

Version 5: 5 samples, 15 variants; truth = 3 clusters, linear

Simulate data

```
I <- 15
K <- 3
S <- 5

set.seed(123)

pi <- c(0.2, 0.5, 0.3)
z <- sample(1:K, size = I, replace = T, prob = pi)

w <- matrix(c(0.9, 0.95, 0.9, 0.8, 0.9,
              0.89, 0.85, 0.85, 0.78, 0.7,
              0.85, 0.7, 0.83, 0.76, 0.6),
            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))
}

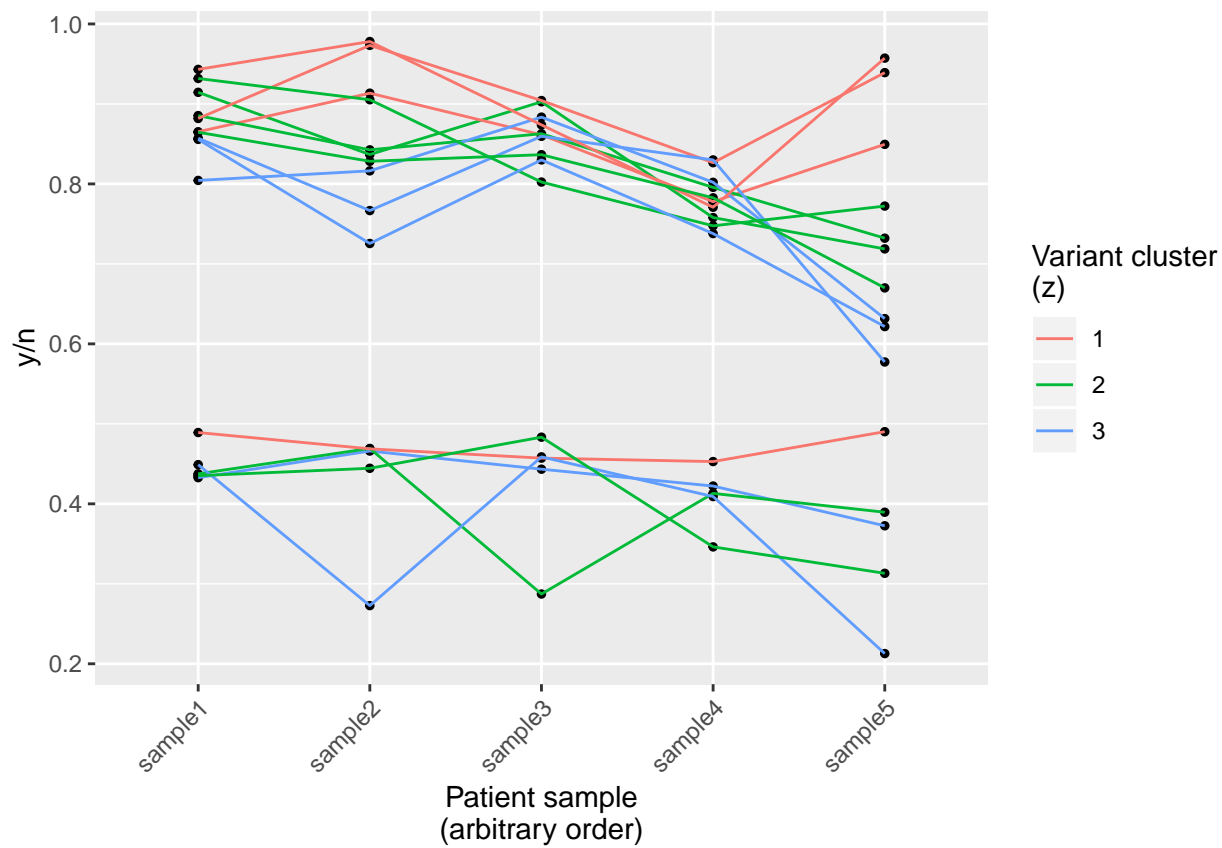
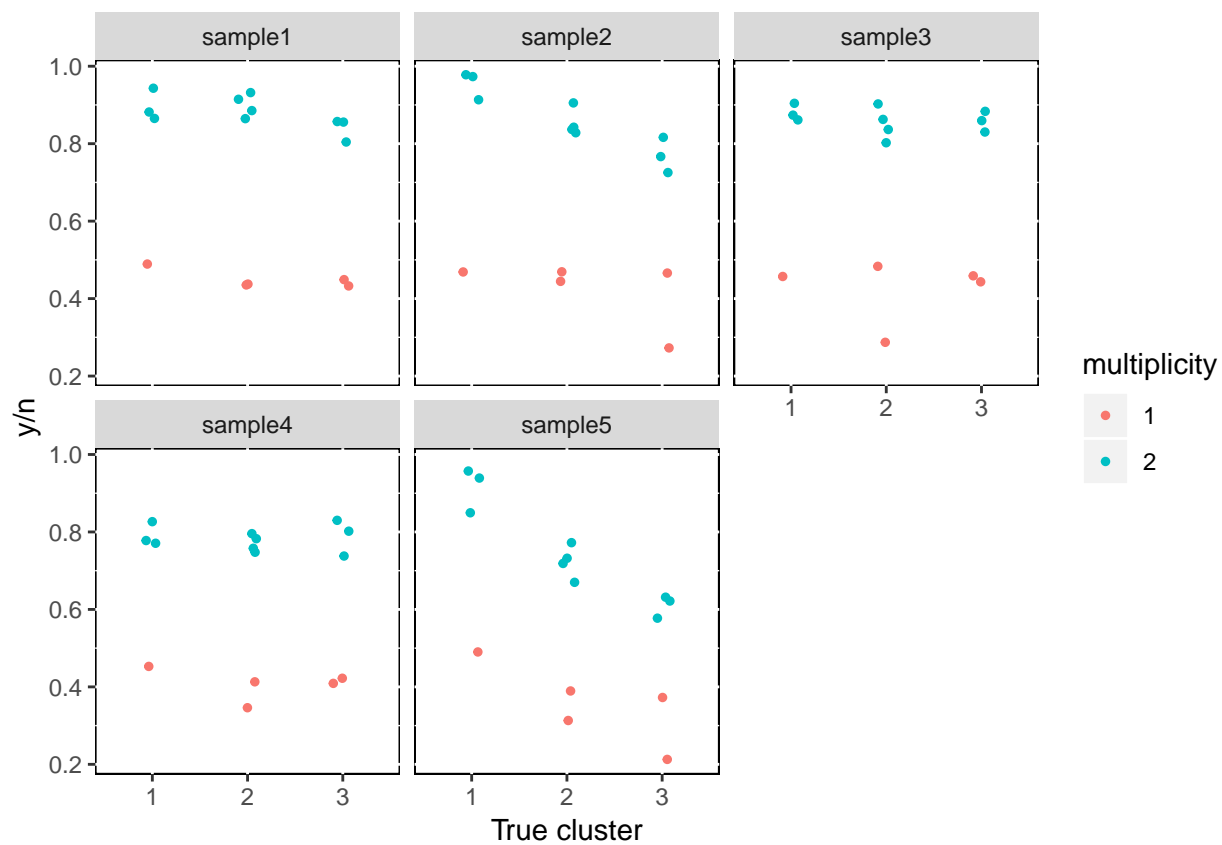
theta <- calcTheta(m, tcn, W)

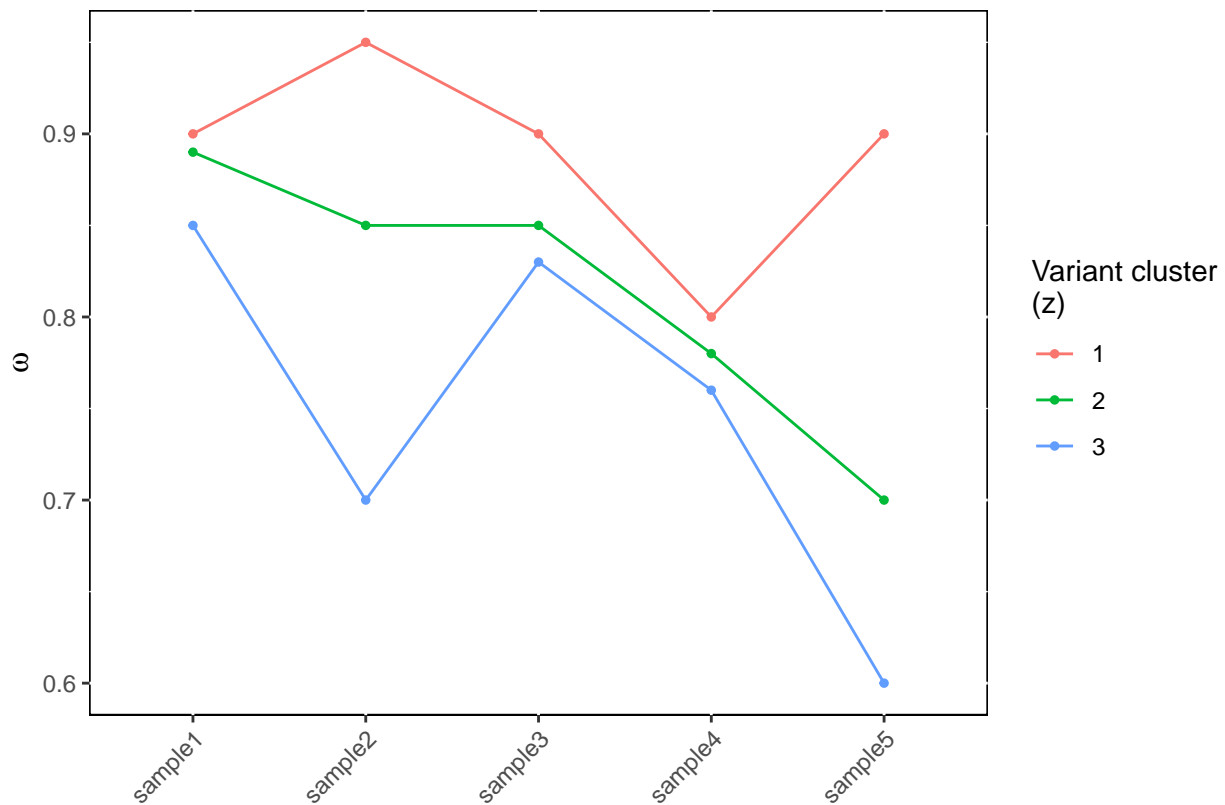
n <- replicate(S, rpois(I, 100))
y <- matrix(NA, nrow=I, ncol=S)
