Bayesian vs Frequentist Accumulation Curve CIs

April 21, 2018

Accumulation Curve CI

Frequentist confidence interval

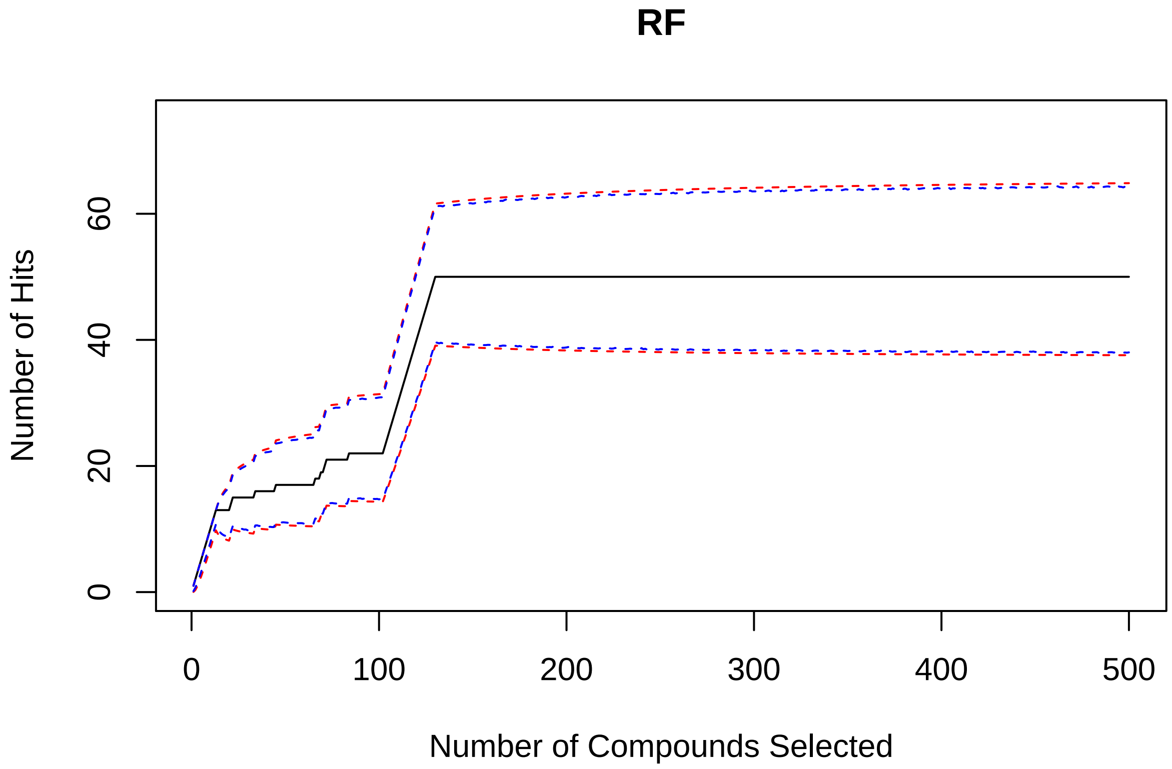
```
int.list.burd <- list(length = ncol(probs.Burd) - 1)
for(i in 2:ncol(probs.Burd)) {
  probs <- probs.Burd[, i]
  hit.vec <- probs.Burd$Observed

  order.idx <- order(probs, decreasing = T)
  probs <- probs[order.idx]
  hit.vec <- hit.vec[order.idx]

  m <- length(probs)

  # Matrix containing the number of hits and lower and upper bounds for 95%
  # confidence intervals for each number of tests
  int.mat <- matrix(ncol = 3, nrow = m)
  colnames(int.mat) <- c("NHits", "LB", "UB")

  for(j in 1:m) {
    int.mat[j, 1] <- sum(hit.vec[1:j])
    int.mat[j, 2:3] <- j * CPInt(x = sum(hit.vec[1:j]), p.vec = probs[1:j])
  }
  int.list.burd[[i-1]] <- int.mat
}
```

