

# baesRCM Simulation Build

Quinton Neville

9/15/2022

## Contents

<b>1. Generate Some Simple Data</b>	<b>1</b>
<b>2. Apply bayesRCM package functions to generate posterior samples of the model</b>	<b>2</b>
2.1 Posterior Samples (Prelim Test)	2
2.2 Testing Individual Updates by Parameter	2

## 1. Generate Some Simple Data

Here we generate 10 volumes of multivariate normal data for 10 subjects in a network of 4 rois, with 2 true connections or edges in the associated group graph.

```
#Generate covariance structure for multivariate gaussian covariance matrices
volumes <- 10
subjects <- 10
rois <- 10 #keep it small to start testing
true_con <- 20 #true connections or no. of edges in G
#Group overall graph with true_con # of edges
set.seed(4)
G <-
  matrix(
    rbinom(volumes * subjects, 1, prob = true_con / (volumes * subjects)),
    nrow = volumes
  )
diag(G) <- 1
G <- Matrix::forceSymmetric(G, uplo = "U")
#G

#Overall Precision Matrix
set.seed(4)
Sigma_0 <-
  (G * matrix(rnorm((volumes * subjects), 0, 10), nrow = volumes)) |>
  Matrix::forceSymmetric(uplo = "U") |>
  (\(x) {as.matrix(Matrix::nearPD(x)$mat)})()
#Sigma_0

#Subject precision matrices based off of group
Sigma_k <- list()

#Fill matrices in list
for (n in 1:subjects) {
```

```

set.seed(n)
Sigma_k[[n]] <- (Sigma_0 + matrix(rnorm((volumes * subjects), 0.25/n, 1), nrow = volumes)) |>
Matrix::forceSymmetric(uplo = "U") |>
  (\(x) {as.matrix(Matrix::nearPD(x)$mat)})()
}

#List of each subject's array in time/by volume
#randomly generated around mean 0 with subject specific precision
#i.e. stead state
data_list <- list()

#Loop through subjects and volumes to generate data
for (n in 1:subjects) { #assumes no temporal mean trend, centered at 0
  set.seed(n)
  data_list[[n]] <- mvtnorm::rmvnorm(volumes, rep(0, rois), Sigma_k[[n]])
}

#Save 'truth' to evaluate recovery
Omega_0 <- solve(Sigma_0)
Omega_k <- map(Sigma_k, solve)

```

## 2. Apply bayesRCM package functions to generate posterior samples of the model

In development, one can source the helper functions from the `./R/` directory and the `.cpp` file from `/src/`. However, at this moment the package is functional and so one can download via github with `devtools::install_github("nevilleq/bayesRCM")` and then load the library like `normal library(bayesRCM)`.

