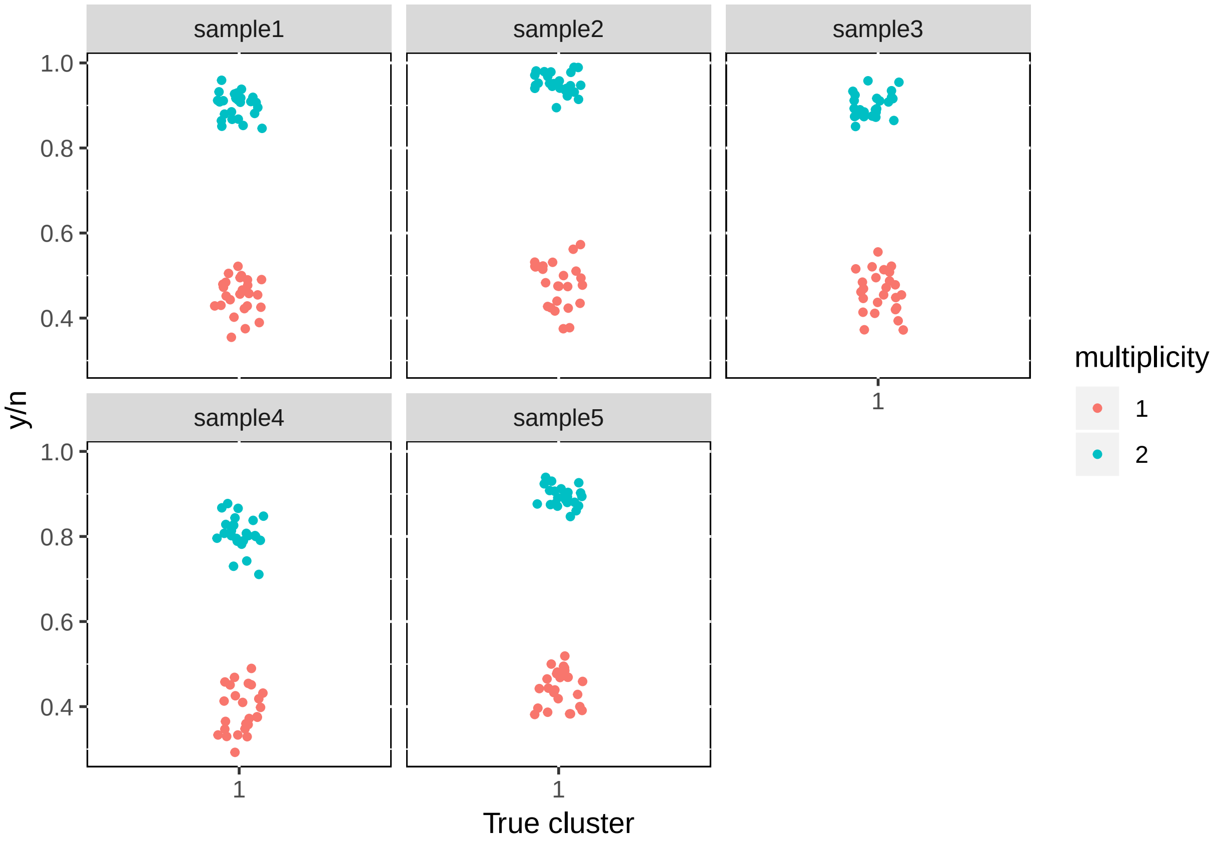
Version 5: 5 samples, 50 variants; truth = 1 cluster

