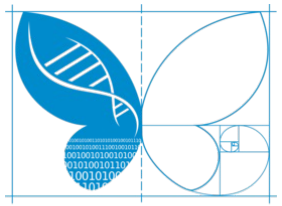
How is Hmsc fitted to data?

Overview on prior distributions and posterior sampling.

Gleb Tikhonov

JSS 2022
17 August 2022

UNIVERSITY OF HELSINKI
FACULTY OF BIOLOGICAL AND ENVIRONMENTAL SCIENCES



Outline

- Fundamentals of Bayesian statistics:
  - Likelihood, prior and posterior
  - Markov Chain Monte Carlo (MCMC) sampling
- Gibbs sampling
  - Conditional conjugacy
  - Block updaters
- MCMC sampling options in HMSC
- Customizable priors in HMSC
- Code demonstration

Likelihood

A **statistical model** is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data.

Ultimately, statistical model characterizes the probability of different data being generated from it: $p(Y_1)$, $p(Y_2)$, …, so that $\sum_i p(Y_i) = 1$

Statistical models often can be encompassed into families of same mathematical structure, which are parametrized by a set of **model parameters** $p(Y) = p(Y|\theta)$

The **likelihood** (or likelihood function) describes the joint probability of the observed data (evidence) as a function of the parameters of the chosen statistical model $f(\theta) = p(Y|\theta)$

Hierarchical model of species communities

$$y_{ij} \sim D_j\left(L_{ij}, \sigma_j^2\right)$$

Generalized modeling approach for various types of data

$$L_{ij} = L_{ij}^F + \sum_{r=1}^{n_r} L_{p_r(i)j}^r$$

Latent variable is a sum of fixed effects part and random effects at each level of sampling design

$$L_{ij}^F = \sum_{k=1}^{n_c} x_{ik}\beta_{kj}$$

Fixed effects part modeled with linear regression.

$$\beta_{\cdot j} \sim N\left(\mu_{\cdot j}, V\right), \quad \mu_{kj} = \sum_{l=1}^{n_t} t_{jl}\gamma_{kl}$$

$$\boldsymbol{\beta}^* = \left[\beta_{1\cdot}, \ldots, \beta_{n_c\cdot}\right]^T \sim N(\boldsymbol{\mu}^*, \Theta)$$

$$\Theta = V \otimes \left[\rho \mathrm{C} + \left(1 - \rho I_{n_s}\right)\right]$$

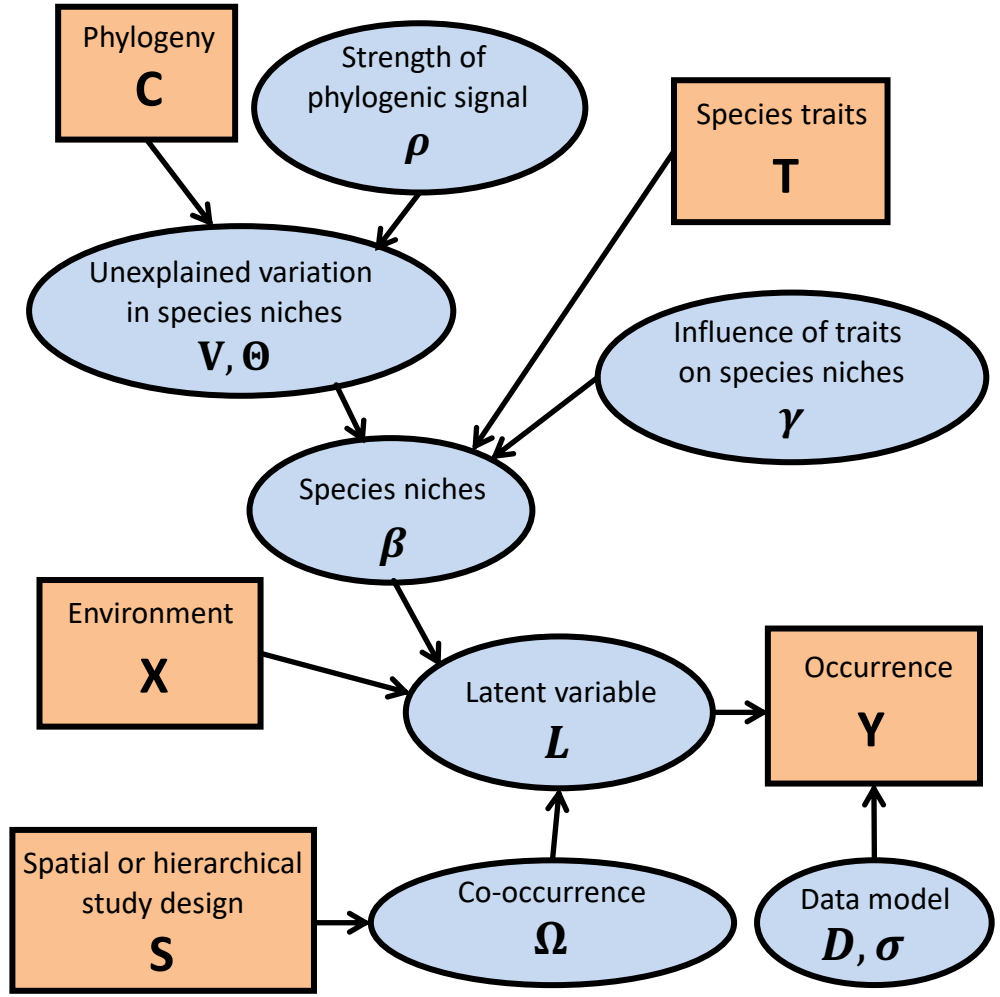
Linear regression coefficients further modeled w.r.t. available species trait information and phylogenic relationships

$$L_{qj}^r = \sum_{h=1}^{n_f^r} \eta_{qh}^r \lambda_{hj}^r$$

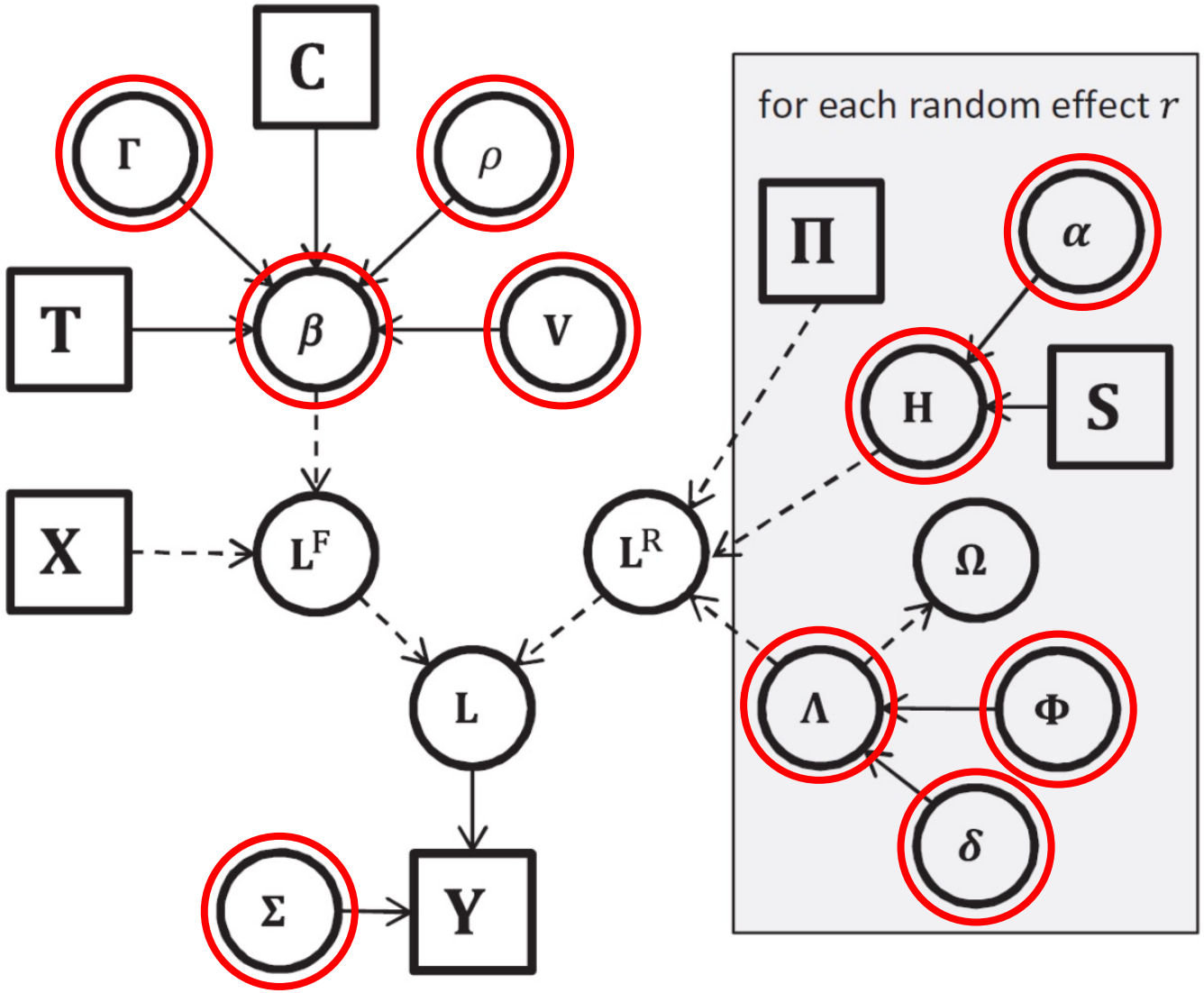
Random effects are modeled via latent factor models

$$\eta_{\cdot h}^r \sim N\left(0, \mathrm{K}_{S^r S^r}^{(rh)}\right)$$

Gaussian process priors for spatial/temporal latent factors



Hierarchical model of species communities



Likelihood of HMSC

$$f(\boldsymbol{\theta}) = p(Y|\boldsymbol{\theta})$$

$$\boldsymbol{\theta} \stackrel{\text{def}}{=} \{\Gamma, \rho, \boldsymbol{\beta}, V, \Sigma, H, \boldsymbol{\alpha}, \Lambda, \Phi, \boldsymbol{\delta}\}$$

Priors and posteriors

Prior distribution of model parameters $p_0(\theta)$ attempts to express the modeler's beliefs about their plausibility for the given modelling task.

Posterior distribution expresses the plausibility of model parameters θ after observing particular data Y, assuming that it was generated under a statistical model $p(Y|\theta)$. It is formally expressed via **Bayes rule** as being proportional to the product of likelihood and prior distribution:

$$p(\theta|Y) = \frac{p(Y|\theta)p_0(\theta)}{\int_\theta p(Y|\theta)p_0(\theta)d\theta}$$

In Bayesian paradigm **model fitting** means finding the posterior distribution or its approximation, which can be represented in some expressible manner.