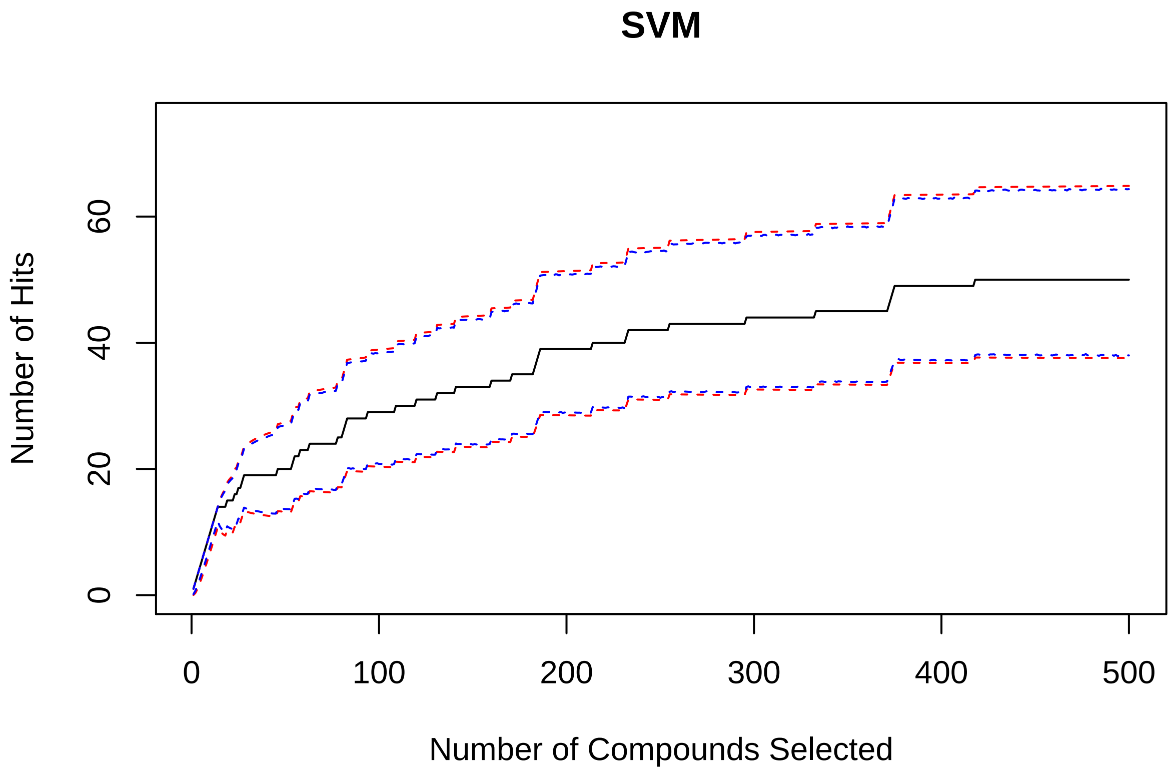
Bayesian credible intervals with Jeffreys prior

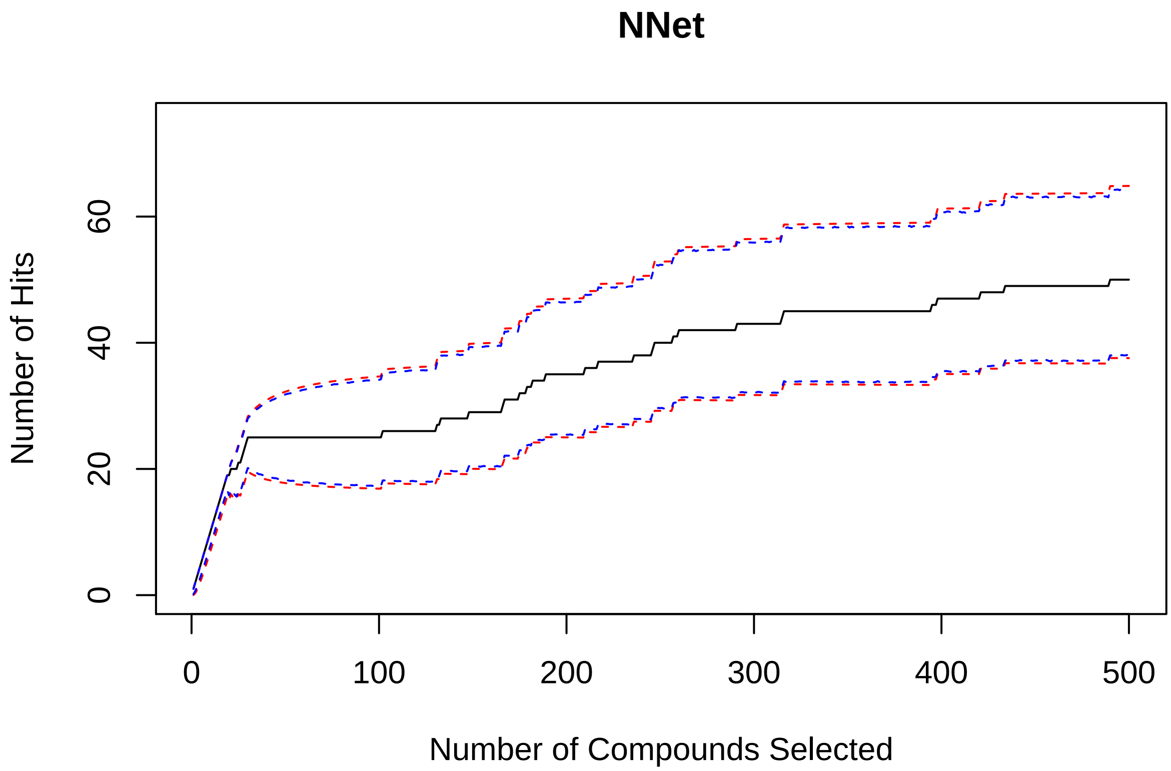
I am using the model that I described today. Let me know if you would like more details.