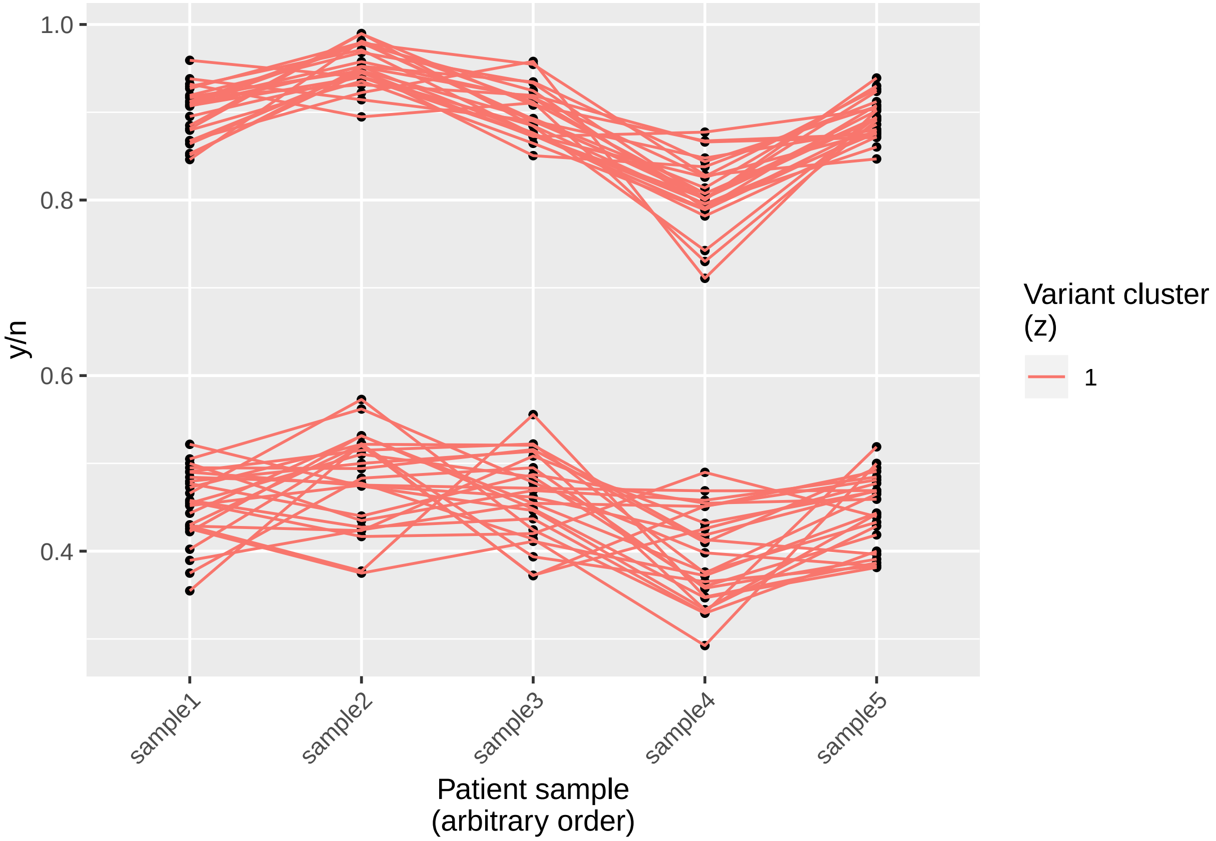
Simulate data

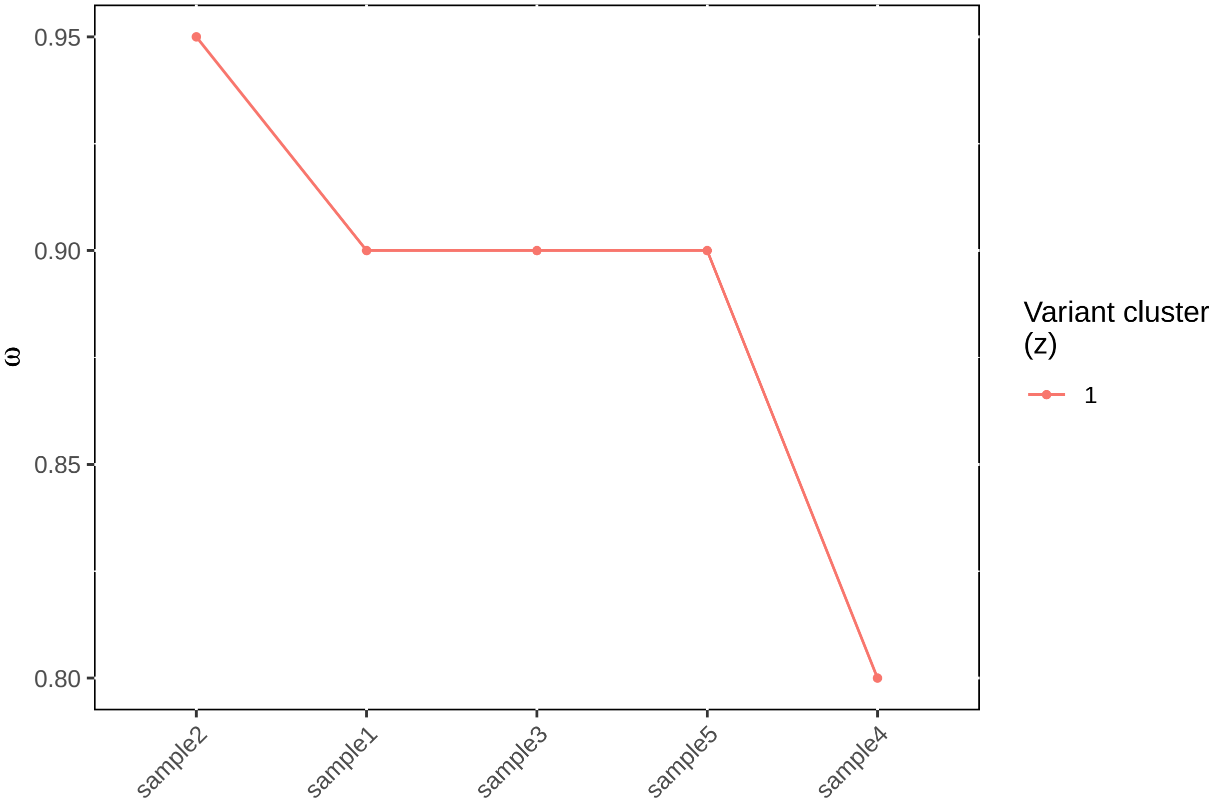
```
I <- 50
K <- 1
S <- 5

set.seed(123)

pi <- c(0.2, 0.5, 0.3)
z <- rep(1, I)

w <- matrix(c(0.9, 0.95, 0.9, 0.8, 0.9),
            byrow=T,
            nrow=K, ncol=S)
# w <- matrix(c(0.9, 0.95, 0.9, 0.8, 0.9,
#               0.7, 0.8, 0.85, 0.78, 0.7,
#               0.7, 0.2, 0.83, 0.76, 0.5),
#             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


sample1
sample2
sample3
sample4
sample5
1.0
0.8
0.6
0.4
y/n
1
True cluster
multiplicity
● 1
● 2


1.0
0.8
0.6
0.4
y/n
sample1
sample2
sample3
sample4
sample5
Patient sample
(arbitrary order)
Variant cluster
(z)
— 1



functions