### 2.1 Posterior Samples (Prelim Test)

Next, let's generate some preliminary posterior samples for 10 iterations and look at the results. There are still bugs we need to work out, especially in the graph update `log_H` and being able to do a cholesky decomp (i.e. enforcing symmetry and positive definite-ness).

```

result <- rcm(y = data_list, n_samples = 10)
#write_rds(result, "./results/prelim_test.RDS")
write_rds(result, "../results/prelim_test.RDS")

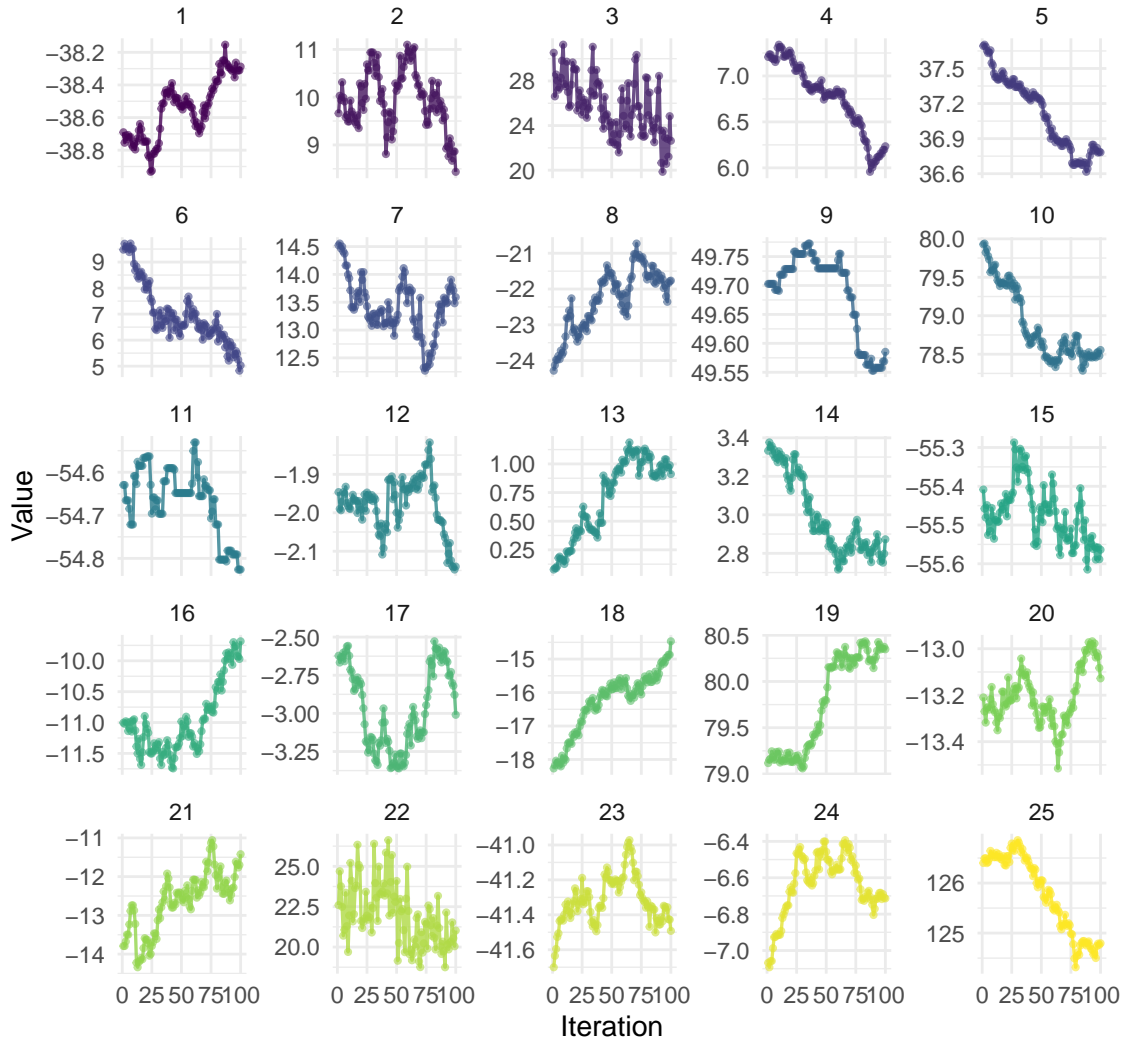
```

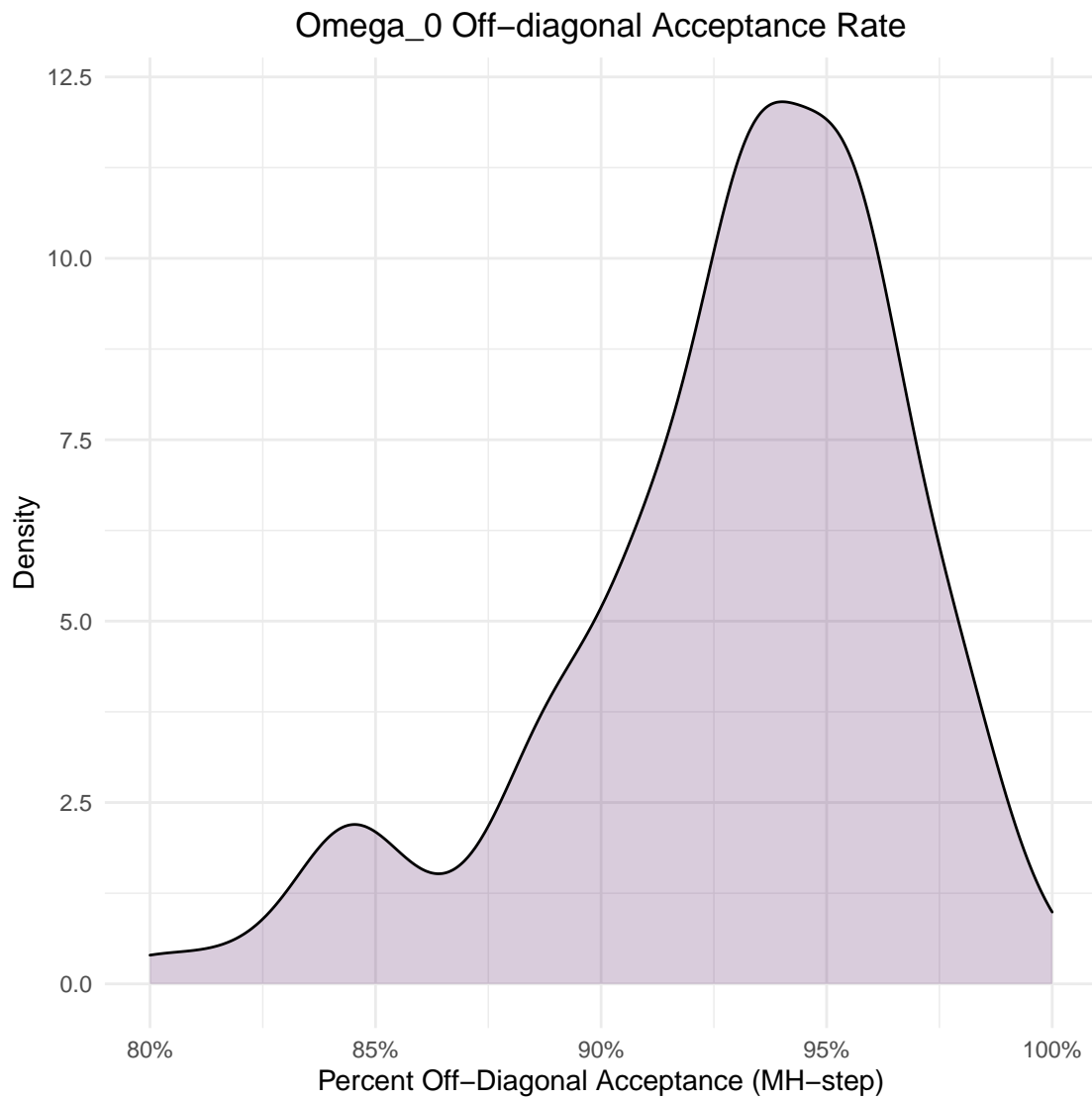
### 2.2 Testing Individual Updates by Parameter