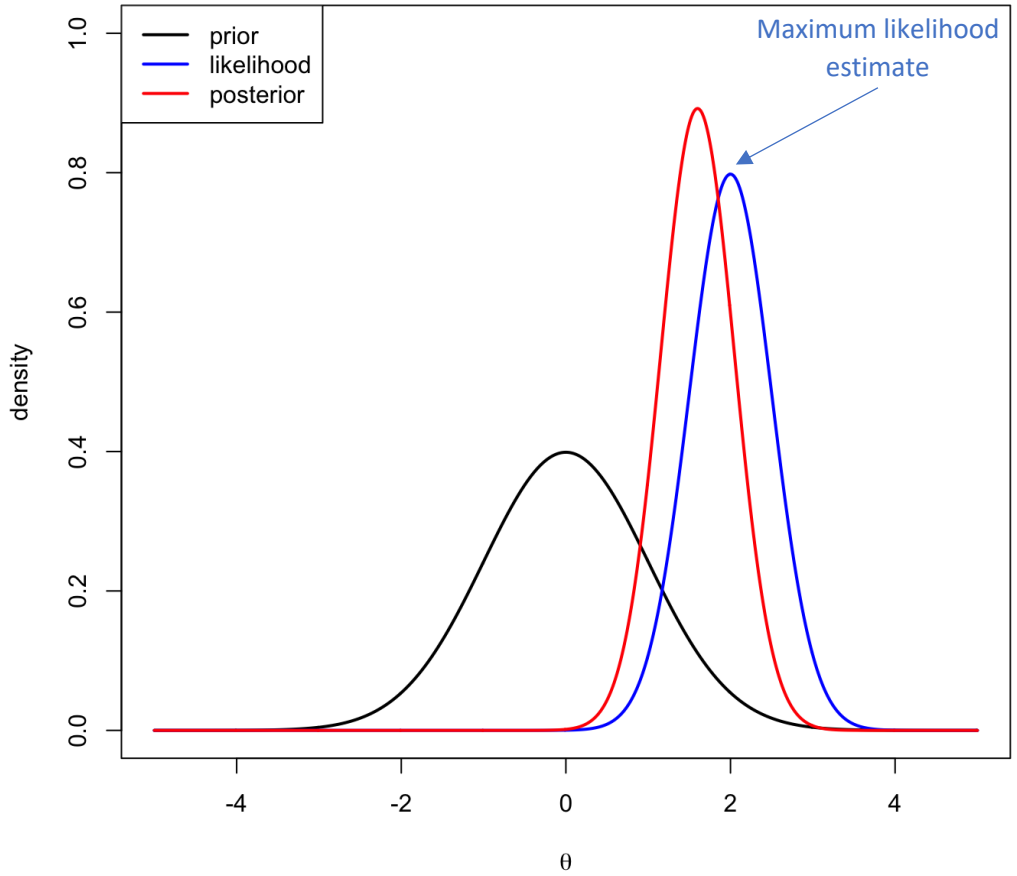
Typically, this means more than obtaining just a single point estimate (e.g. as in maximum likelihood).

Prior: $p_0(\theta) = N(\theta|0,1^2)$

Likelihood: $p(Y|\theta) = N(Y|\theta, 0.5^2)$

Observation: $Y = 2$

Posterior: ???



Markov Chain Monte Carlo (MCMC)

How-to conduct the analytical integration in the denominator of $p(\theta|Y) = \frac{p(Y|\theta)p_0(\theta)}{\int_\theta p(Y|\theta)p_0(\theta)d\theta}$ is known only for very few (and typically simple) problems.

E.g. $p(\theta|Y) = N(\theta|\mu, \sigma^2)$, where $\mu = \sigma^2 \frac{2}{0.5^2}, \sigma^{-2} = 0.5^{-2} + 1^{-2}$

Characterize $p(\theta|Y)$ by providing N samples from this distribution. Unfortunately, direct Monte Carlo sampling is equally complex as denominator integration.

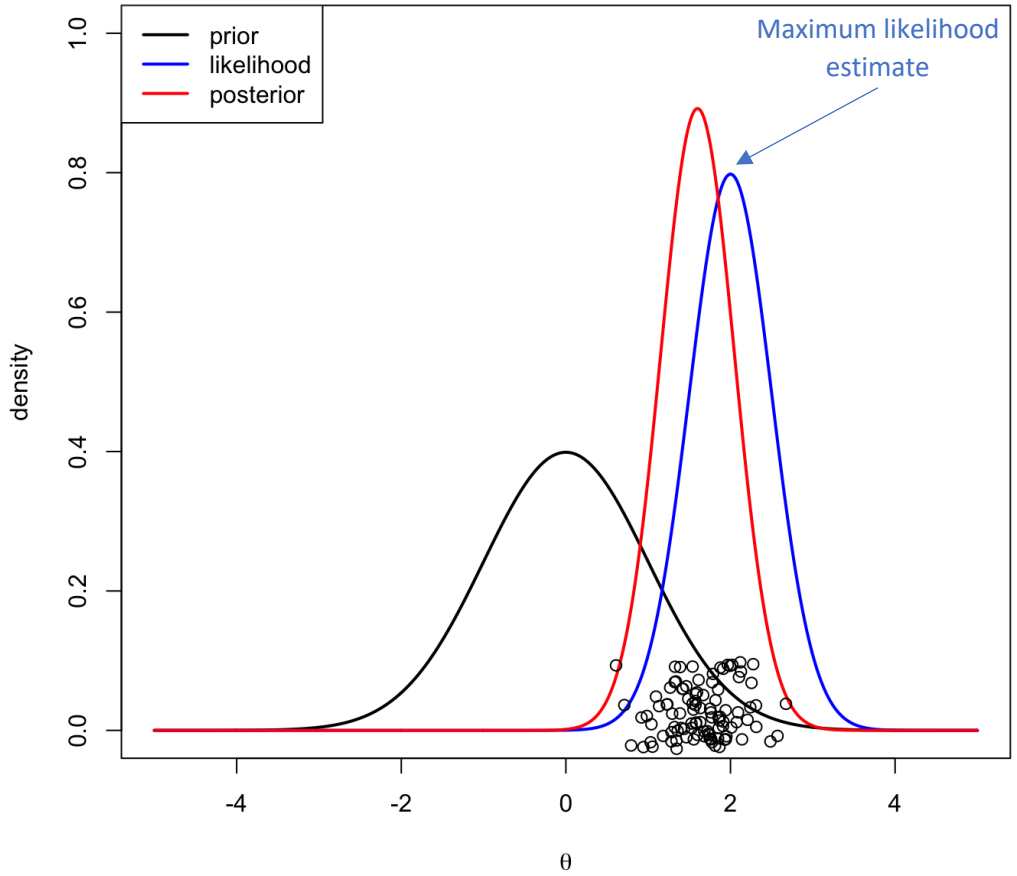
MCMC can be used to resolve this issue. Sequentially acquiring draws based on $p(Y|\theta)p_0(\theta)$ only into chains of MCMC samples.

Asymptotic guarantees. Autocorrelation – thinning. Starting position – transient.

Many different algorithms exist, constant field of development in Bayesian statistics. Hmsc uses **Gibbs sampling**.

Good visualization of various MCMC algorithms:
https://chi-feng.github.io/mcmc-demo/app.html

Prior:	$p_0(\theta) = N(\theta|0,1^2)$
Likelihood:	$p(Y|\theta) = N(Y|\theta, 0.5^2)$
Observation:	$Y = 2$
Posterior:	???



Gibbs sampling

Basic concept – avoid solving a complex problem directly, but repeatedly solve much simpler problem.

It is done by sampling only a single scalar parameter at once $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_m]$

$$\forall i \in 1 \dots m, \qquad p(\theta_i|\boldsymbol{\theta}_{-i}, Y) = \frac{p(Y|\theta_i, \boldsymbol{\theta}_{-i})p_0(\theta_i, \boldsymbol{\theta}_{-i})}{\int_{\theta_i} p(Y|\theta_i, \boldsymbol{\theta}_{-i})p_0(\theta_i, \boldsymbol{\theta}_{-i})d\theta_i}$$

Unlike in the original $p(\theta|Y) = \frac{p(Y|\theta)p_0(\theta)}{\int_\theta p(Y|\theta)p_0(\theta)d\theta}$, the univariate **conditional distributions** $p(\theta_i|\boldsymbol{\theta}_{-i}, Y)$ feature only a 1D integration, and it can be computed much easier.

Sampling from conditional distributions is called **conditional update**.

Cycling conditional updates through all parameters sequentially yields valid MCMC.

Many popular software available e.g. BUGS, JAGS.

If $p(\theta_i|\boldsymbol{\theta}_{-i}, Y)$ can be derived analytically within some known family, there is no need for numerical integration.

One important special case is when $p(\theta_i|\boldsymbol{\theta}_{-i}, Y)$ is of the same family of distributions as $p_0(\theta_i, \boldsymbol{\theta}_{-i})$, which is denoted that conditional likelihood $p(Y|\theta_i, \boldsymbol{\theta}_{-i})$ is **conjugate** to the prior $p_0(\theta_i, \boldsymbol{\theta}_{-i})$.

Sometimes conditional sampling can be done jointly $p\left(\theta_i, \theta_j|\boldsymbol{\theta}_{-i,j}, Y\right)$. This is called **block-Gibbs** update, reduces MCMC autocorrelation, but $p\left(\theta_i, \theta_j|\boldsymbol{\theta}_{-i,j}, Y\right)$ is generally more challenging than $p(\theta_i|\boldsymbol{\theta}_{-i}, Y)$ and $p\left(\theta_j|\boldsymbol{\theta}_{-j}, Y\right)$.

Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0, \sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2, \sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2, \sigma), \omega(x_2, \sigma)\big)$$

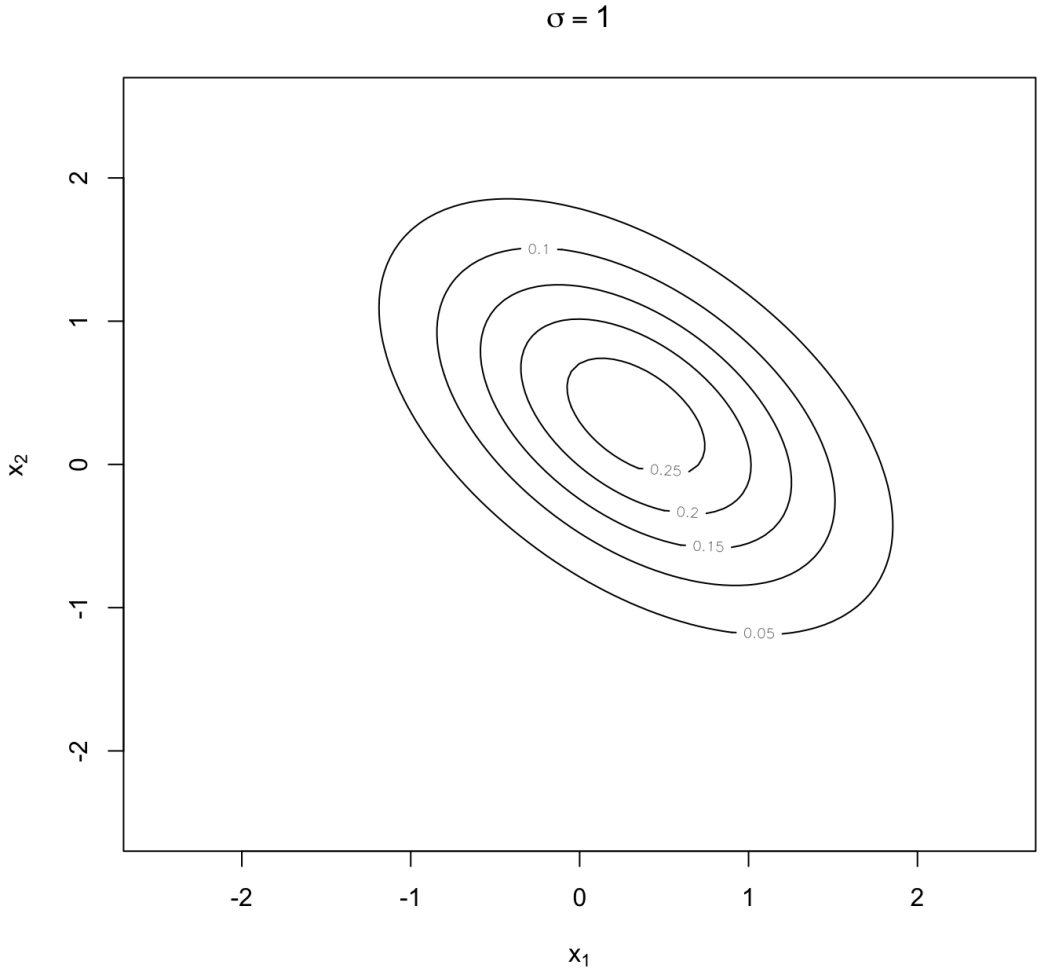
$$\mu(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}$$

$$p(x_2|x_1, \sigma^2) = N\big(x_2|\mu(x_1, \sigma), \omega(x_1, \sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and thinning.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0, \sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2, \sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2, \sigma), \omega(x_2, \sigma)\big)$$

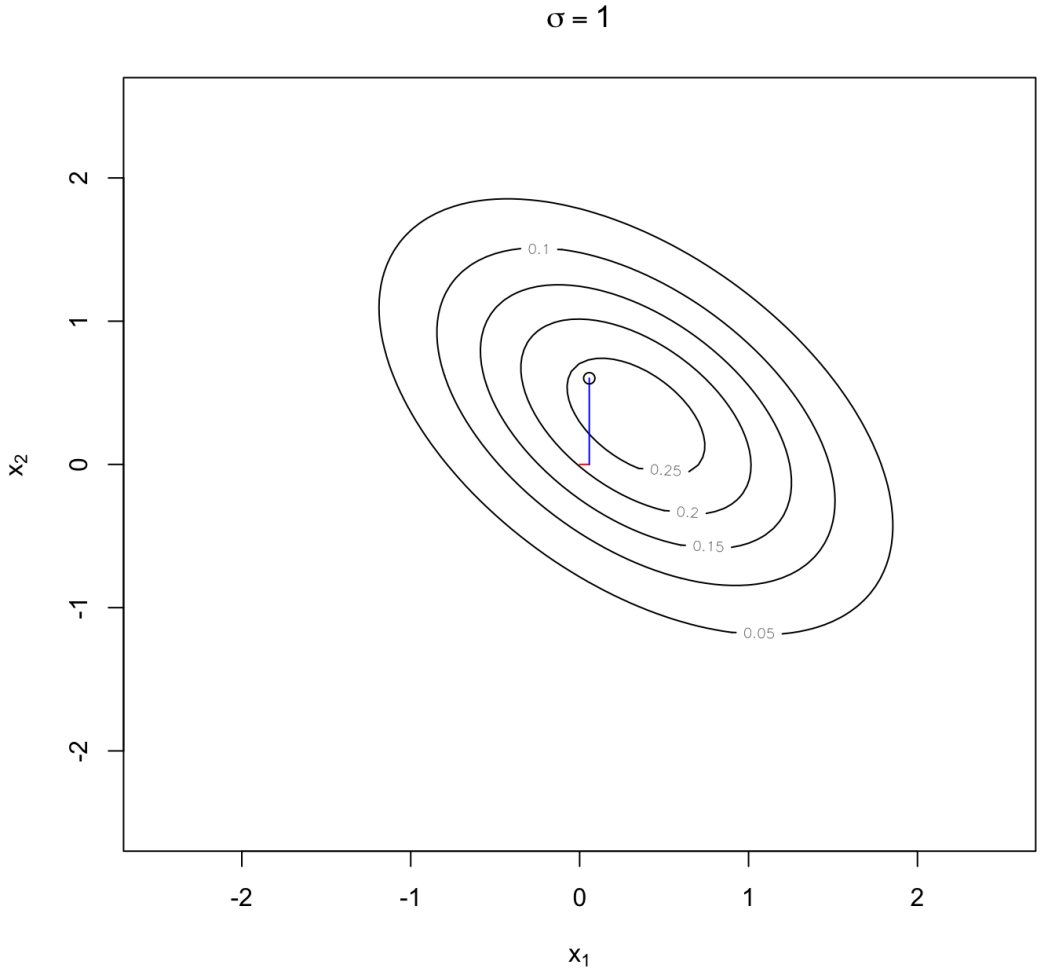
$$\mu(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}$$

$$p(x_2|x_1, \sigma^2) = N\big(x_2|\mu(x_1, \sigma), \omega(x_1, \sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and thinning.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0, \sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2, \sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2, \sigma), \omega(x_2, \sigma)\big)$$

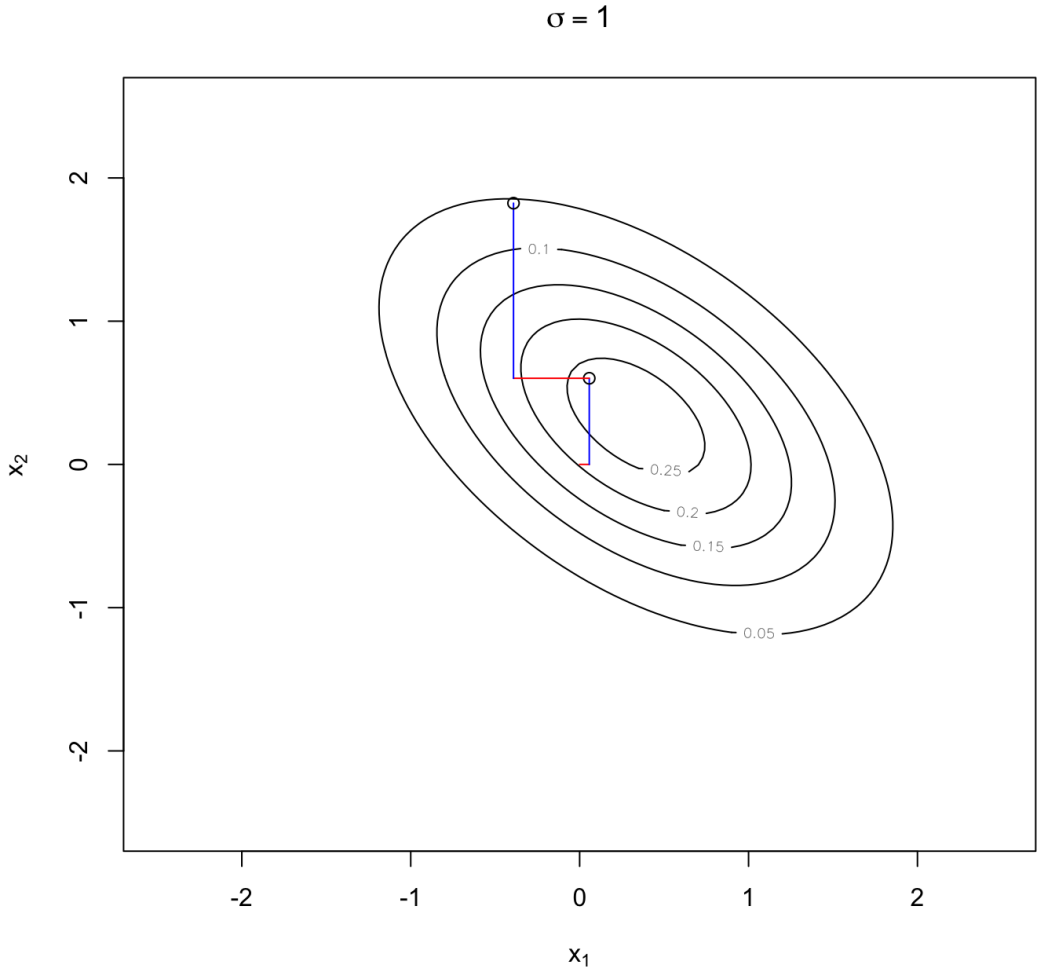
$$\mu(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2, \sigma) = (\sigma^{-2} + 1)^{-1}$$

$$p(x_2|x_1, \sigma^2) = N\big(x_2|\mu(x_1, \sigma), \omega(x_1, \sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and thinning.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0,\sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2,\sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2,\sigma), \omega(x_2,\sigma)\big)$$