```
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
```

JAGS

```
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")

n.iter = 10000
thin = 7
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
}
```

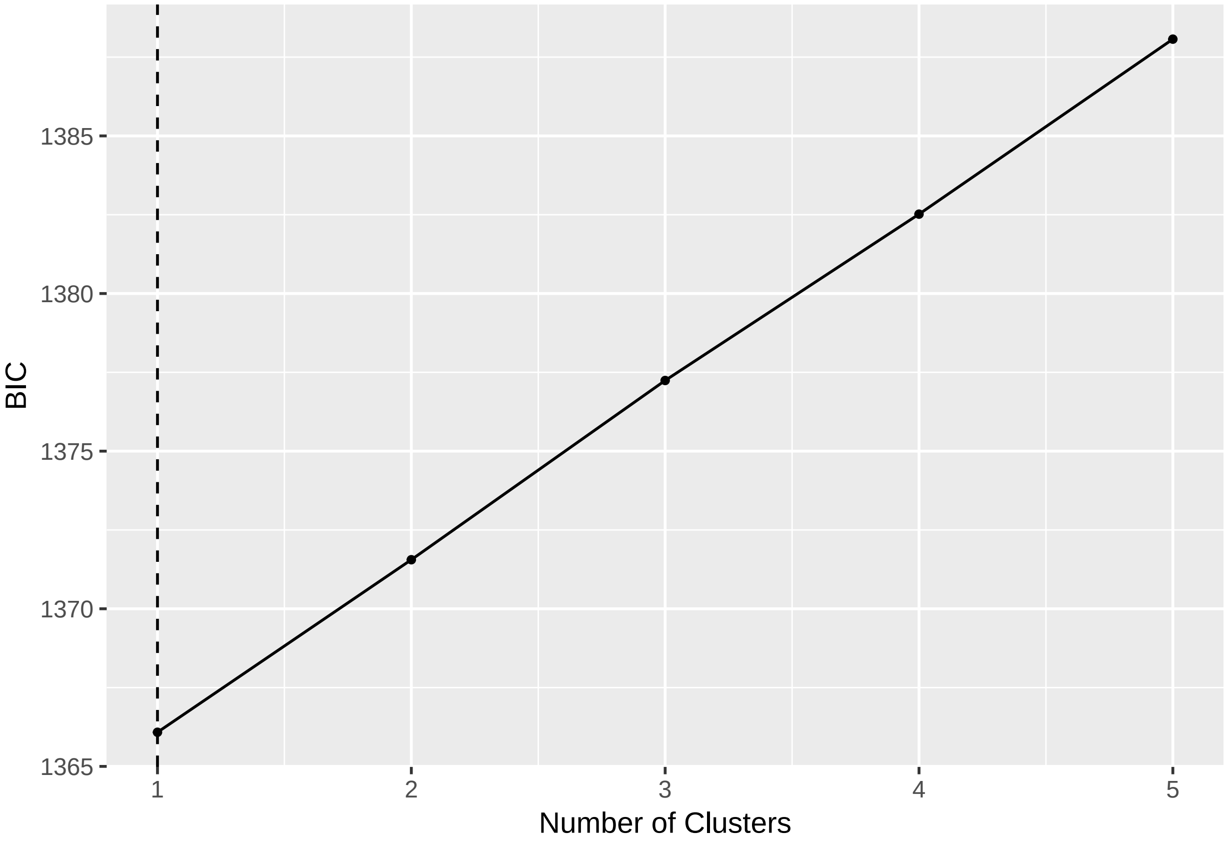
```
## Warning in jags.model(jags.file, data, n.chains = 1, inits = inits, n.adapt
## = 1000): Unused variable "K" in data
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 250
##    Unobserved stochastic nodes: 255
##    Total graph size: 1335
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 250
##    Unobserved stochastic nodes: 311
##    Total graph size: 3943
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 250
