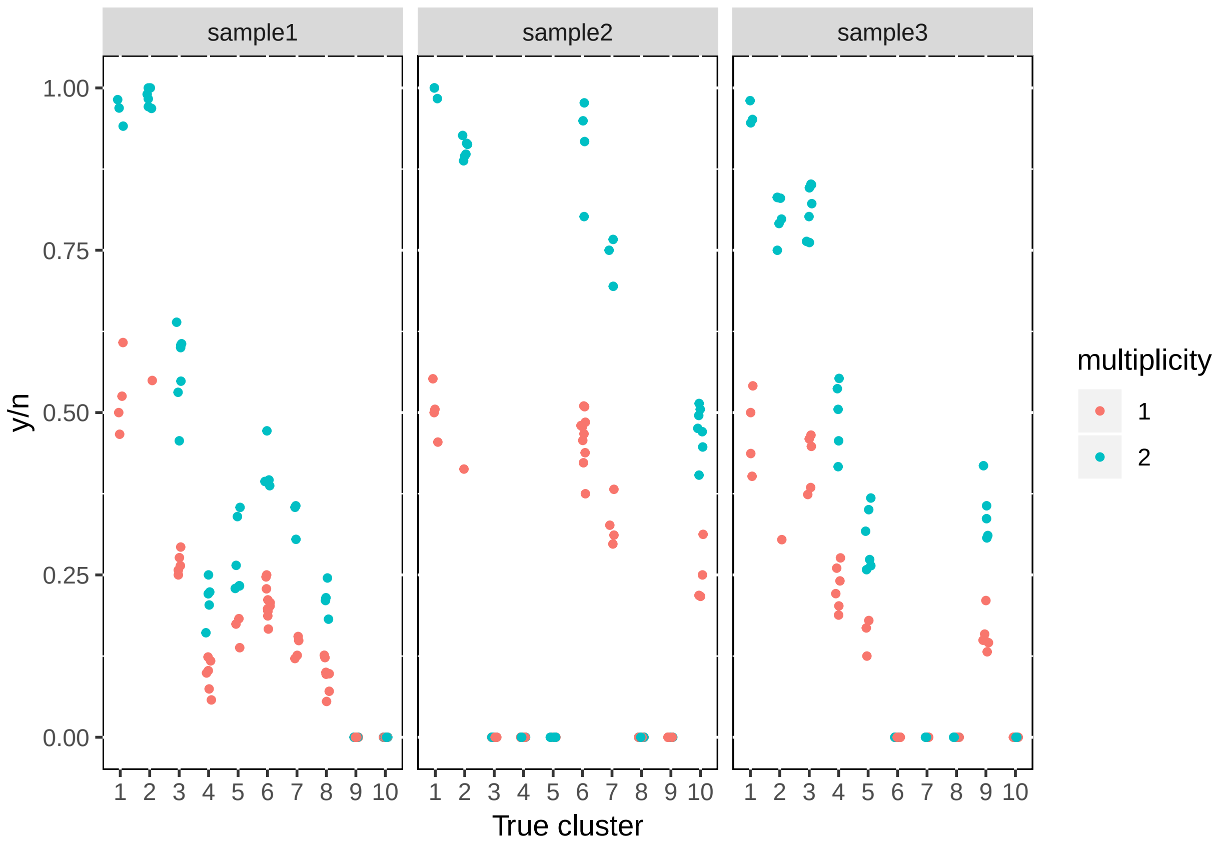
Version 6: Cluster first

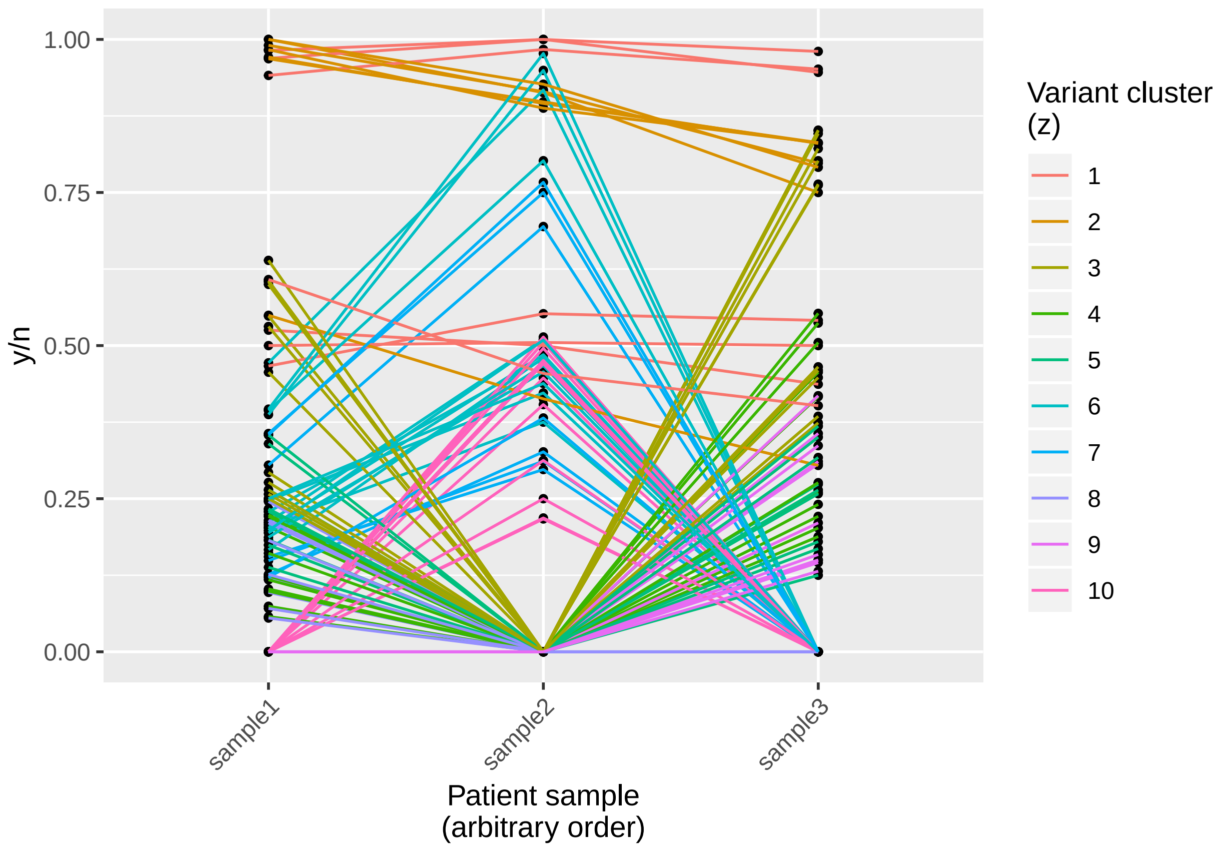
Simulate data

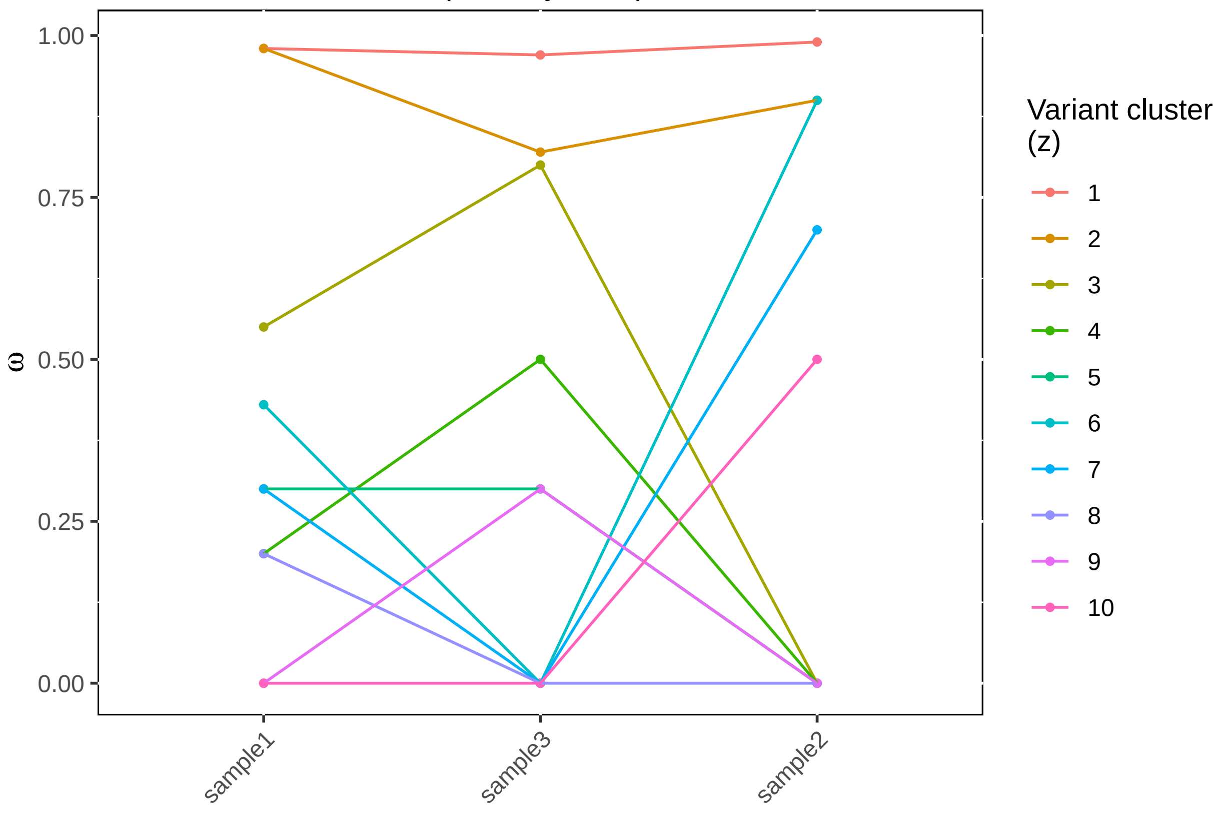
```r
I <- 100
K <- 10
S <- 3

set.seed(123)

pi <- rep(0.1, 10)
z <- sample(1:K, size = I, replace = T, prob = pi)

w <- matrix(c(0.98, 0.99, 0.97,
              0.98, 0.90, 0.82,
              0.55, 0.00, 0.80,
              0.20, 0.00, 0.50,
              0.30, 0.00, 0.30,
              0.43, 0.90, 0.00,
              0.30, 0.70, 0.00,
              0.20, 0.00, 0.00,
              0.00, 0.00, 0.30,
              0.00, 0.50, 0.00),
            byrow=T,
            nrow=K, ncol=S)

colnames(w) <-  paste0("sample", 1:S)

tcn <- matrix(2, nrow=I, ncol=S)
m <- matrix(rep(sample(1:2, size = I, replace = T), S),
            nrow=I, ncol=S)
W <- w[z, ]
calcTheta <- function(m, tcn, w) {