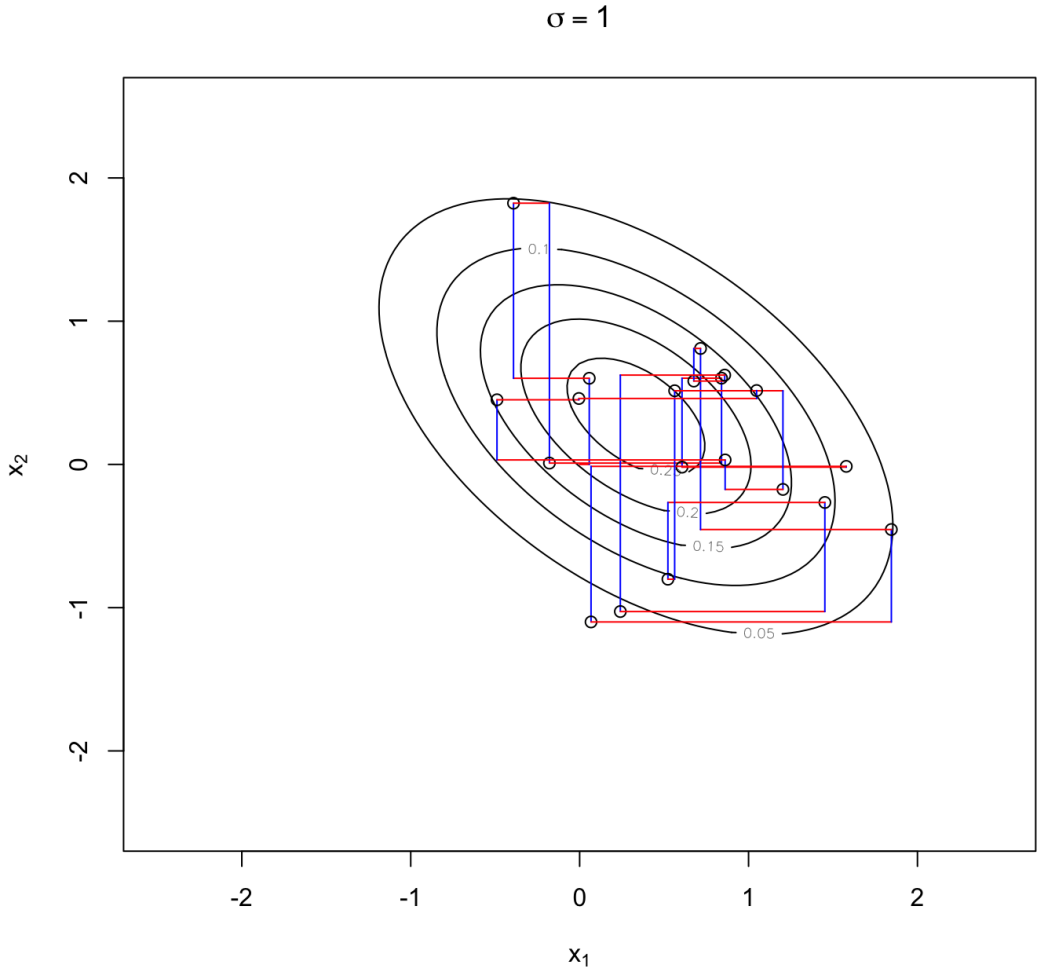
$$\mu(x_2,\sigma) = (\sigma^{-2}+1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2,\sigma) = (\sigma^{-2}+1)^{-1}$$

$$p(x_2|x_1,\sigma^2) = N\big(x_2|\mu(x_1,\sigma), \omega(x_1,\sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and thinning.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0,\sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2,\sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2,\sigma), \omega(x_2,\sigma)\big)$$

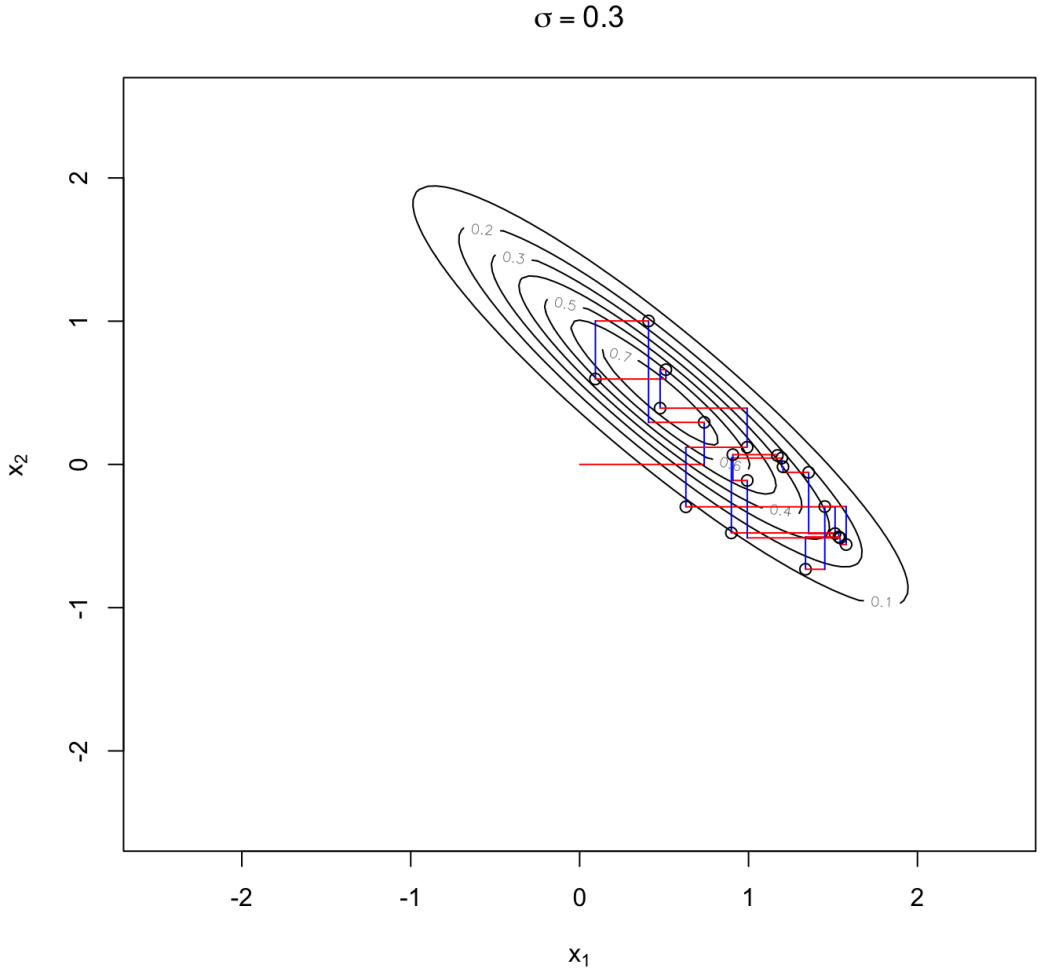
$$\mu(x_2,\sigma) = (\sigma^{-2}+1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2,\sigma) = (\sigma^{-2}+1)^{-1}$$

$$p(x_2|x_1,\sigma^2) = N\big(x_2|\mu(x_1,\sigma), \omega(x_1,\sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and **thinning**. Initial mismatch is addressed by **transient** phase.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0,\sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2,\sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2,\sigma), \omega(x_2,\sigma)\big)$$

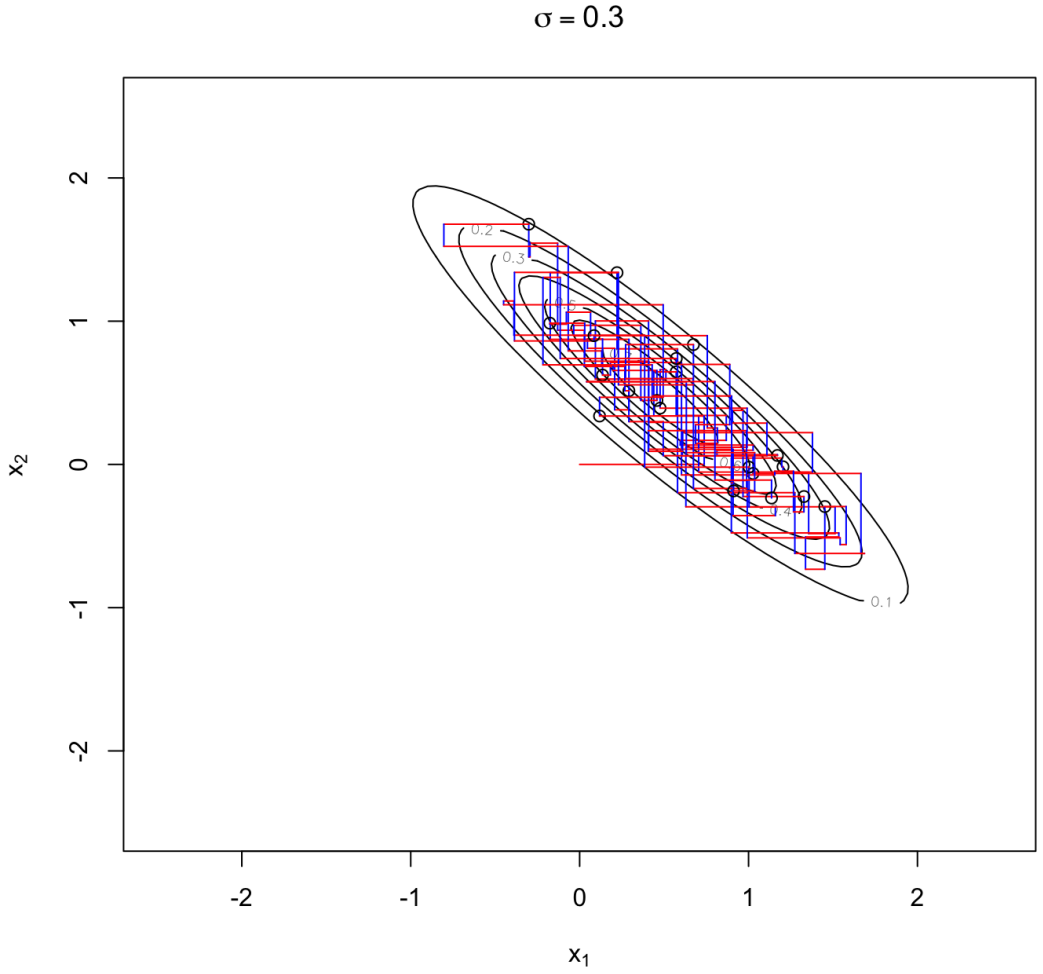
$$\mu(x_2,\sigma) = (\sigma^{-2}+1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2,\sigma) = (\sigma^{-2}+1)^{-1}$$

$$p(x_2|x_1,\sigma^2) = N\big(x_2|\mu(x_1,\sigma), \omega(x_1,\sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and **thinning**. Initial mismatch is addressed by **transient** phase.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0,\sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2,\sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2,\sigma), \omega(x_2,\sigma)\big)$$