##    Unobserved stochastic nodes: 316
##    Total graph size: 3948
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 250
##    Unobserved stochastic nodes: 321
##    Total graph size: 3953
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 250
##    Unobserved stochastic nodes: 326
##    Total graph size: 3958
##
## Initializing model
```

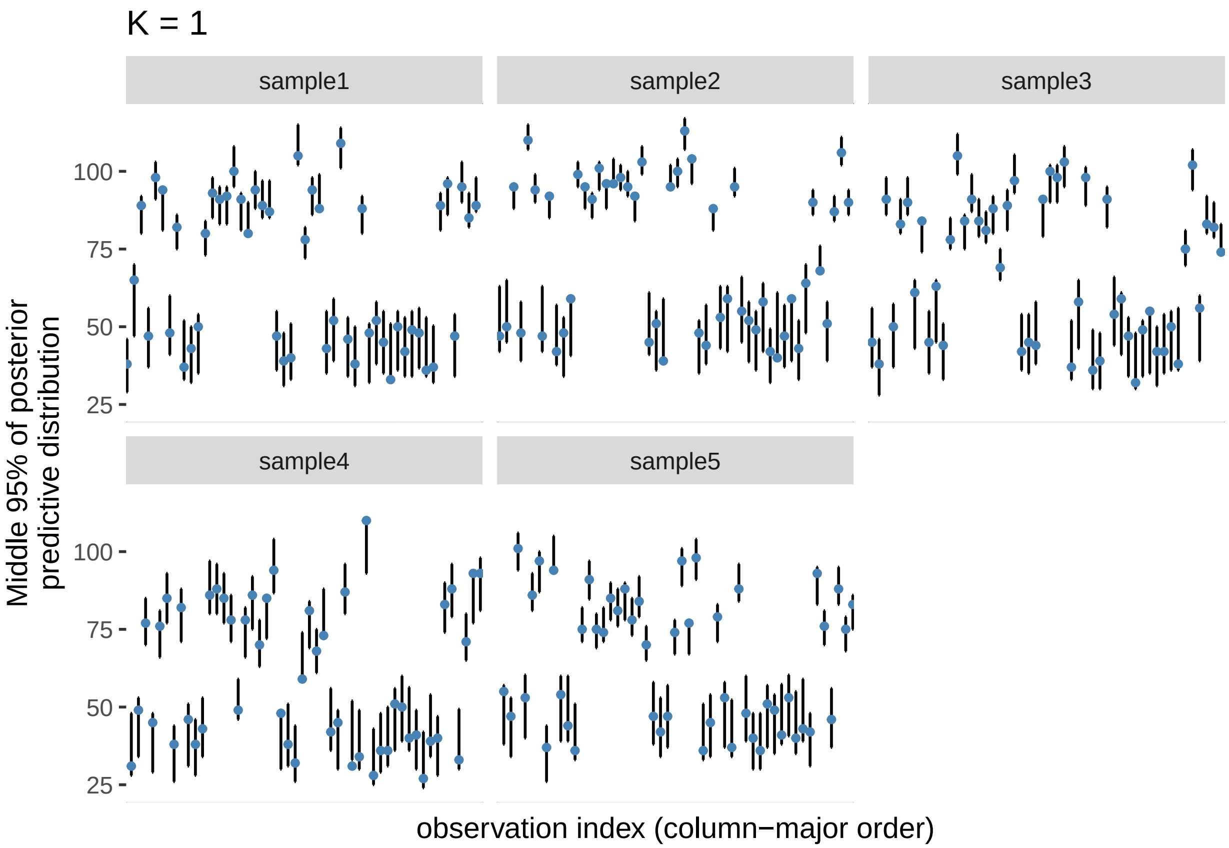
```r
names(BIC) <- paste0("K", kToTest)
BIC
```

```
##        K1       K2       K3       K4       K5
## 1366.082 1371.555 1377.241 1382.518 1388.069
```

```
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")
```


K = 1
sample1
sample2
sample3
sample4
sample5
Middle 95% of posterior predictive distribution
100
75
50
25
100
75
50
25
observation index (column−major order)