

C. Andrieu, A. Jasra, A. Doucet, and P. Del Moral. Convergence of the equi-energy sampler. *ESAIM: Proceedings*, 19:1–5, 2007. doi: 10.1051/proc:071901.

C. Andrieu, A. Jasra, A. Doucet, and P. Del Moral. A note on convergence of the equi-energy sampler. *Stochastic Analysis and Applications*, 26(2):298–312, 2008. doi: 10.1080/07362990701857178.

Y.F. Atchadé and J.S. Liu. Discussion of equi-energy sampler by kou, zhou and wong. *The Annals of Statistics*, 34(4):1620–1628, 2006.

Y.F. Atchadé, G.O. Roberts, and .S. Rosenthal. Towards optimal scaling of metropolis-coupled markov chain monte carlo. *Statistics and Computing*, 2010.

K.B. Athreya, H. Doss, and J. Sethuraman. On the convergence of the markov chain simulation method. *The Annals of Statistics*, 24(1):69–100, 1996.

G. Behrens, N. Friel, and M. Hurn. Tuning tempered transitions. *Unpublished manuscript*, 2009.

G.E. Crooks, G. Hon, J.M. Chandonia, and S.E. Brenner. Weblogo: A sequence logo generator-crooks, g.echandonia, j.m. *Genome Research*, 14:1188–1190, 2004.

G. Fort, E. Moulines, and P. Priouret. Convergence of adaptive and interacting mcmc algorithm. Technical report, 2010.

C.J. Geyer. Markov chain monte carlo maximum likelihood. *Computing Science and Statistics: Proceedings of the 23rd Symposium on the interface*, pages 156–163, 1991.

C.J. Geyer and E.A. Thompson. Annealing markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920, 1995.

P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

P.J. Green and A. Mira. Delayed rejection in reversible jump metropolis-hastings. *Biometrika*, 88:1035–1053, 2001.

W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 88:1035–1053, 1970.

X Hua and S. Kou. Convergence of the equi-energy sampler and its application to the ising model. *Statistica Sinica*, In press, 2010.

A. Jasra, C.C. Holmes, and D.A. Stephens. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.

A. Jasra, D.A. Stephens, and C.C. Holmes. Population-based reversible jump markov chain monte carlo. *Biometrika*, 94:787–807, 2007.

S.T. Jensen, X.S. Liu, Q. Zhou, and J.S. Liu. Computational discovery of gene regulatory binding motifs: A bayesian perspective. *Statistical Science*, 19:188–294, 2004.

S.C. Kou, Q. Zhou, and W.H. Wong. Equi-energy sampler with application in statistical inference and statistical mechanics. *The Annals of Statistics*, 34(4):1581–1619, 2006.

C.E. Lawrence and A.A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *PROTEINS: Structure, Fonction and Genetics*, 7:41–51, 1990.

C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, and A.F. Neuwald. Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

F. Liang and W.H. Wong. Real-parameter evolutionary monte carlo with applications to bayesian mixture models. *Journal of the American Statistical Association*, 96:653–666, 2001.

J.S. Liu. The collapsed gibbs sampler in bayesian computations with application to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

J.S. Liu, A.F. Neuwald, and C.E. Lawrence. Bayesian models for multi- ple local sequence alignment and gibbs sampling strategies. *Journal of the American Statistical Association*, 90 (432):1156–1170, 1995.

X. Liu, D.L. Brutlag, and J.S. Liu. Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing*, 6:127–138, 2001.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

K. Nagata and S. Watanabe. Asymptotic behavior of exchange ratio in exchange monte carlo method. *Neural Networks*, 21:980–988, 2008.

R.M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6:353–366, 1996.

S. Richardson and P.J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society B*, 59:731–792, 1997.

C. Robert and G. Casella. *Monte Carlo statistical methods, second edition.* Springer, 2004.

G.O. Roberts and J.S. Rosenthal. Harris recurrence of metropolis-within-gibbs and trans-dimensional markov chains. *The Annals of Applied Probability*, 16(4):2123–2139, 2006.