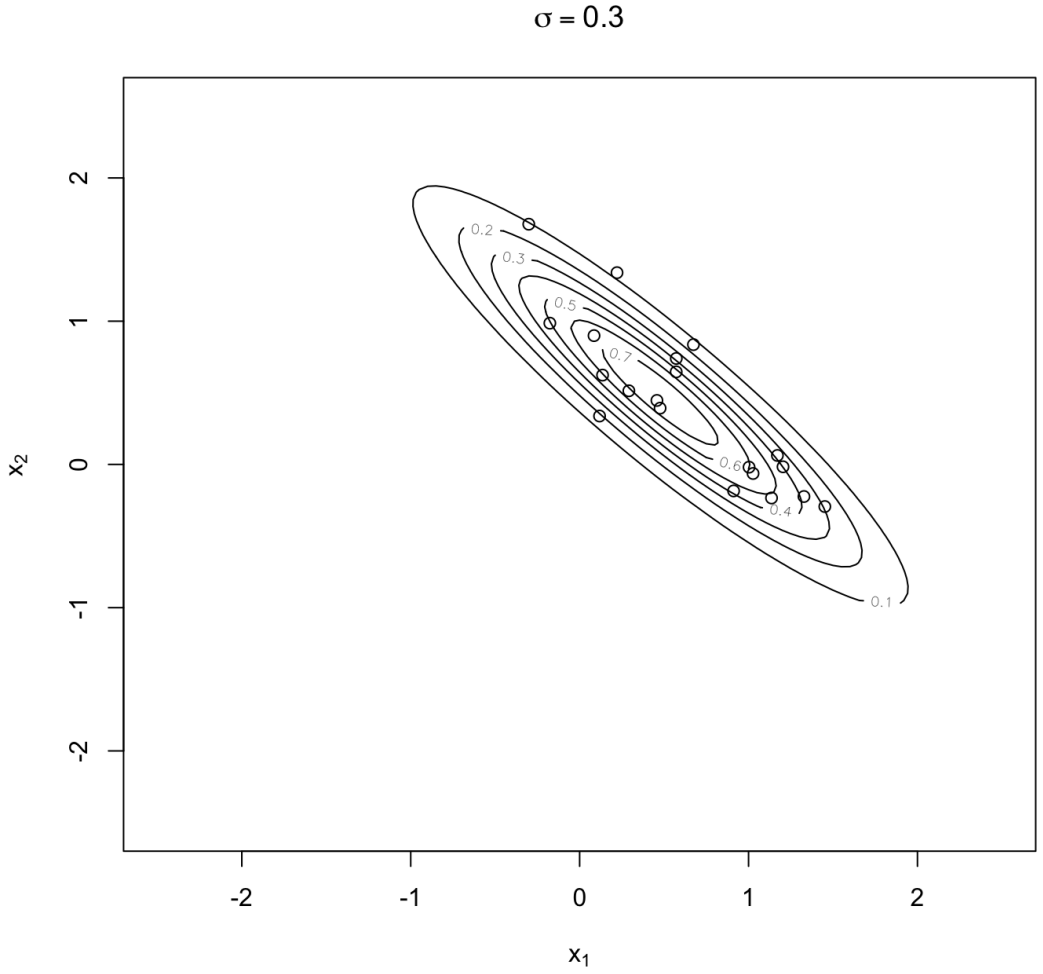
$$\mu(x_2,\sigma) = (\sigma^{-2}+1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2,\sigma) = (\sigma^{-2}+1)^{-1}$$

$$p(x_2|x_1,\sigma^2) = N\big(x_2|\mu(x_1,\sigma), \omega(x_1,\sigma)\big)$$

If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and **thinning**. Initial mismatch is addressed by **transient** phase.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



Gibbs sampling example

Toy statistical model with $\boldsymbol{\theta} = [x_1, x_2, \sigma]$:

$$y = x_1 + x_2 + \varepsilon$$

$$x_1 \sim N(0,1^2), \qquad x_2 \sim N(0,1^2), \qquad \varepsilon \sim N(0,\sigma^2)$$

Observed data $y = 1$

Conditional updaters for x_1 and x_2 have closed form:

$$p(x_1|x_2,\sigma^2) \propto N(y|x_1 + x_2, \sigma^2)N(x_1|0,1)$$
$$= N\big(x_1|\mu(x_2,\sigma), \omega(x_2,\sigma)\big)$$

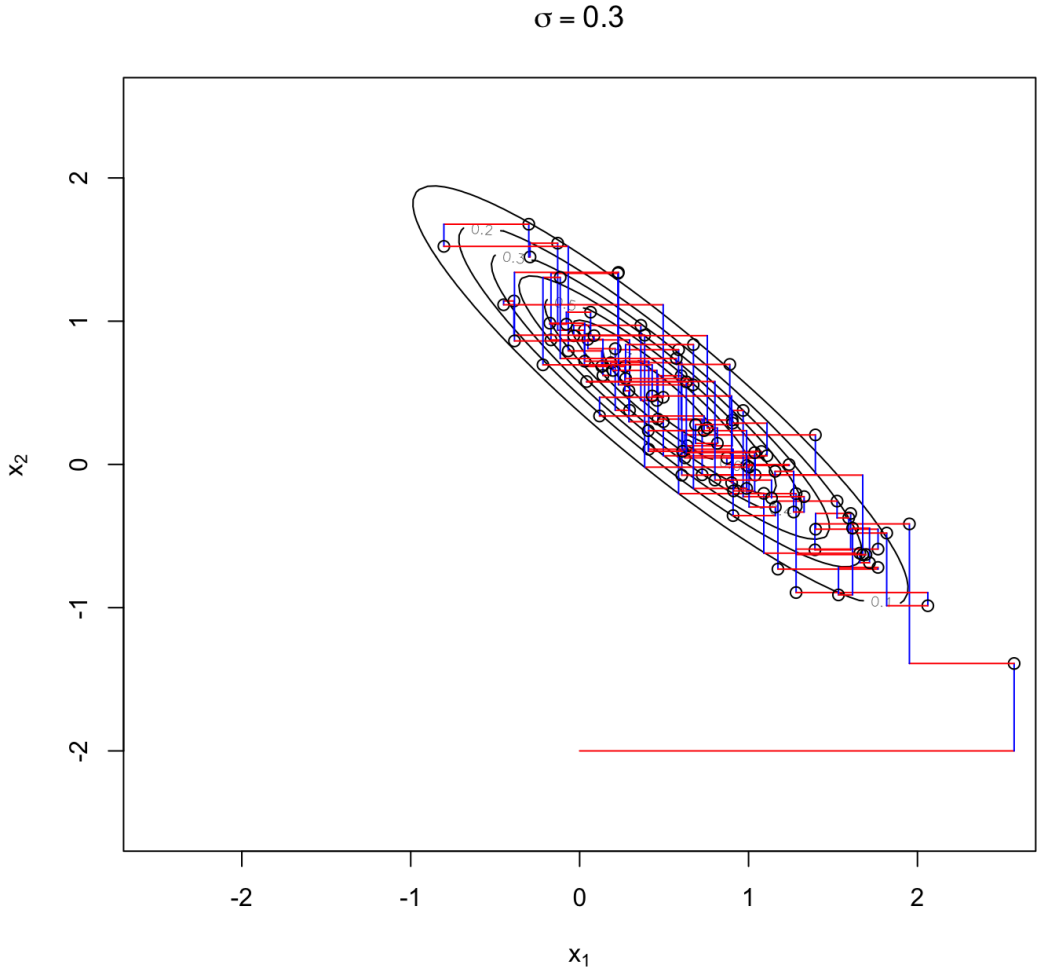
$$\mu(x_2,\sigma) = (\sigma^{-2}+1)^{-1}\sigma^{-2}(y - x_2), \qquad \omega(x_2,\sigma) = (\sigma^{-2}+1)^{-1}$$

$$p(x_2|x_1,\sigma^2) = N\big(x_2|\mu(x_1,\sigma), \omega(x_1,\sigma)\big)$$

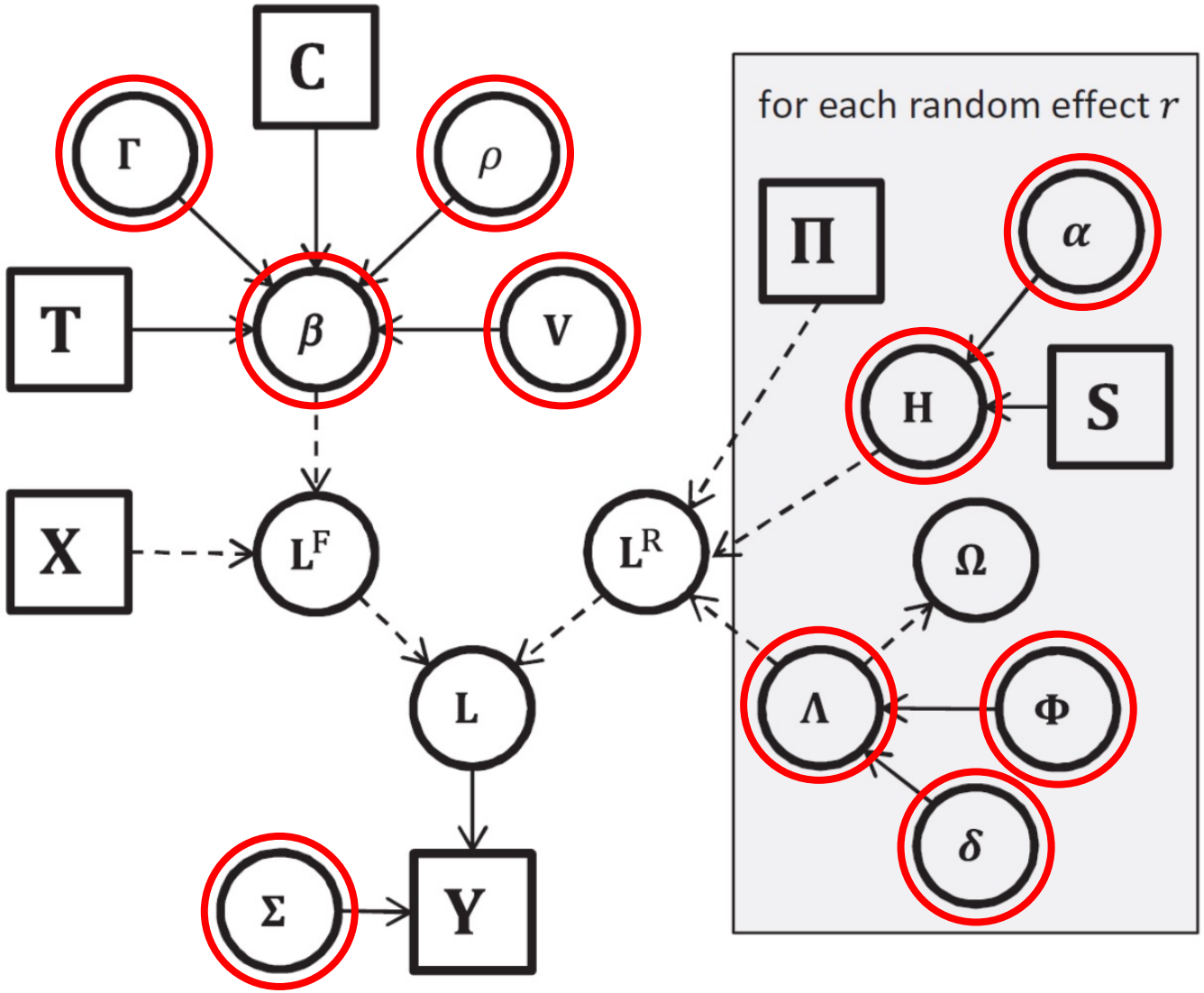
If σ is small, then magnitude of conditional updaters is small compared to the size of whole posterior. This causes autocorrelation in MCMC.