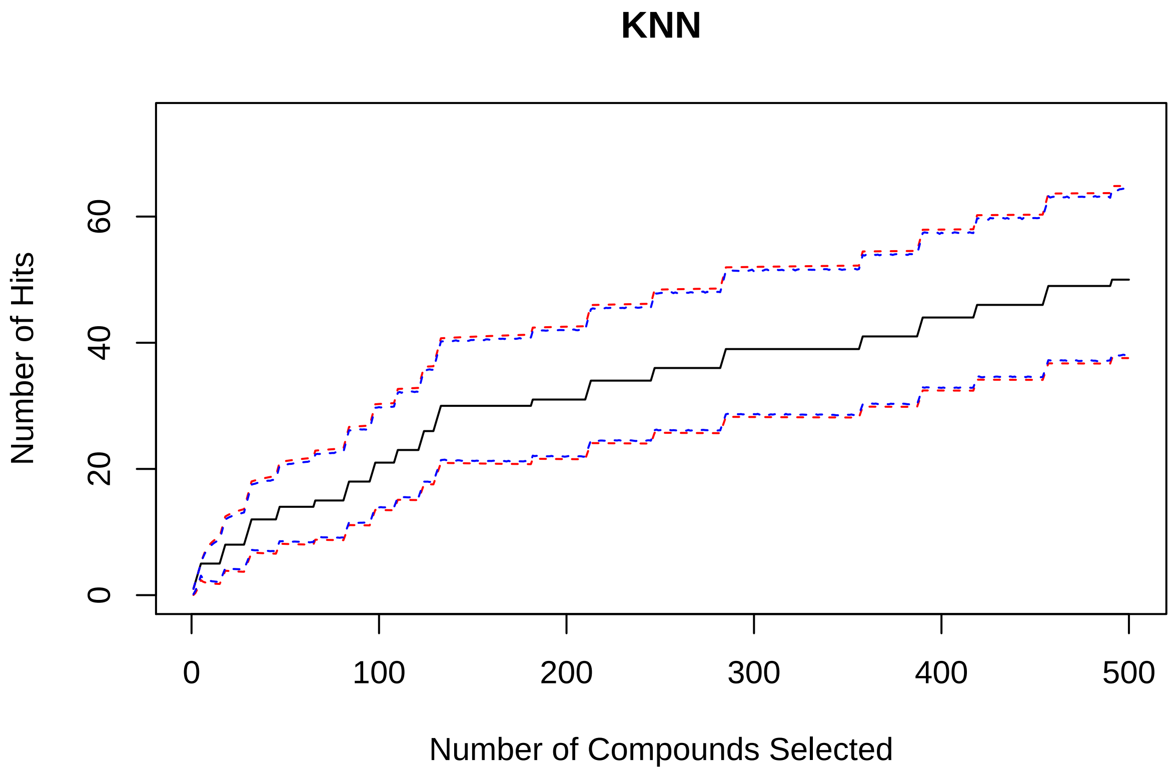
Blue CIs are Bayesian, Red are Frequentist.