F.P. Roth, J.D. Hugues, J.W. Estep, and G.M. Church. Finding dna regu- latory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation roth, f.p. *Nature Biotechnology*, 16:939–945, 1998.

G.D. Stormo and G.W. Hartzell. Identifying protein-binding sites from un- aligned dna fragments. *Proceedings Of The National Academy Of Sciences Of the USA*, 86:1183–1187, 1989.

L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762, 1994.

D.A. van Dyk and T. Park. Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103:790–796, 2008.

A Formula used for the comparisons in case of a Gibbs sampler

A.1 Joint posterior densities

Write $x = (\mu, \sigma^{-2}, w, c, \beta)$. The joint posterior density from which the parameters should be drawn is:

$$\pi(x) \quad \propto \quad p(y \mid \mu, \sigma^{-2}, c) p(\mu, \sigma^{-2}, c, \beta, w).$$

Hence the ith chain should be drawn from

$$\pi_i(x) \propto \pi(x)^{\frac{1}{T_i}} \propto p(y \mid \mu, \sigma^{-2}, c)^{\frac{1}{T_i}} p(\mu, \sigma^{-2}, w, c, \beta)^{\frac{1}{T_i}}.$$

However, as noted by Jasra et al. [2007] and Behrens et al. [2009], tempering the whole posterior is problematic as there is no guarantee that the tempered posterior will remain proper. As a consequence, only the likelihood contribution is tempered and the priors are left untempered. The ith chain is then drawn from

$$\pi_i'(x) \quad \propto \quad p(y \mid x)^{\frac{1}{T_i}} p(x).$$

A.2 Full conditional distributions

Concerning the ith chain, the full conditional distributions to be used in the Gibbs sampler of the algorithms are easily obtained through conjugacy. We use the following notations:

$$\begin{aligned} x_i &= (\mu_i, \sigma_i^{-2}, w_i, c_i, \beta_i), \qquad & \mu_i &= (\mu_{i1}, \mu_{i2}, \ldots, \mu_{ik}), \\ \sigma_i^{-2} &= (\sigma_{i1}^{-2}, \sigma_{i2}^{-2}, \ldots, \sigma_{ik}^{-2}), \qquad & w_i &= (w_{i1}, w_{i2}, \ldots, w_{ik}), \\ c_i &= (c_{i1}, c_{i2}, \ldots, c_{in}), \qquad & m_i &= (m_{i1}, m_{i2}, \ldots, m_{ik}), \end{aligned}$$

with $p = 1, \ldots, k$ index of component and $l = 1, \ldots, n$ index of observation. For μ_i, σ_i^{-2} and w_i the full conditional distributions are the following

$$\mu_{ip} \mid \sigma_{ip}^{-2}, y, c_i, \xi, \kappa^{-1} \sim \mathcal{N}\left(\Big(\frac{m_{ip}\sigma_{ip}^{-2}}{T_i} + \kappa \Big)^{-1} \Big(\frac{\sigma_{ip}^{-2}}{T_i} \sum_{l:c_{il}=p} y_l + \xi\kappa \Big), \Big(\frac{m_{ip}\sigma_{ip}^{-2}}{T_i} + \kappa \Big)^{-1} \right),$$

$$\sigma_{ip}^{-2} \mid \mu_{ip}, y, c_i, \alpha, \beta_i \sim \Gamma\Big(\alpha + \frac{m_{ip}}{2T_i}, \beta_i + \sum_{l:c_{il}=p} \frac{(y_l - \mu_{ip})^2}{2T_i} \Big),$$

$$w_i \mid c_i, \delta \sim D(\delta + m_{i1}, \delta + m_{i2}, \ldots, \delta + m_{ik}).$$

For the allocation vector c, the full conditional distribution is multinomial with the following probabilities:

$$p_i(c_{il} = p \mid y, \mu_i, \sigma_i^{-2}, w_i) \propto \frac{1}{\sigma_{ip}^{\frac{1}{T_i}}} exp\Big(- \frac{(y_l - \mu_{ip})^2}{2\sigma_{ip}^2 T_i} \Big) w_{ip}.$$