Here, we fix all MCMC, graph updates, and/or non-direct sampling components of the algorithm, then observe the behaviour of the resulting Markov Chain(s). To do so, we are going to pull out the `rcm` source code, fix the desired elements at the “truth” (see above), and then sample the parameter(s) of interest. This should help troubleshoot and debug, especially for the  $G_k/\Omega_k$  update.

### 2.2.1 $\Omega_0$ with fixed $\tau_k$

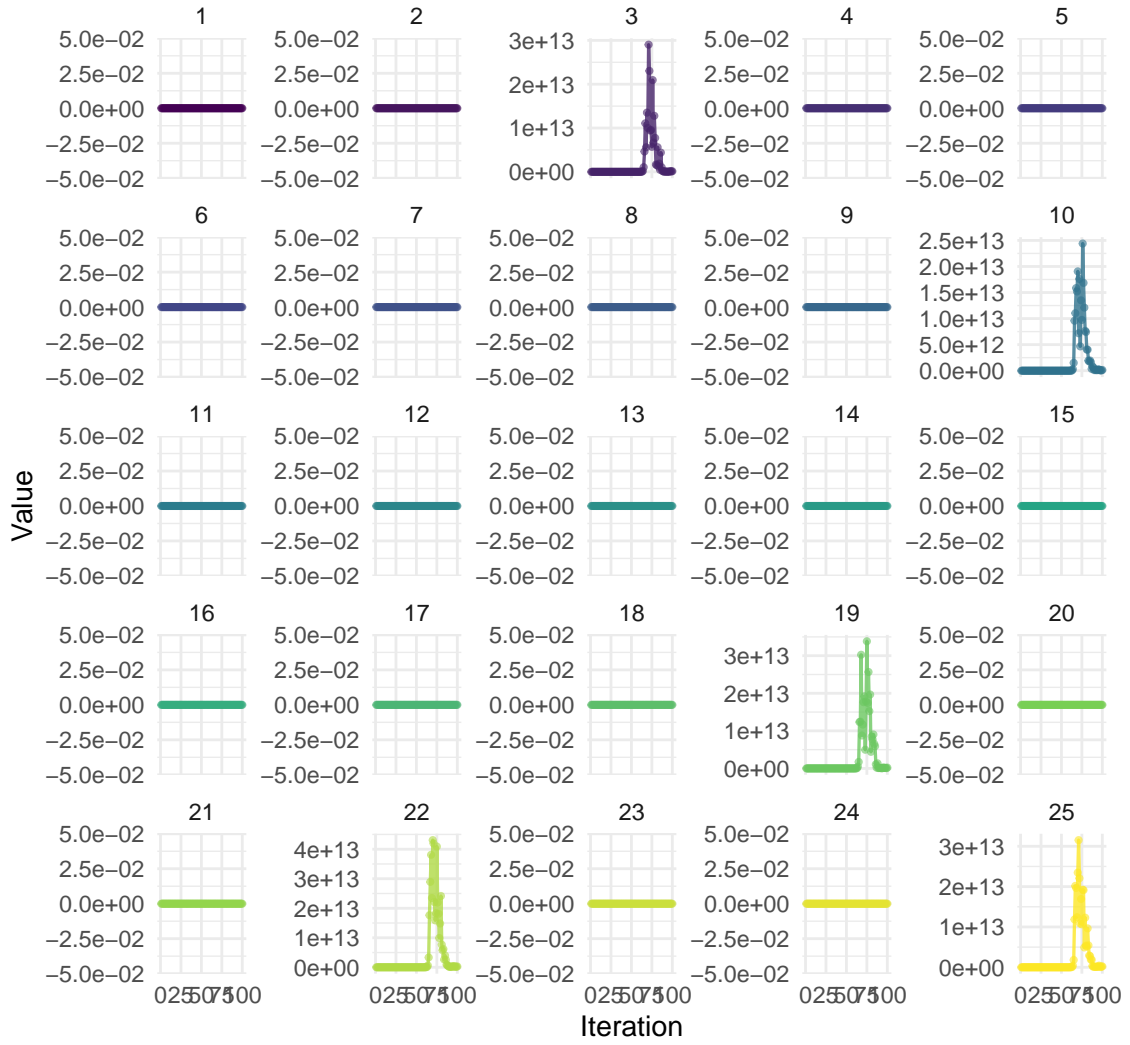
Omega\_0 Chain with Fixed tau\_k and G\_k/Omega\_k