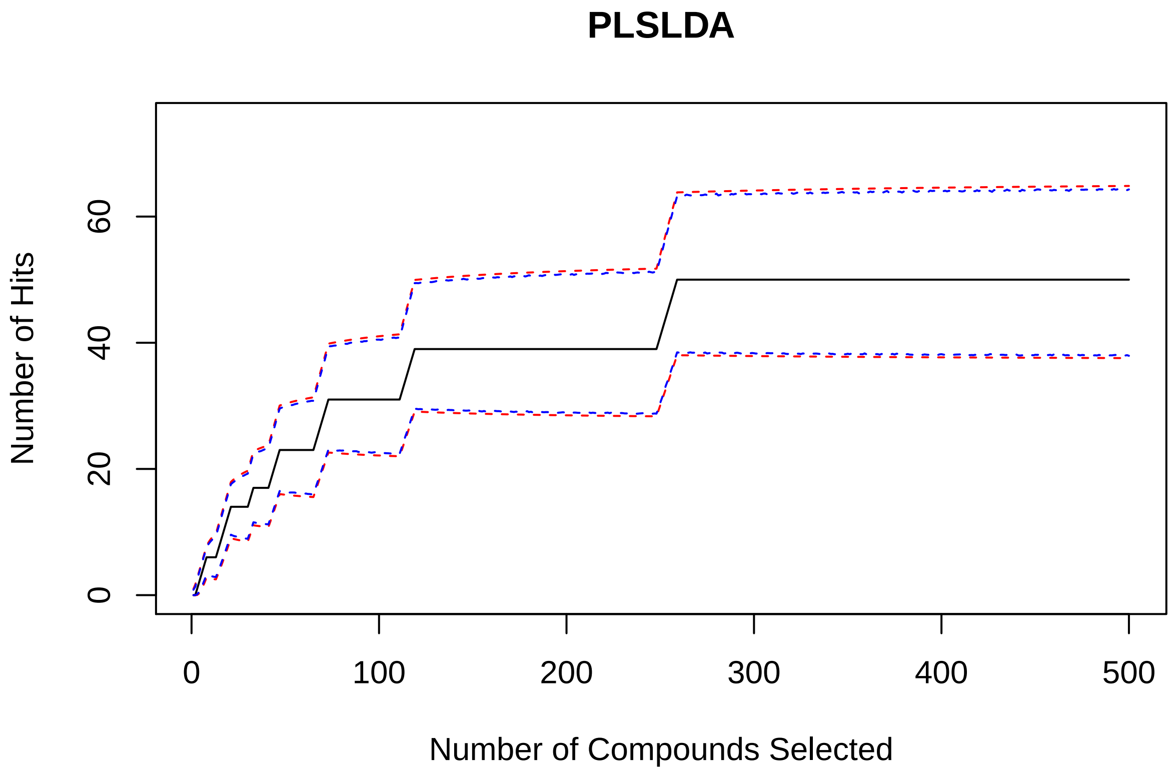
```
modelType <- c("Tree", "RF", "SVM", "NNet", "KNN", "PLSLDA")

u <- 1:500
i <- 2
for (i in 2:7) {

  hit.vec <- probs.Burd$Observed
  probs <- probs.Burd[, i]

  order.idx <- order(probs, decreasing = T)
  probs <- probs[order.idx]
  hit.vec <- hit.vec[order.idx]

  m <- length(probs)
  int.mat <- matrix(ncol = 3, nrow = m)
  colnames(int.mat) <- c("NHits", "LB", "UB")

  a <- vector(length = 500)
  b <- vector(length = 500)
  for(j in 1:500) {
    a[j] <- sum(hit.vec[1:j]) + .5
    b[j] <- j - sum(hit.vec[1:j]) + .5
  }

  sum.samp <- vector(length = 100000)
  for(j in 1:500) {
    int.mat[j, 1] <- sum(hit.vec[1:j])
    sum.samp <- rbeta(100000, a[j], b[j]) * j
    int.mat[j, 2] <- quantile(sum.samp, probs = .025)
    int.mat[j, 3] <- quantile(sum.samp, probs = .975)
  }

  plot(int.list.burd[[i-1]][, 1], type = "l", ylim = c(0, 75),
       main = colnames(probs.Burd)[i+1], ylab = "Number of Hits",
       xlab = "Number of Compounds Selected")
  lines(int.list.burd[[i-1]][, 2], type = "l", lty = "dashed", col = "red")
  lines(int.list.burd[[i-1]][, 3], type = "l", lty = "dashed", col = "red")

  lines(int.mat[, 2], type = "l", lty = "dashed", col = "blue")
  lines(int.mat[, 3], type = "l", lty = "dashed", col = "blue")
}
```