The parameter β_i has the following full conditional distribution:

$$\beta_i \mid \sigma_i^{-2}, \alpha, g, h \sim \Gamma\Big(g + k\alpha, h + \sum_{p=1}^{k} \sigma_{ip}^{-2} \Big).$$

A.3 Acceptance rate of an equi-energy move

Assuming that two chains i and j are selected from an energy ring to be swapped, the acceptance probability of an equi-energy move proposed between two chains is given by

$$\min\left(1, \frac{\pi_i'(x_j)\pi_j'(x_i)}{\pi_i'(x_i)\pi_j'(x_j)}\right) = \min\left(1, \Big(\frac{p(y|x_i)}{p(y|x_j)}\Big)^{(1/T_j - 1/T_i)}\right),$$

where

$$p(y|x) \quad \propto \quad \prod_{p=1}^{k} (\sigma_p \sqrt{(}2\pi))^{-m_p} exp\Big(- \sum_{l=1}^{n} \frac{(y_l - \mu_{c_l})^2}{2\sigma_{c_l}^2} \Big).$$

B Proofs of Proposition 3.1 and Lemma 3.1

B.1 Proof of Proposition 3.1

During an iteration of the PTEEM algorithm all chains are locally updated by a MCMC algorithm and an exchange move is proposed. By assumption, $PL_i(.,.)$ is reversible with stationary distribution π_i. It is then clear that $PL = \prod_{i=1}^{N} PL_i$ is also reversible. Let $A \in \mathcal{B}(\mathcal{X})^N$, which can be written as $A_1 \times A_2 \times \ldots \times A_N$, with $A_i \in \mathcal{X}$. We have

$$\pi^*(A) \quad = \quad \int_{\mathcal{X}^N} PL(s, A)\pi^*(ds),$$

which implies that π^* is the stationary distribution of $PL(.,.)$. Then, the transition kernel PE can be written as

$$PE(s, s') = q(s, s')\rho(s, s') + \int_{\mathcal{X}} q(s, s'')(1 - \rho(s, ds''))\mathbb{1}_{\{s'\}}(s). \tag{6}$$

A sufficient condition to satisfy the detailed balance condition is the following:

$$q(s, ds')\rho(s, s')\pi^*(ds) = q(s', ds)\rho(s', s)\pi^*(ds'). \tag{7}$$

In the PTEEM algorithm, the two candidate chains to exchange their actual states are chosen uniformly among all chains in the same energy ring. Hence we have $q(s, s') = q(s', s)$. Using (1), it follows that (7) is satisfied, and the detailed balance condition holds. Therefore the transition kernel PE for the equi-energy move is reversible, with stationary distribution π^*. The transition kernels PE and PL are reversible with stationary distribution π^*. It is then clear that P is also reversible and that π^* is its stationary distribution. In addition, each PL_i is supposed to be strongly λ-irreducible and aperiodic, hence PL is aperiodic and strongly λ_N-irreducible. Since PE is just an exchange kernel between two actual states it is clear that $P = PE * PL$ is also strongly λ_N-irreducible and aperiodic. Theorem 1 of Tierney [1994] then allows to conclude. □

B.2 Proof of Lemma 3.1

From Proposition 3.1, S is reversible with stationary distribution π^*, and strongly λ_N-irreducible. It follows that S is positive. Note that a state s' reached from a starting point s after an iteration of PTEEM can not be part of a set $A \in \mathcal{X}^N$ such that $\pi^*(A) = 0$ (proof inspired from Roberts and Rosenthal [2006], Theorem 8).

To show that S is Harris-recurrent we use Theorem 2 of Tierney [1994] that characterizes Harris-recurrent chains as follows: a Markov chain is Harris-recurrent if and only if the only bounded functions h satisfying

$$\mathbb{E}(h(S^{(n)})|s_0) = \mathbb{E}(h(S^{(1)})|s_0) = h(s_0), \qquad \forall n \in \mathbb{N}, \tag{8}$$

are the constant functions. Functions h satisfying (8) are called harmonic. We use Theorem 6.80 of Robert and Casella [2004], inspired from Athreya et al. [1996] as follows:
If the transition kernel P satisfies: $\exists B \in \mathcal{B}(\mathcal{X})^N$ such that

(i) $\forall s_0$, $\sum_{n=1}^{\infty} \int_B P^n(s_0, s) d\mu(s) > 0$, with μ the initial distribution of the chain.