Autocorrelation is addressed by increasing number of iterations and **thinning**. Initial mismatch is addressed by **transient** phase.

Block updater $p(x_1, x_2|\sigma^2)$ completely removes the autocorrelation.



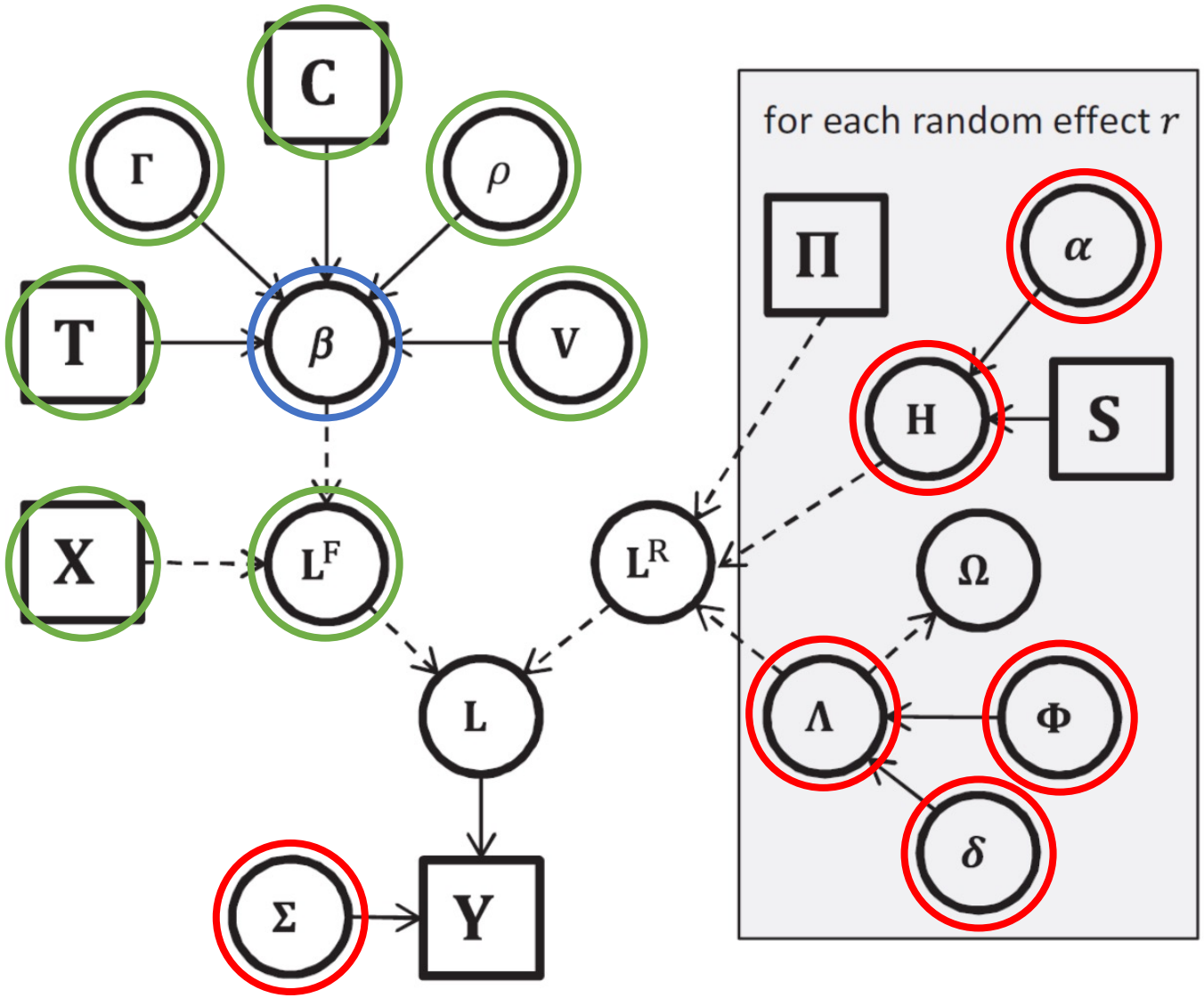
Gibbs sampling in HMSC



Get some random initialization

1. Update parameter β with others being fixed
2. Update parameter Γ with others being fixed
3. ….
4. ….
5. Update parameter Λ with others being fixed
6. Update parameter H with others being fixed
7. Move to step 1

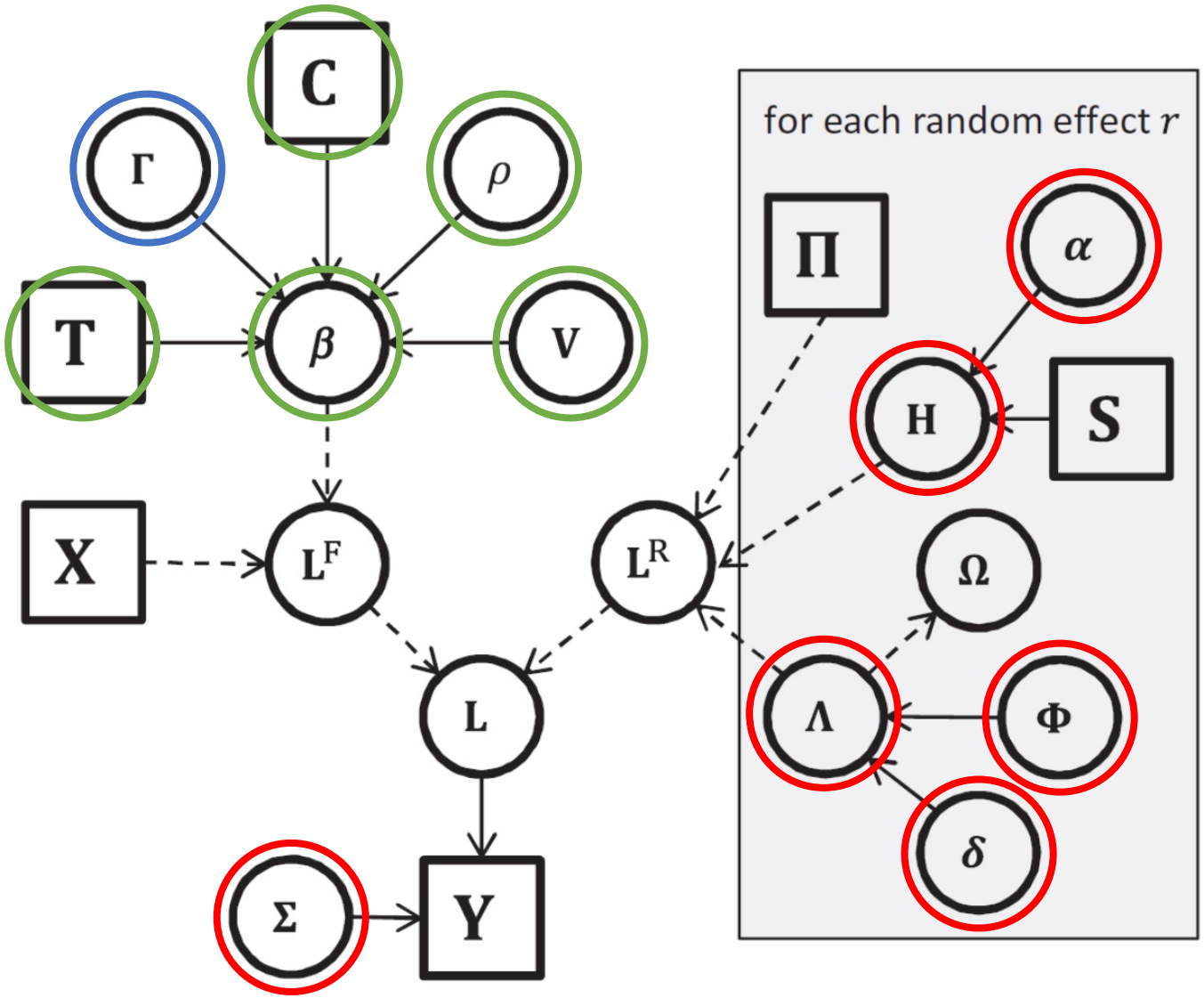
Gibbs sampling in HMSC



Get some random initialization

1. Update parameter β with others being fixed
2. Update parameter Γ with others being fixed
3. ….
4. ….
5. Update parameter Λ with others being fixed
6. Update parameter H with others being fixed
7. Move to step 1