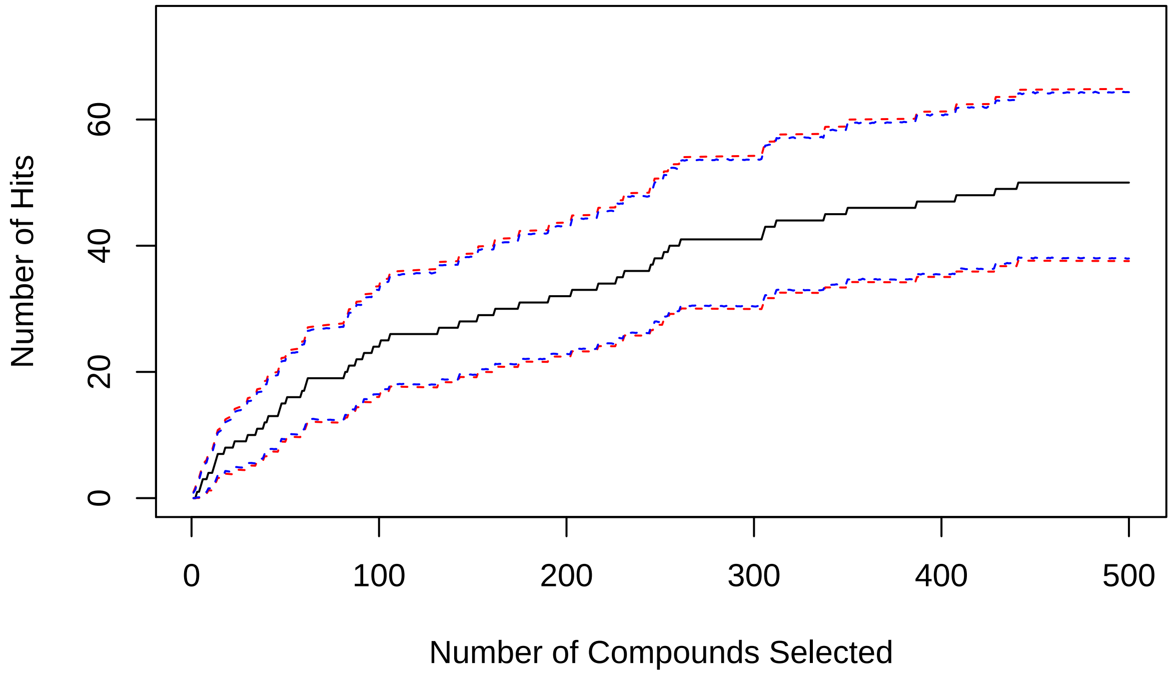



SVM
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


NNet
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


KNN
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