(ii) $\inf_{s,s' \in B} P(s, s') > 0$

Then, for π^*-almost all s_0,

$$\lim_{n\to\infty} \sup_{A \in \mathcal{B}(\mathcal{X})^N} \Big| \int_A P^n(s_0, s) ds - \int_A \pi^*(s) ds \Big| = 0, \tag{9}$$

To apply this result, notice that Assumptions (i) and (ii) are verified for $B = \mathcal{X}^N$. Equation (9) is then satisfied for π^*-almost all s_0.

Using

$$\|\mu\|_{TV} = \sup_{A \in \mathcal{B}(\mathcal{X})^N} |\mu(A)| = \frac{1}{2} \sup_{|h|<1} | \int h(x) \mu(dx)|,$$

this equation (9) can be written as

$$\lim_{n\to\infty} \sup_{|h|<1} \Big| E[h(S_n)|s_0] - E_{\pi^*}[h(s)] \Big| = 0.$$

We can extend this result for all bounded function h. Moreover, if h bounded satisfies (8), then $E[h(S_n)|s_0] = h(s_0)$. We then have $h(s_0) = E_{\pi^*}[h(s)]$ for π^*-almost all s_0, and h is π^*-almost everywhere constant and equal to $\mathbb{E}_{\pi^*}(h(S))$. Analysis similar to that in the proof of Theorem 6.80 of Robert and Casella [2004] shows that h is everywhere constant and equal to $\mathbb{E}_{\pi^*}(h(S))$. The Harris-recurrence then follows.

	Bad repartition					Good repartition				
Energy ring	1	2	3	4	5	1	2	3	4	5
chain $i-2$	990	10	0	0	0	990	10	0	0	0
chain $i-1$	950	50	0	0	0	701	202	97	0	0
chain i	900	100	0	0	0	387	408	205	0	0
chain $i+1$	0	2	237	511	250	45	312	355	288	0
chain $i+2$	0	0	105	610	285	0	64	517	353	66

Table 1: Illustration for bad and good repartitions of the states in the energy rings. There is an energy gap between chains i and $i+1$ in the bad repartition case.

	Local moves	Exchange moves
EES	0.387	0.799
PT	0.337	0.905
PTEEM	0.333	0.822

Table 2: Mean acceptance rates for local moves and exchange moves on 100 runs, for EES, PT and PTEEM algorithms.

	$E(X_1)$	$E(X_2)$	$E(X_1)^2$	$E(X_2)^2$
True value	4.478	4.905	25.605	33.920
EES	4.448 (0.301)	4.953 (0.458)	25.229 (3.112)	34.226 (4.507)
PT	3.971 (0.809)	4.137 (1.114)	21.510 (7.741)	27.510 (10.407)
PTEEM	4.483 (0.324)	4.912 (0.454)	25.556 (3.366)	33.889 (4.406)

Table 3: Estimations of the mean vector $(E(X_1), E(X_2))$ and of the second moments $(E(X_1^2), E(X_2^2))$ using the samples generated from the target chain, obtained on 100 runs for EES, PT and PTEEM algorithms. The standard deviations are given between parentheses.

PT	EES	PTEEM
2 to 10 missed.	1 missed for 4 runs.	1 missed for 2 runs.
A mean of 5.69 missed.	2 missed for 2 runs.	

Table 4: Number of missed modes by the 100 runs for EES, PT and PTEEM algorithms.