for (i in 1:I) {
  for (s in 1:S) {
    y[i, s] <- rbinom(1, n[i, s], theta[i,s])
  }
}

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





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

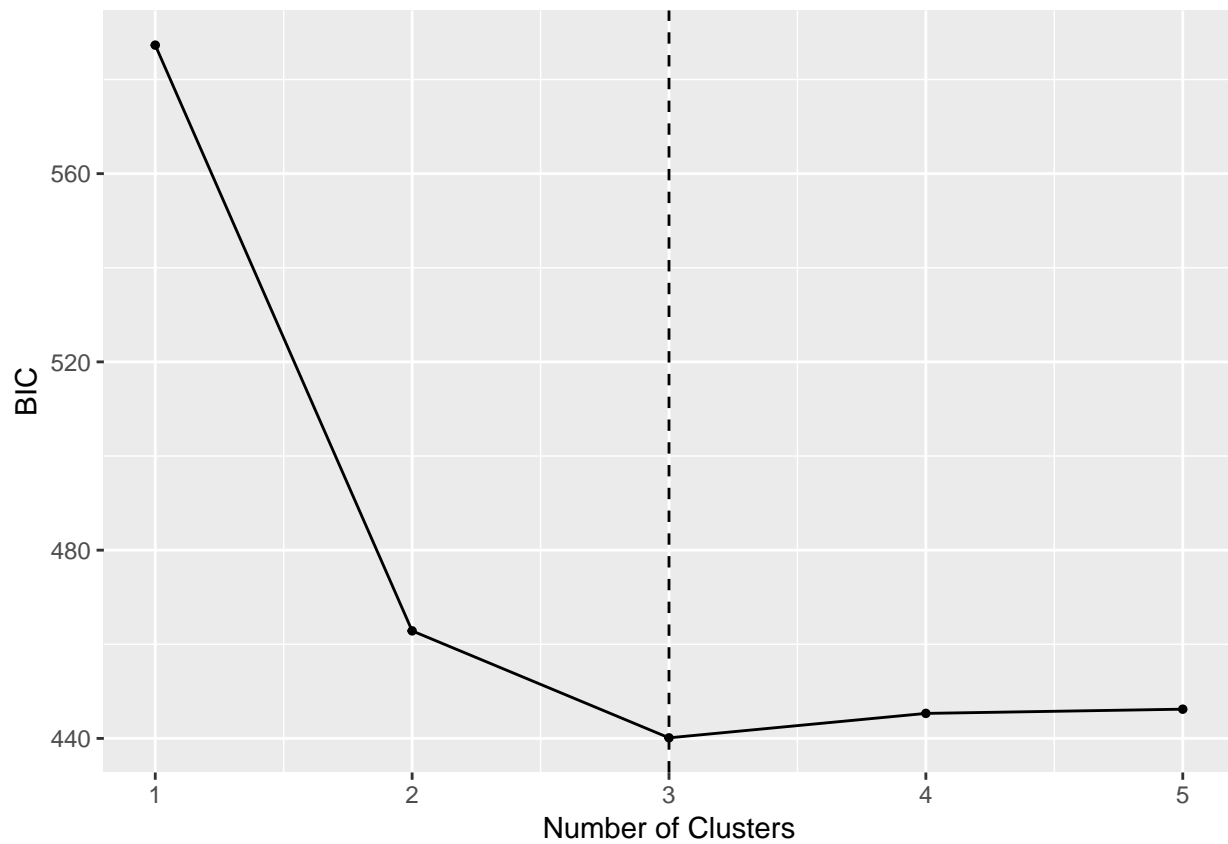
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 75
##   Unobserved stochastic nodes: 80
##   Total graph size: 460
##
## Initializing model
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 75
##   Unobserved stochastic nodes: 101
##   Total graph size: 1183
##
## Initializing model
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 75
##   Unobserved stochastic nodes: 106
##   Total graph size: 1188
##
## Initializing model
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 75
##   Unobserved stochastic nodes: 111
##   Total graph size: 1193
##
## Initializing model
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 75
##   Unobserved stochastic nodes: 116
##   Total graph size: 1198
##
## Initializing model
names(BIC) <- paste0("K", kToTest)
BIC

```

```
##      K1      K2      K3      K4      K5
## 587.3050 462.8531 440.1249 445.3188 446.2112
```

```
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 = 3