  (m * w) / (tcn * w + 2*(1-w))
}

theta <- calcTheta(m, tcn, W)

n <- replicate(S, rpois(I, 100))
y <- matrix(NA, nrow=I, ncol=S)
for (i in 1:I) {
  for (s in 1:S) {
    y[i, s] <- rbinom(1, n[i, s], theta[i,s])
  }
}

test.data <- list("I" = I, "S" = S, "K" = K,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
```

Visualize densities of simulated data

Clustering is by ω




1.00
0.75
0.50
0.25
0.00
y/n
sample1
sample2
sample3
Patient sample
(arbitrary order)
Variant cluster
(z)
1
2
3
4
5
6
7
8
9
10


1.00
0.75
0.50
0.25
0.00
ω
sample1
sample3
sample2
Variant cluster
(z)
1
2
3
4
5
6
7
8
9
10

functions

```r
runMCMC <- function(data, K, jags.file, inits, params, n.iter, thin) {
  data$K <- K
  jags.m <- jags.model(jags.file, data,
                       n.chains = 1,
                       inits = inits,
                       n.adapt = 1000)
  samps <- coda.samples(jags.m, params, n.iter=n.iter, thin=thin)
  samps
}

getParamChain <- function(samps, param) {
  chains <- do.call(rbind, samps)
  chain <- chains[, grep(param, colnames(chains))]
}

reshapeW <- function(w, S, K) {
  w.mat <- matrix(w, nrow = K)
  colnames(w.mat) <- paste0("sample", 1:S)
  w.mat
}

calcLogLik <- function(z.iter, w.iter, data) {
  W <- w.iter[z.iter, ]
  theta <- calcTheta(data$m, data$tcn, W)
  sum(dbinom(data$y, data$n, theta, log=T))
}

calcChainLogLik <- function(samps, data, K) {
  z.chain <- getParamChain(samps, "z")
  w.chain <- getParamChain(samps, "w")
  lik <- c()
  for(iter in 1:nrow(z.chain)) {
    z.iter <- z.chain[iter,]
    w.iter <- reshapeW(w.chain[iter,], data$S, K)
    lik <- c(lik, calcLogLik(z.iter, w.iter, data))
  }
  mean(lik)
}

calcBIC <- function(n, k, ll) log(n)*k - 2*ll
```