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
R_{med} PT/EES	2.16	2.80	2.92	2.22	1.98	2.21	3.10	2.07	2.07	2.69
R_{max} PT/EES	3.59	2.61	2.81	2.10	1.55	2.43	2.54	1.53	2.93	4.50
R_{med} PT/PTEEM	2.60	3.72	2.63	2.19	1.79	2.97	2.77	2.55	2.32	2.64
R_{max} PT/PTEEM	3.44	1.76	2.27	2.30	2.44	3.06	5.23	2.92	2.83	5.23
R_{med} EES/PTEEM	1.21	1.33	0.90	0.99	0.91	1.35	0.89	1.23	1.12	0.98
R_{max} EES/PTEEM	0.96	0.67	0.81	1.09	1.58	1.26	2.06	1.91	0.96	1.16
	μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{16}	μ_{17}	μ_{18}	μ_{19}	μ_{20}
R_{med} PT/EES	2.51	2.46	2.77	2.63	2.39	1.76	3.06	2.22	2.10	2.37
R_{max} PT/EES	4.58	1.60	3.23	4.61	3.26	2.10	2.83	4.77	3.50	1.36
R_{med} PT/PTEEM	2.14	1.98	1.79	2.84	2.75	2.18	2.72	2.78	2.43	2.60
R_{max} PT/PTEEM	3.05	2.02	2.35	4.16	3.44	1.79	3.72	3.78	3.50	2.16
R_{med} EES/PTEEM	0.85	0.81	0.65	1.08	1.15	1.24	0.89	1.25	1.16	1.10
R_{max} EES/PTEEM	0.67	1.26	0.73	0.90	1.06	0.85	1.32	0.79	1.00	1.58

Table 5: For each mode, ratios of median (R_{med}) and ratios of maximum (R_{max}) are for PT over EES, PT over PTEEM, and EES over PTEEM. Each ratio is obtained on 100 runs.

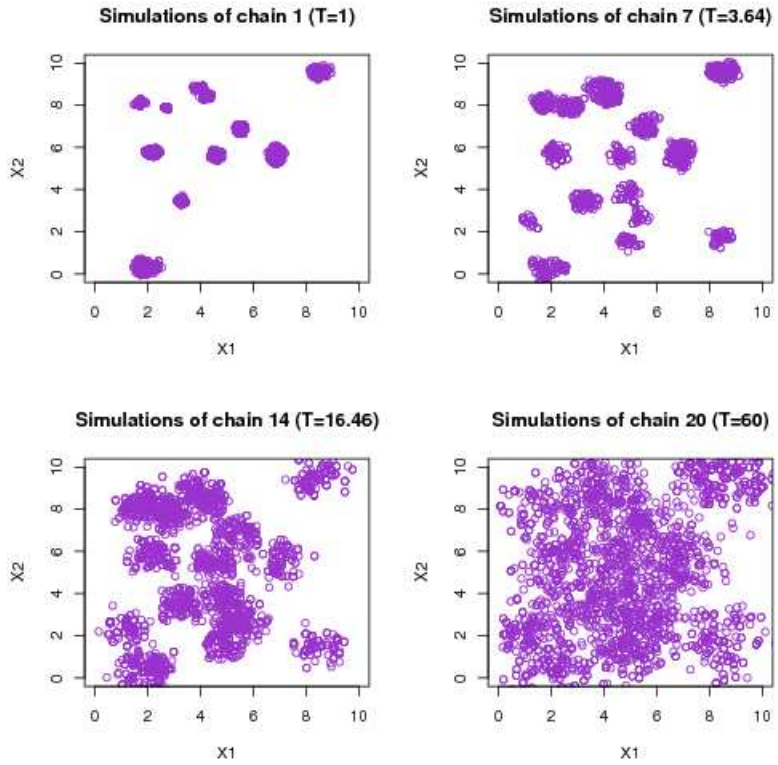

Figure 1: Simulations for chains 1, 7, 14 and 20 obtained by one run of the PT algorithm.