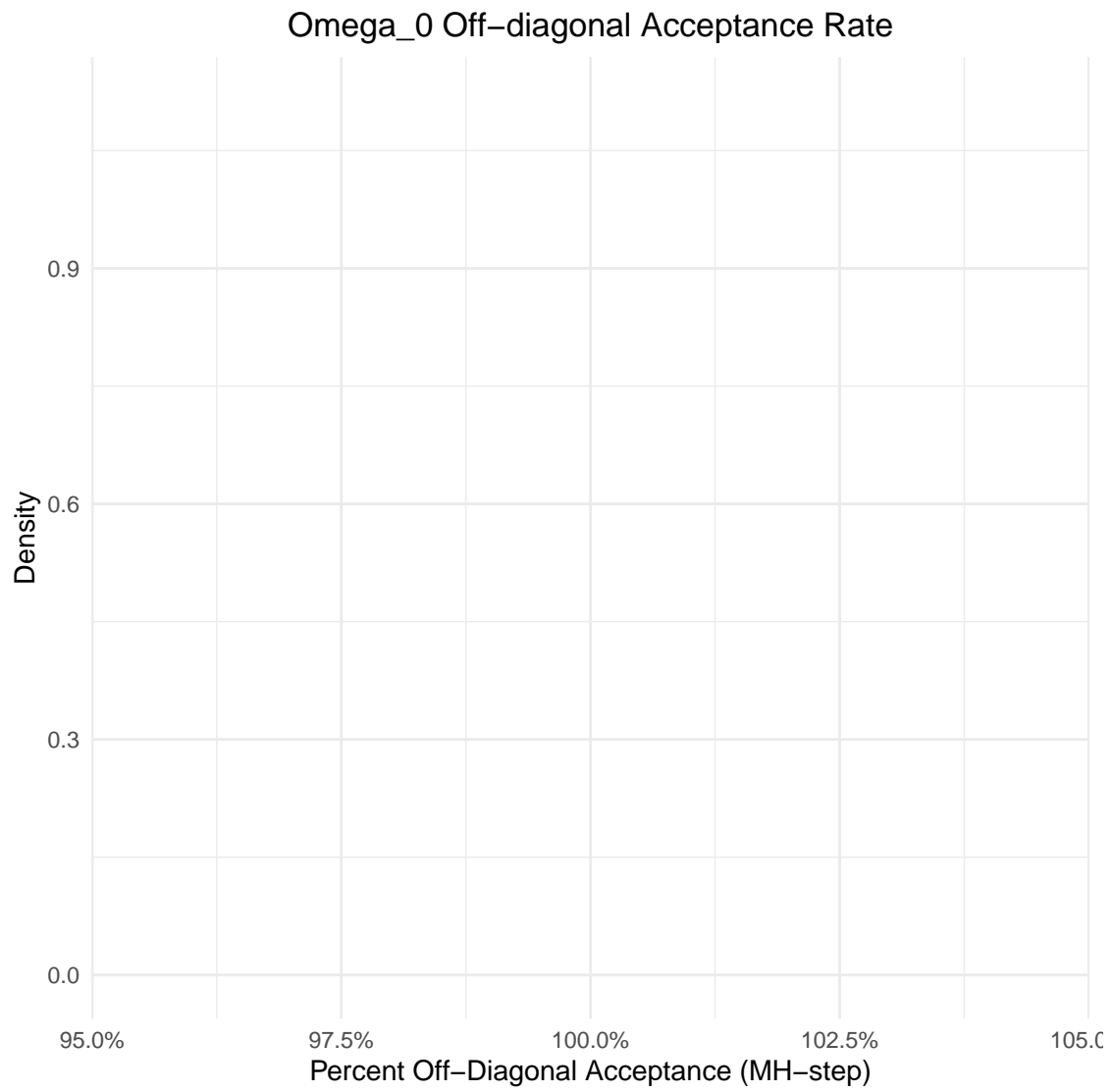




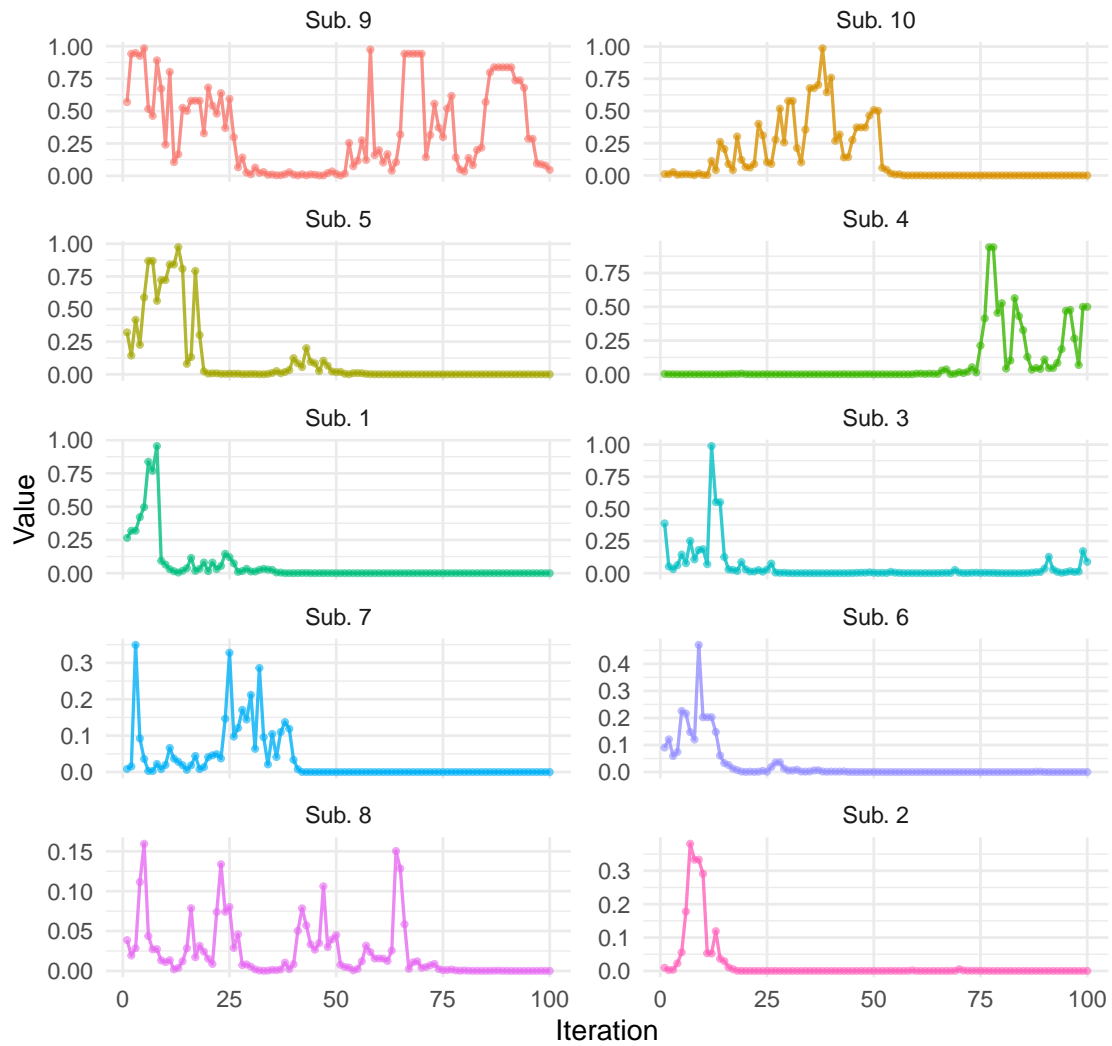
### 2.2.2 $\Omega_0$ with