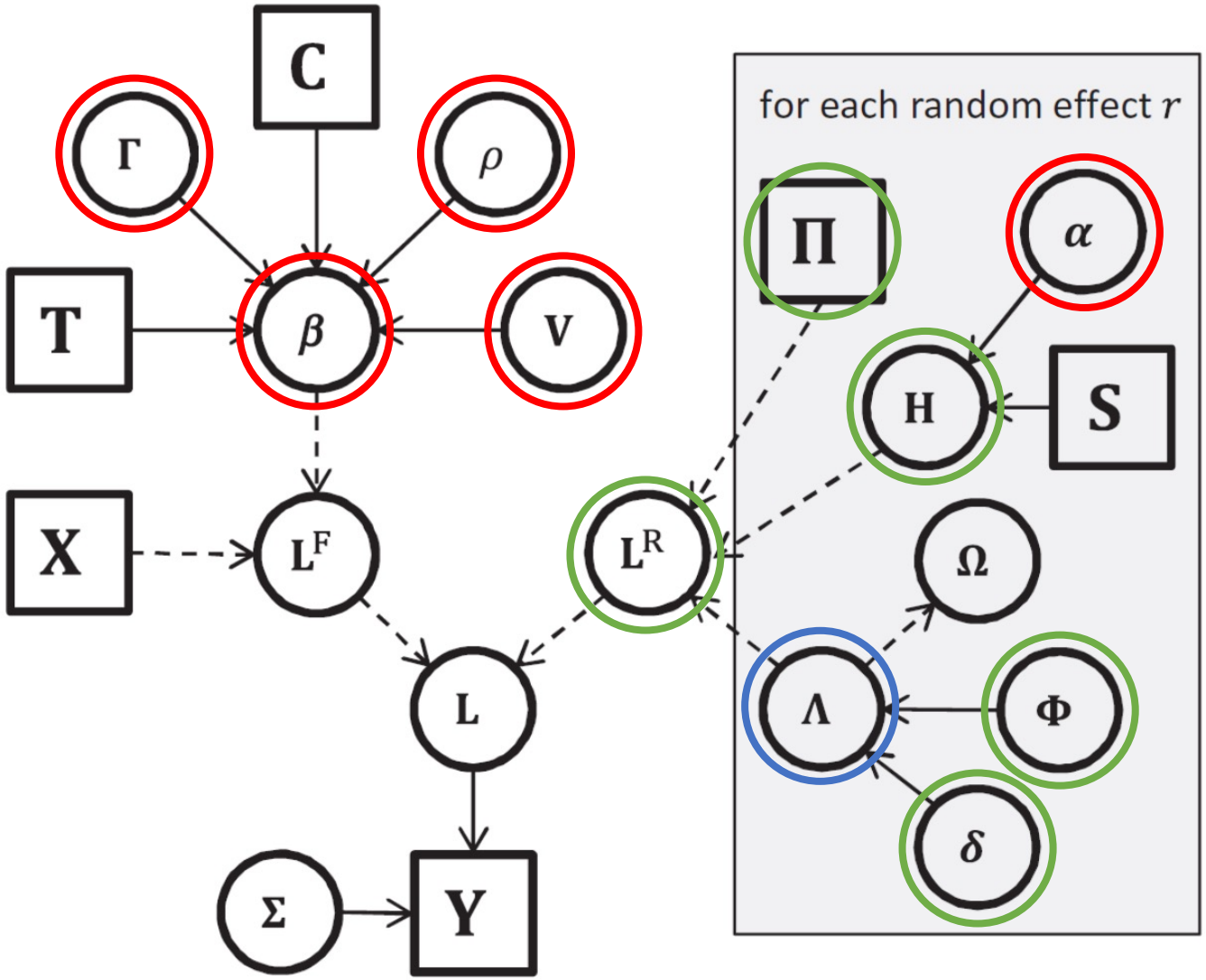
Gibbs sampling in HMSC



Get some random initialization

1. Update parameter β with others being fixed
2. Update parameter Γ with others being fixed
3. ….
4. ….
5. Update parameter Λ with others being fixed
6. Update parameter H with others being fixed
7. Move to step 1

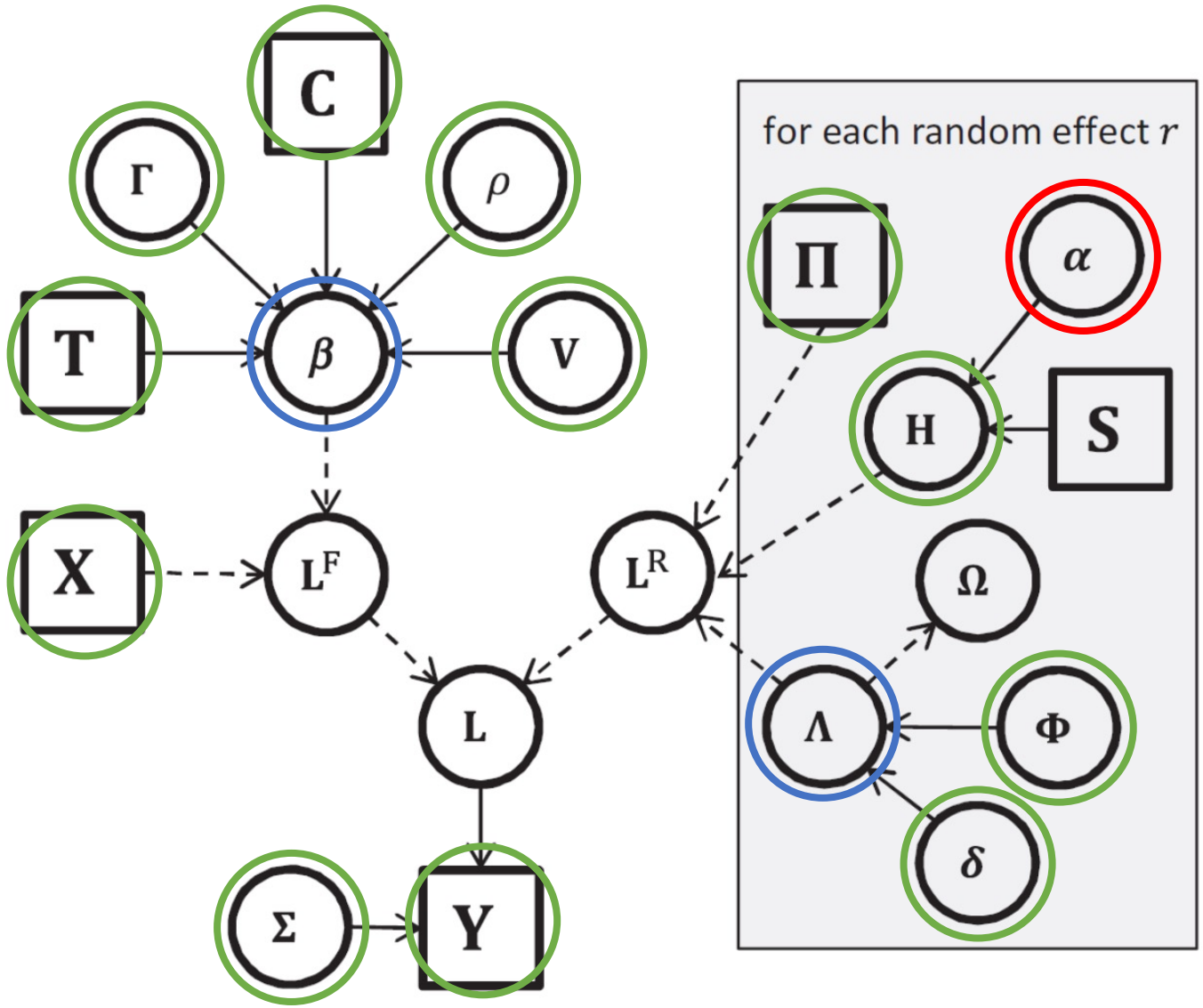
Gibbs sampling in HMSC



Get some random initialization

1. Update parameter β with others being fixed
2. Update parameter Γ with others being fixed
3. ….
4. ….
5. Update parameter Λ with others being fixed
6. Update parameter H with others being fixed
7. Move to step 1

Advanced Gibbs sampling in HMSC



For some of the HMSC components' combinations, it is possible to derive their joint conditional updates.

These enable to reduce autocorrelation in MCMC and thus run for shorter chains.

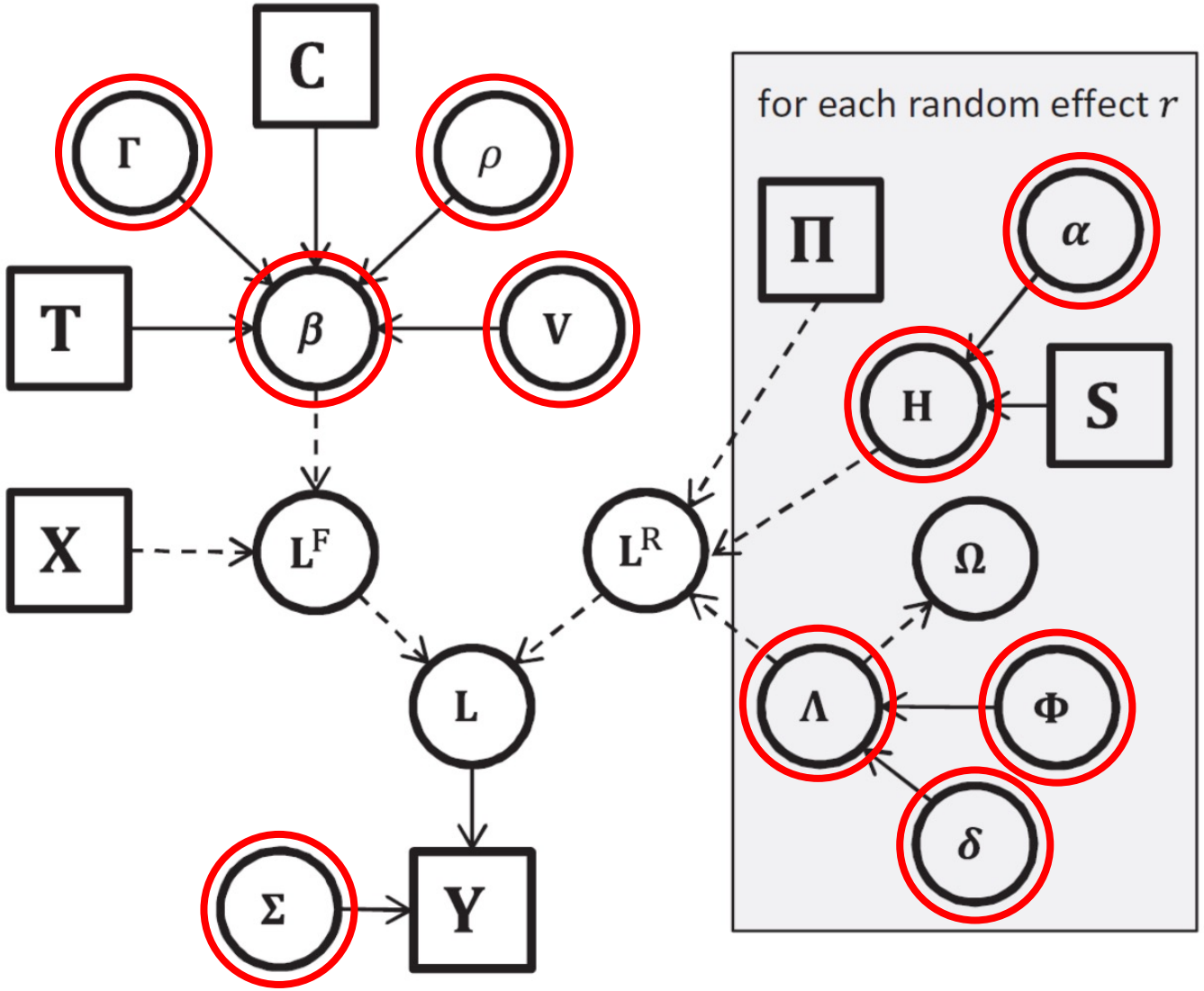
Derivation is much more complex than for unicomponental updaters.

Many of them are numerically more heavy than corresponding unicomponental updaters.

The trade-off between autocorrelation improvement and increased numerical cost is problem-specific.

Many of these are optional in Hmsc sampler, but generally recommended to stick with default set-up.