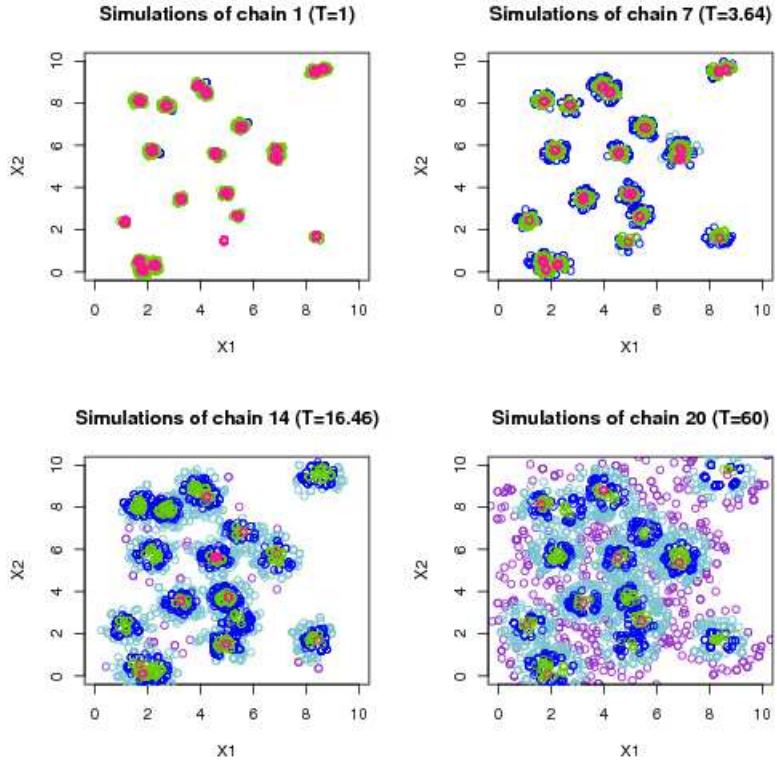

Figure 2: Simulations for chains 1, 7, 14 and 20 obtained by one run of the PTEEM algorithm. The colors correspond to the five energy levels.

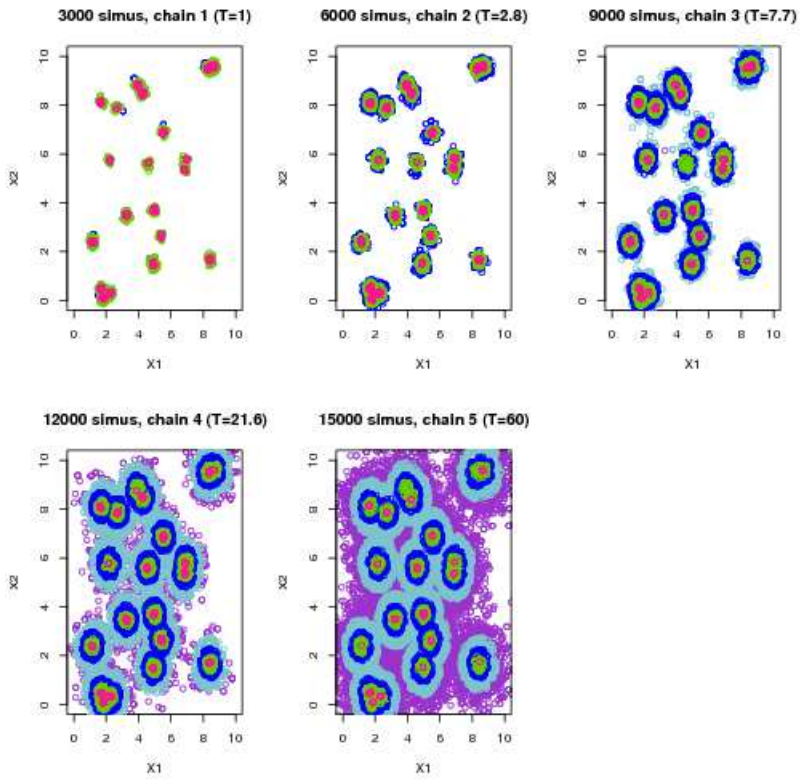

Figure 3: Simulations for chains 1 to 5 obtained by one run of the EES. The colors correspond to the five energy levels.