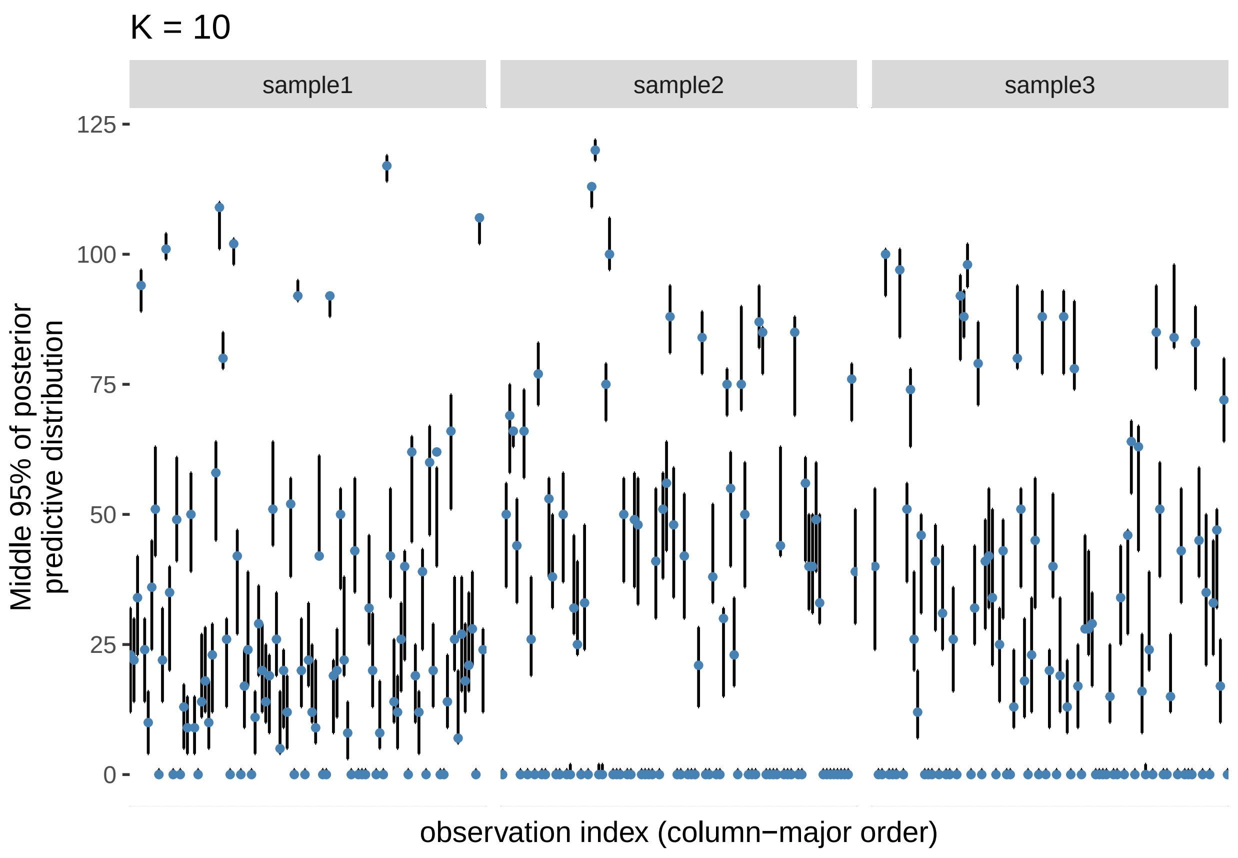
Cluster – JAGS

```r
jags.files <- c(file.path(models.dir, "w.jags"),
                file.path(models.dir, "w_K1.jags"))
inits <- list(".RNG.name" = "base::Wichmann-Hill",
              ".RNG.seed" = 123)
test.data <- list("I" = I, "S" = S,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
params <- c("z", "w", "ystar")
```