free $\tau_k$

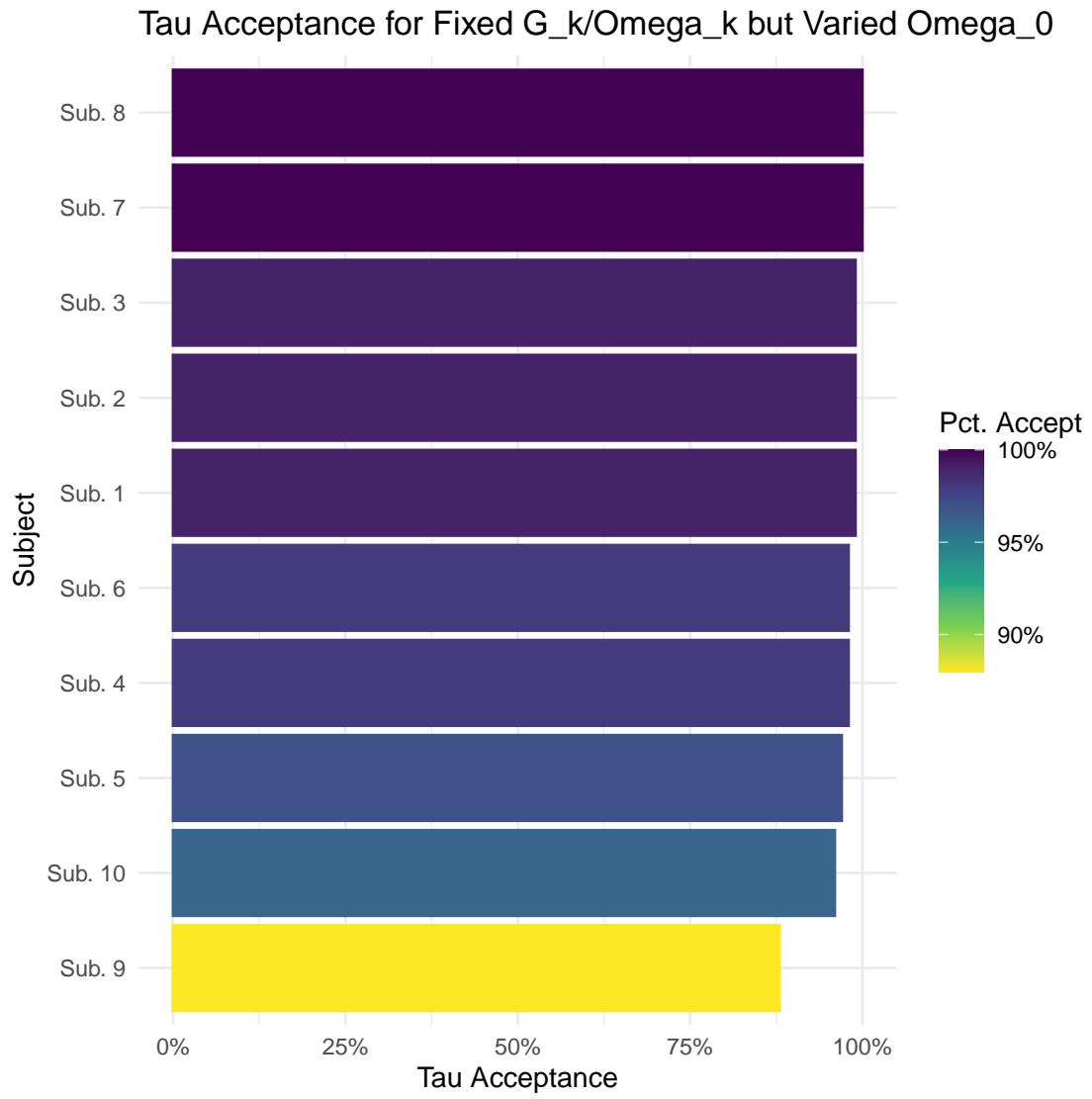
Omega\_0 Chain with Fixed G\_k/Omega\_k and Free Tau\_k