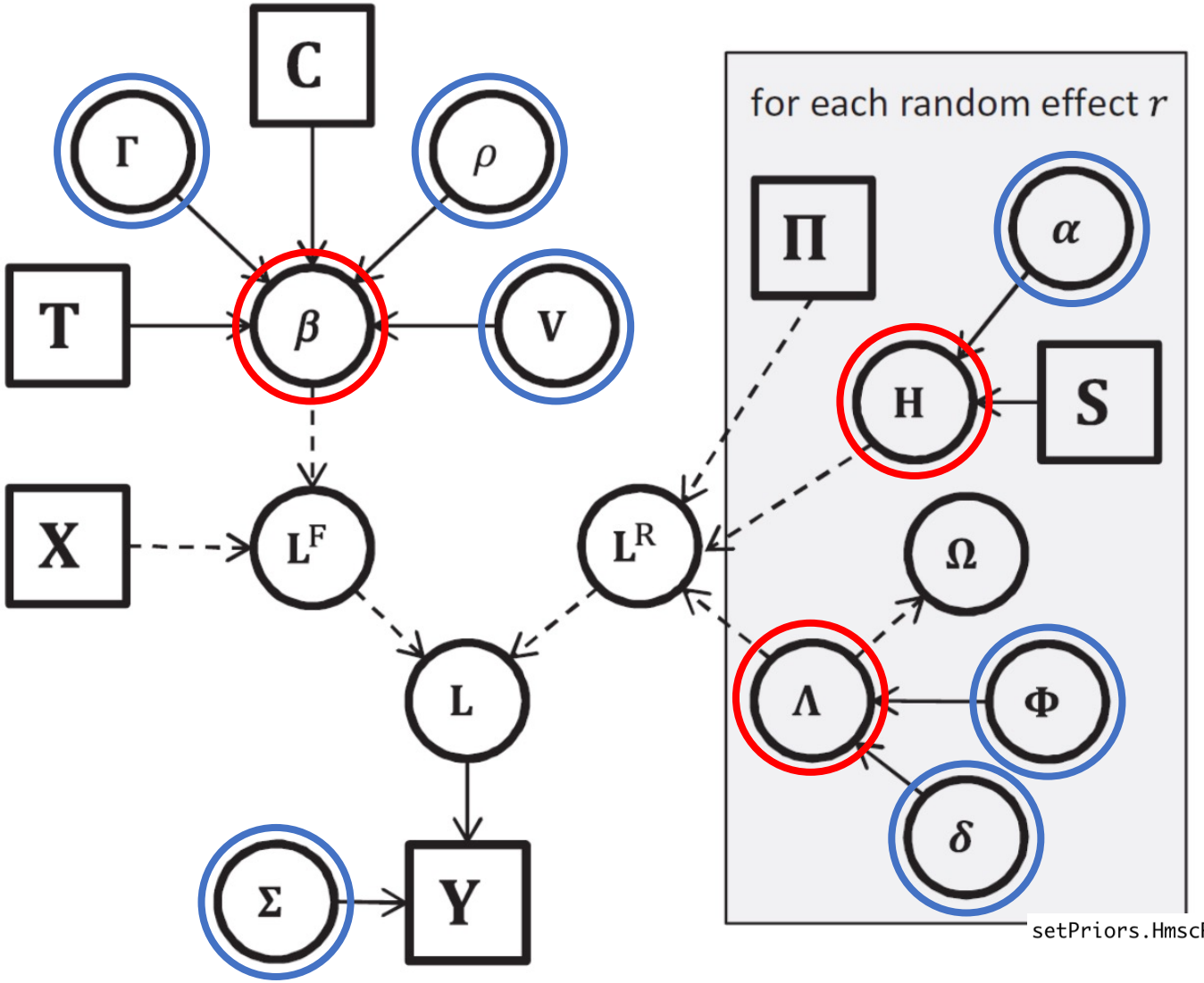
Options of Gibbs sampling in HMSC




```
sampleMcmc =
 function(hM, samples, transient=0, thin=1, initPar=NULL,
          verbose, adaptNf=rep(transient,hM$nr),
          nChains=1, nParallel=1,
          useSocket = TRUE,
          dataParList=NULL, updater=list(GammaEta=FALSE),
          fromPrior = FALSE, alignPost = TRUE)
```

- Number of samples
- Length of transient phase
- Thinning between recorded samples
- Start values for MCMC chains
- Number of chains
- How many processes to use
- What type of R parallelization
- Provide internal Hmsc preprocessed objects
- Which updaters not to use
- Sample from the prior only
- Try to align the order of latent loadings in the posterior

Customizable priors in Hmsc



All model parameters that are sources in the DAG are given priors, which have hyperparameters.

Priors defined for whole model:

- Tolerance to stochastic variation in species niches $V \sim IW(f_0, V_0)$
- Phylogenic strength ρ~grid prior on $[0,1]$
- Contribution of traits to species niches $\boldsymbol{\gamma} \sim N(\boldsymbol{\mu}_\gamma, \Sigma_\gamma)$
- Residual variation $\sigma_j \sim Ga^{-1}(a_\sigma, b_\sigma)$

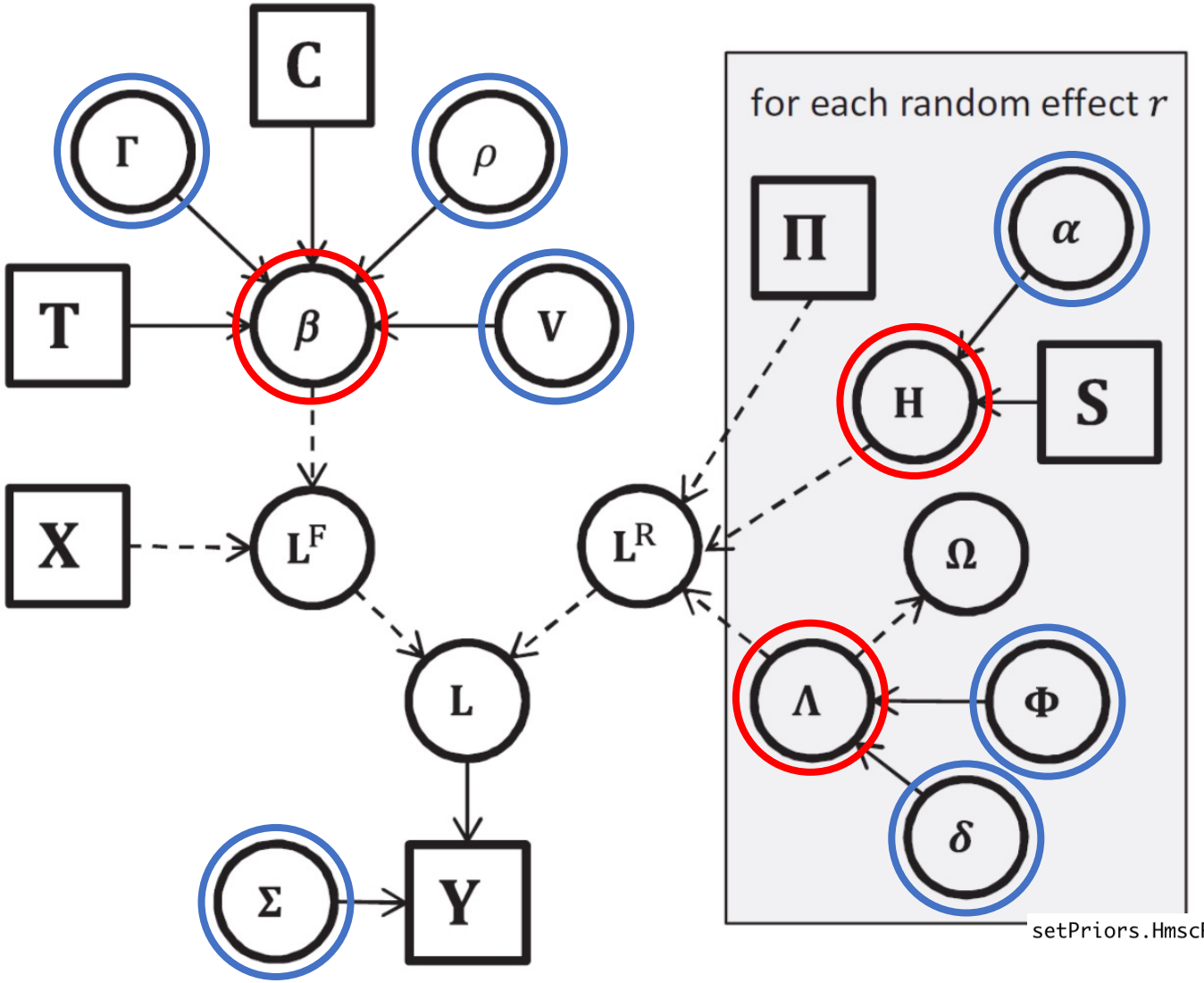
```
setPriors.Hmsc = function(hM, V0=NULL, f0=NULL, mGamma=NULL,
   UGamma=NULL, aSigma=NULL, bSigma=NULL, nuRRR=NULL, a1RRR=NULL,
   b1RRR=NULL, a2RRR=NULL, b2RRR=NULL, rhopw=NULL, setDefault=FALSE, ...)
```

Priors defined for each random level:

- Spatial covariance lengthscale α~grid prior on $[0, \alpha_+]$, α_+-max spatial distance
- Local shrinkage of latent loadings $\phi_{hj} \sim Ga\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$
- Progressive shrinkage of latent loadings $\delta_1 \sim Ga(a_1, b_1)$, $\delta_h \sim Ga(a_2, b_2)$

```
setPriors.HmscRandomLevel = function(rL, nu=NULL, a1=NULL, a2=NULL, b1=NULL, b2=NULL,
                                     alphapw=NULL, nfMax=NULL, nfMin=NULL, setDefault=FALSE, ...)
```

Customizable priors in Hmsc



All model parameters that are sources in the DAG are given priors, which have hyperparameters.

Priors defined for whole model:

- Tolerance to stochastic variation in species niches $V \sim IW(f_0, V_0)$
- Phylogenic strength ρ~grid prior on $[0,1]$
- Contribution of traits to species niches $\boldsymbol{\gamma} \sim N(\boldsymbol{\mu}_\gamma, \Sigma_\gamma)$
- Residual variation $\sigma_j \sim Ga^{-1}(a_\sigma, b_\sigma)$

```
setPriors.Hmsc = function(hM, V0=NULL, f0=NULL, mGamma=NULL,
   UGamma=NULL, aSigma=NULL, bSigma=NULL, nuRRR=NULL, a1RRR=NULL,
   b1RRR=NULL, a2RRR=NULL, b2RRR=NULL, rhopw=NULL, setDefault=FALSE, ...)
```

Priors defined for each random level:

- Spatial covariance lengthscale α~grid prior on $[0, \alpha_+]$, α_+-max spatial distance
- Local shrinkage of latent loadings $\phi_{hj} \sim Ga\left(\frac{\nu}{2}, \frac{\nu}{2}\right)$
- Progressive shrinkage of latent loadings $\delta_1 \sim Ga(a_1, b_1)$, $\delta_h \sim Ga(a_2, b_2)$

```
setPriors.HmscRandomLevel = function(rL, nu=NULL, a1=NULL, a2=NULL, b1=NULL, b2=NULL,
                                     alphapw=NULL, nfMax=NULL, nfMin=NULL, setDefault=FALSE, ...)
```