Figure 4: WebLogo corresponding to the product multinomial used to generate data.

	chain 1	chain 10	chain 20
chain 1	0.00	4.63	0.50
chain 2	16.32	4.33	0.62
chain 3	14.34	4.29	0.64
chain 4	11.98	4.64	0.70
chain 5	9.96	4.89	0.76
chain 6	8.26	5.46	1.03
chain 7	6.57	5.76	1.17
chain 8	6.01	6.26	1.50
chain 9	4.96	6.74	2.03
chain 10	4.32	0.00	2.30
chain 11	3.25	7.11	3.13
chain 12	2.85	6.67	4.33
chain 13	2.42	6.65	5.61
chain 14	1.98	6.11	7.38
chain 15	1.61	5.76	8.75
chain 16	1.44	5.13	10.64
chain 17	1.15	4.64	13.72
chain 18	1.08	4.32	16.09
chain 19	0.87	3.63	19.10
chain 20	0.62	2.99	0.00

Table 6: Repartition (in %) of accepted equi-energy moves between chain 1 and other possible chains (mean on 100 runs of PTEEM). Idem for chains 10 and 20.

	$(-\infty, 197.3)$	$[197.3, 216.3)$	$[216.3, 237.2)$	$[237.2, 260)$	$[260, +\infty)$
chain 1	9602	396	2	0	0
chain 4	4487	5343	170	0	0
chain 8	225	6123	2863	768	21
chain 10	5	990	3528	5017	460
chain 12	0	50	1047	6662	2241
chain 16	0	0	5	2266	7729
chain 20	0	0	0	312	9688

Table 7: Distribution in the energy rings of states from 10000 iterations, for one run of PTEEM and for chains 1, 4, 8, 10 12, 16 and 20.

	mean	standard deviation	min	max
PT	645.04	13.52	610	683
PTEEM	666.52	9.23	641	692

SimuMH

Table 8: Means, standard deviations, minimal and maximal values of the number of visited modes, on 100 runs of PT and PTEEM.

.

	chain 1	chain 10	chain 20
chain 1	0.00	0.02	0.00
chain 2	63.65	0.10	0.00
chain 3	23.90	0.32	0.00
chain 4	7.75	0.87	0.00
chain 5	2.78	1.77	0.00
chain 6	1.12	3.30	0.00
chain 7	0.47	6.45	0.00
chain 8	0.23	12.65	0.01
chain 9	0.07	22.44	0.05
chain 10	0.02	0.00	0.23
chain 11	0.00	21.39	0.67
chain 12	0.00	13.92	1.52
chain 13	0.00	7.99	3.12
chain 14	0.00	4.29	5.46
chain 15	0.00	2.12	8.56
chain 16	0.00	1.11	12.18
chain 17	0.00	0.61	16.58
chain 18	0.00	0.34	22.37
chain 19	0.00	0.19	29.26
chain 20	0.00	0.13	0.00

Table 9: Proportions (%) of accepted equi-energy moves between chain 1 and other possible chains (mean on 100 runs of PTEEM). Idem for chains 10 and 20.

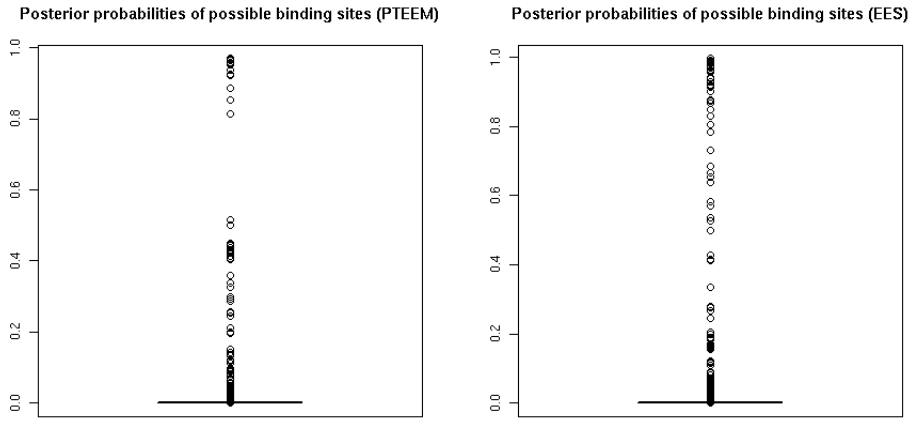

Figure 5: representing posterior probabilities $P(a_i = 1 \mid S), i = 1, \ldots, L^*$, obtained during a run of PTEEM, and during a run of EES.