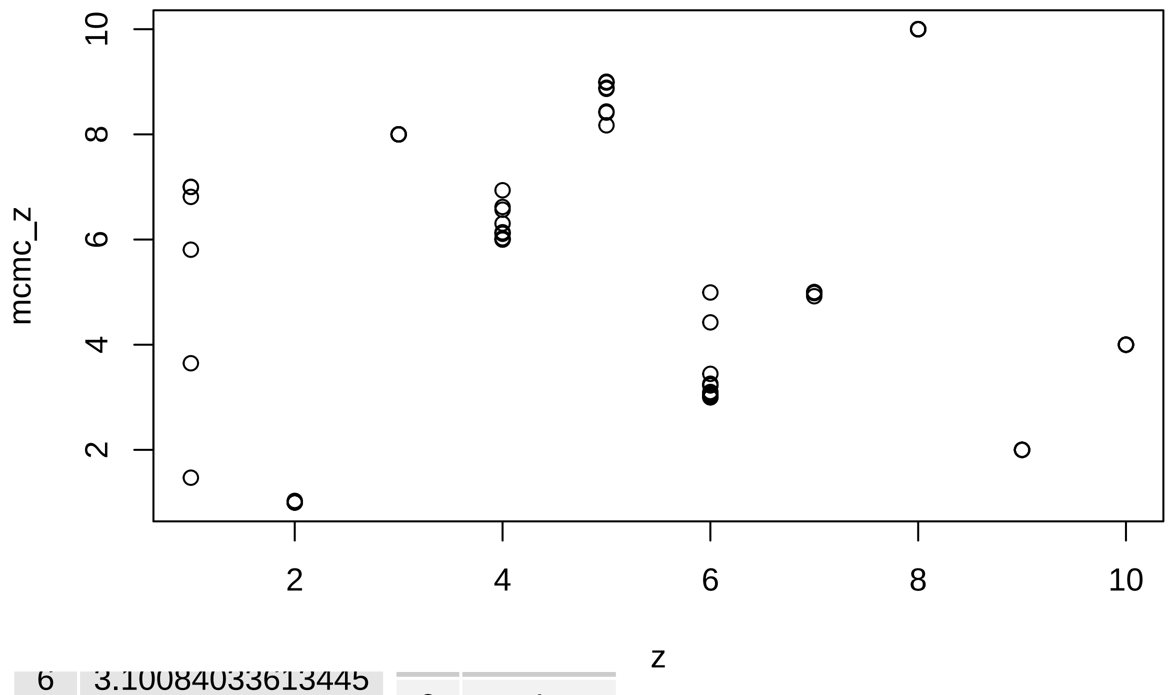
```r
n.iter = 10000
thin = 7
K <- 10

samps <- runMCMC(test.data, K, jags.files[1], inits, params, n.iter, thin)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 300
##    Unobserved stochastic nodes: 431
##    Total graph size: 4788
##
## Initializing model
```

```r
z.chain <- getParamChain(samps, "z")
w.chain <- getParamChain(samps, "w")

#kToTest <- 1:5

#BIC <- c()
#samps.list <- list()

# for(ix in 1:length(kToTest)) {
#   if(kToTest[ix] == 1) {
#     jags.file <- jags.files[2]
#   } else {
#     jags.file <- jags.files[1]
#   }
#
#   K <- kToTest[ix]
#   samps <- runMCMC(test.data, K, jags.file, inits, params, n.iter, thin)
#   z.chain <- getParamChain(samps, "z")
#   w.chain <- getParamChain(samps, "w")
#   bic <- calcBIC(length(test.data$y), K, calcChainLogLik(samps, test.data, K))
#   BIC <- c(BIC, bic)
#   samps.list[[ix]] <- samps
# }
# names(BIC) <- paste0("K", kToTest)
# BIC
# best <- which.min(BIC)
# s1 <- samps.list[[best]]
# bestK <- kToTest[best]
#
# ggplot(data.frame(numClust = kToTest, BIC), aes(x=numClust, y=BIC)) +
#   geom_point(size=1) +
#   geom_line() +
#   xlab("Number of Clusters") +
#   geom_vline(xintercept=bestK, linetype="dashed")

#s1.w <- getParamChain(s1, "w")
```



```
plot.z <- function(samps, z) {
  mcmc_vals <- summary(samps)$statistics
  mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
  plot(z, mcmc_z, type = "p")
  z_comp <- data.frame(z, mcmc_z)
  plot.new()
  grid.table(z_comp, rows=NULL)
  z_mapping <- distinct(round(z_comp, 0))
  z_mapping <- z_mapping[order(z_mapping$mcmc_z), ]
  plot.new()
  grid.table(z_mapping, rows=NULL)
}
mcmc_vals <- summary(samps)$statistics
mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
plot.z(samps, z)
```



6	3.10084033613445
6	3.2296918767507
5	8.98529411764706
3	8
3	8
4	6.3046218487395
6	3.4453781512605
4	6.5672268907563
10	4
2	1
6	3.25770308123249
9	2
3	8
7	4.92156862745098
4	6.61974789915966
3	8
9	2

2	1
1	1
9	2
6	3
6	4
10	4
1	4
7	5
6	5
4	6
1	6
4	7
1	7
3	8
5	8
5	9