## Tau\_k Chain with Fixed $G_k/\Omega_k$ , Varied $\Omega_0$

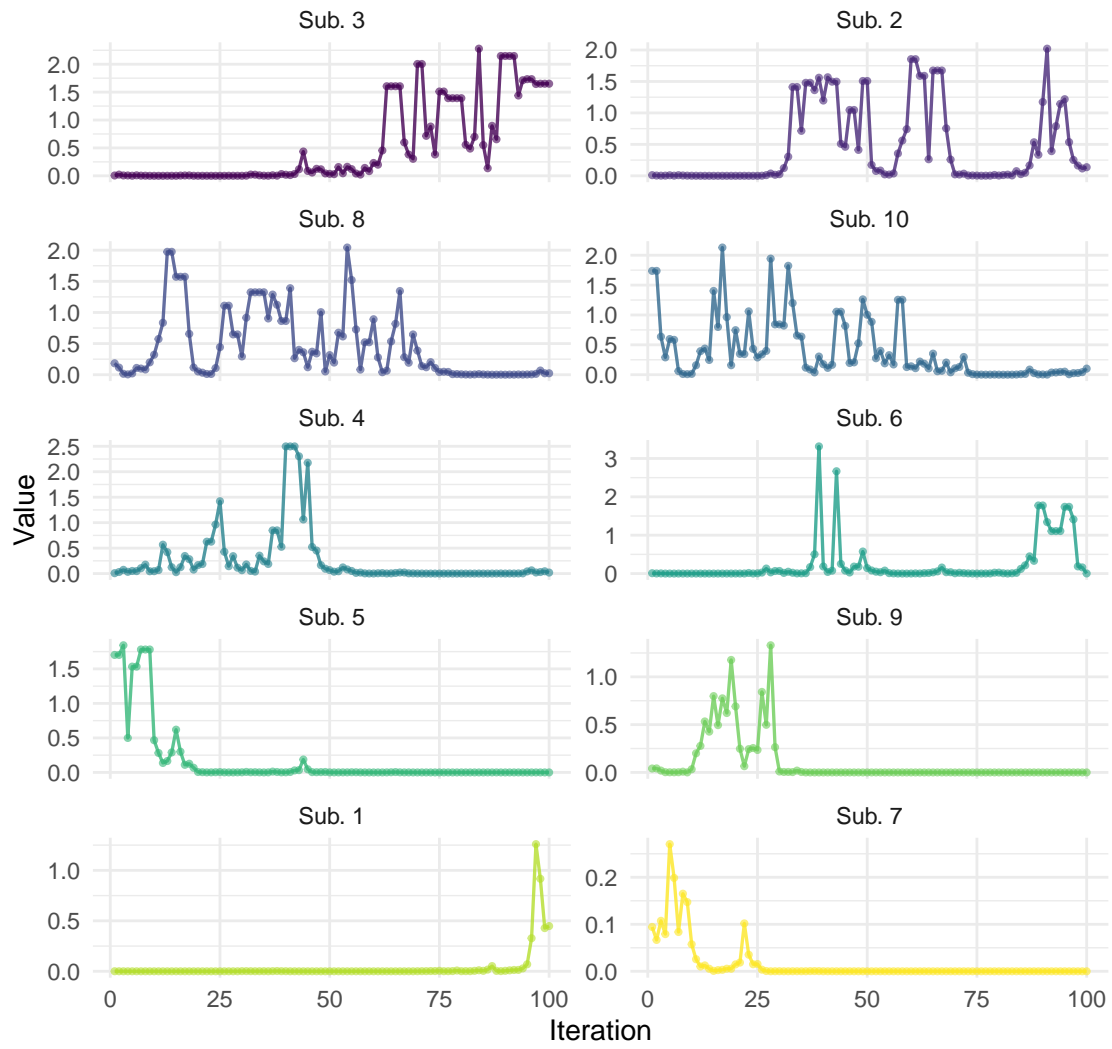


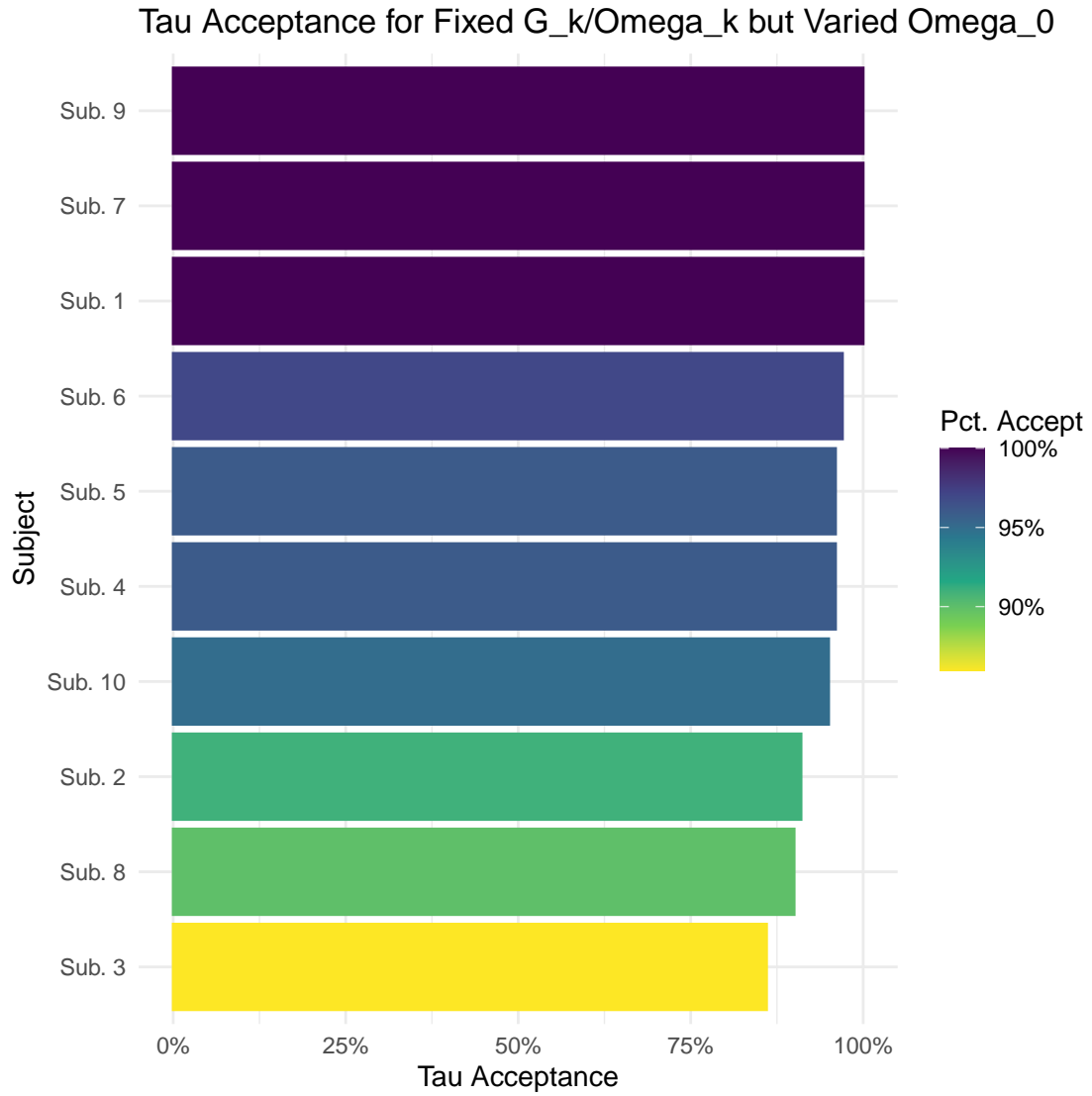




### 2.2.3 $\tau_k$

Tau\_k Chain with Fixed  $G_k/\Omega_k$ , Varied  $\Omega_0$





Adaptive Tau-proposal Step-size/Variance

Subj. 1	Subj. 2	Subj. 3	Subj. 4	Subj. 5	Subj. 6	Subj. 7	Subj. 8	Subj. 9	Subj. 10
Adaptive step 1									
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Adaptive step 2									
1.05	1.05	1.00	0.95	0.95	1.05	0.95	0.95	1.05	0.95
Adaptive step 3									
1.10	1.10	1.05	0.90	0.90	1.10	0.90	1.00	1.10	1.00
Adaptive step 4									
1.15	1.15	1.10	0.95	0.95	1.15	0.95	1.05	1.15	1.05
Adaptive step 5									
1.20	1.20	1.15	1.00	1.00	1.20	1.00	1.10	1.20	1.00

### **2.2.2 $\Omega_k$ & $G_k$**

### **2.2.4 Sensitivity to Regularization Hyperparameters ( $\lambda_j \mid a_j, b_j \sim \Gamma(a_j, b_j)$ )**

### **2.2.1 Posterior Distribution**

### **2.2.2. Accept/Reject Rates for Proposals**