PLSLDA
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected

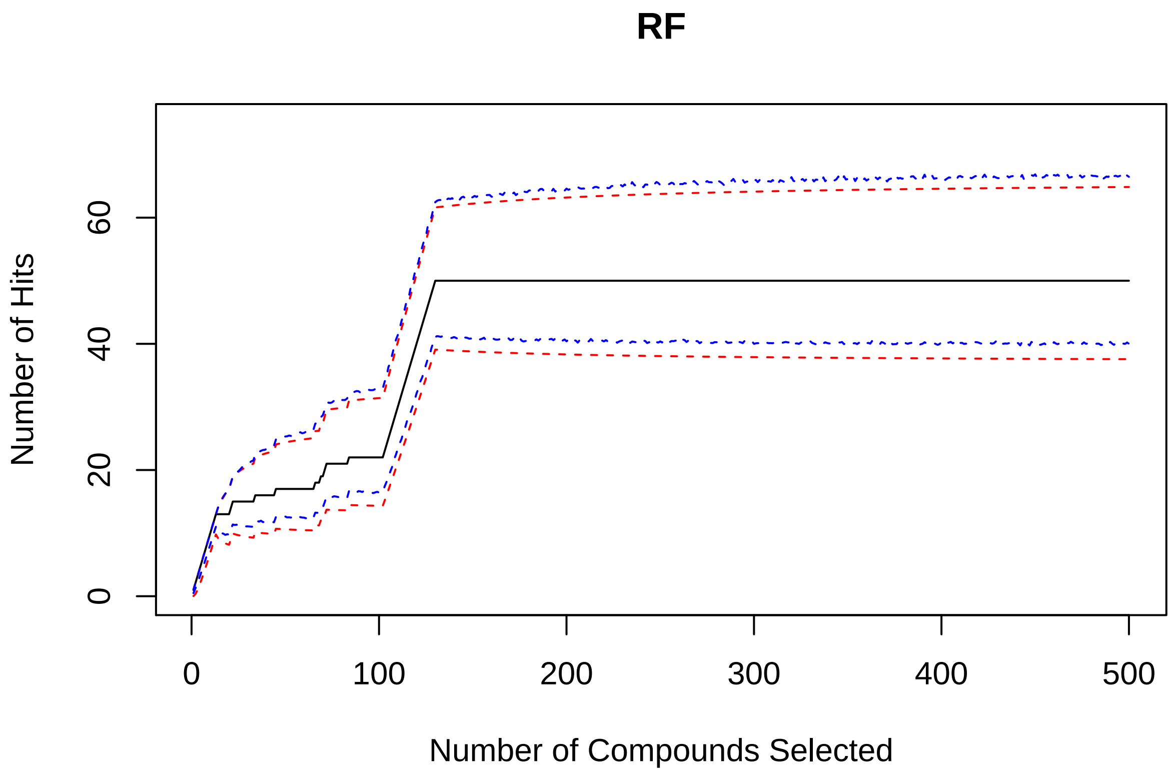


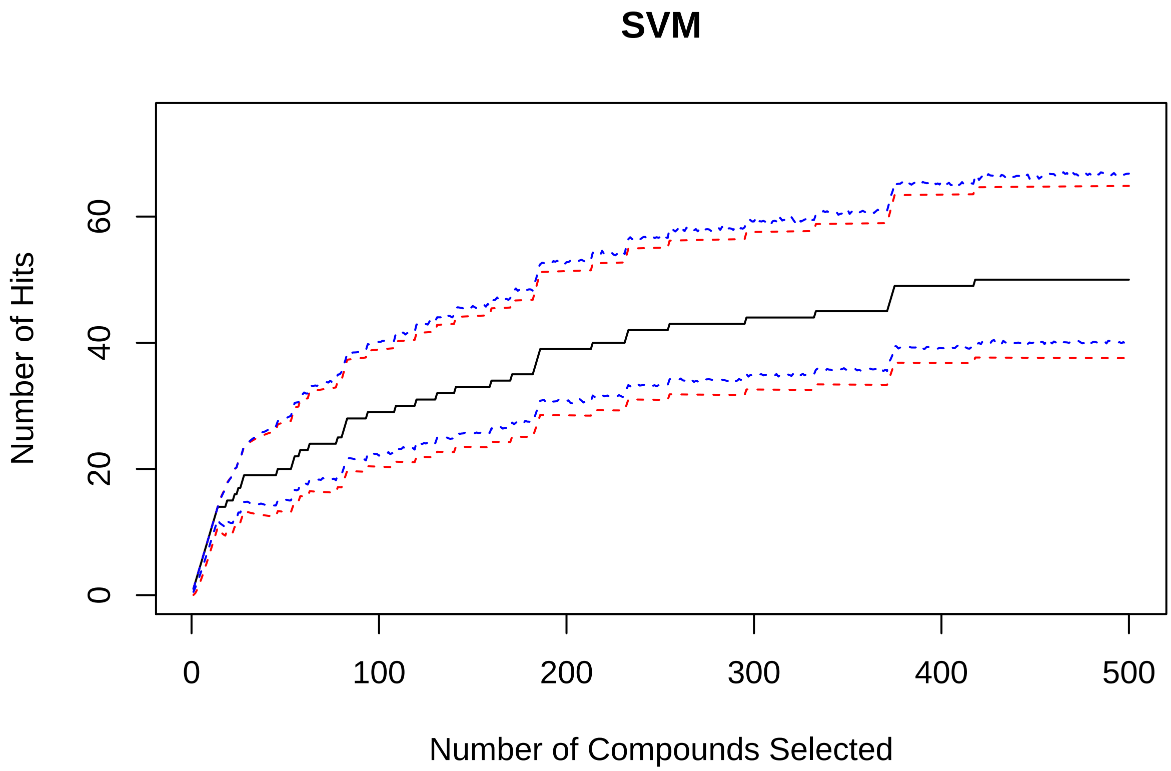
Bayesian credible intervals with more informative prior

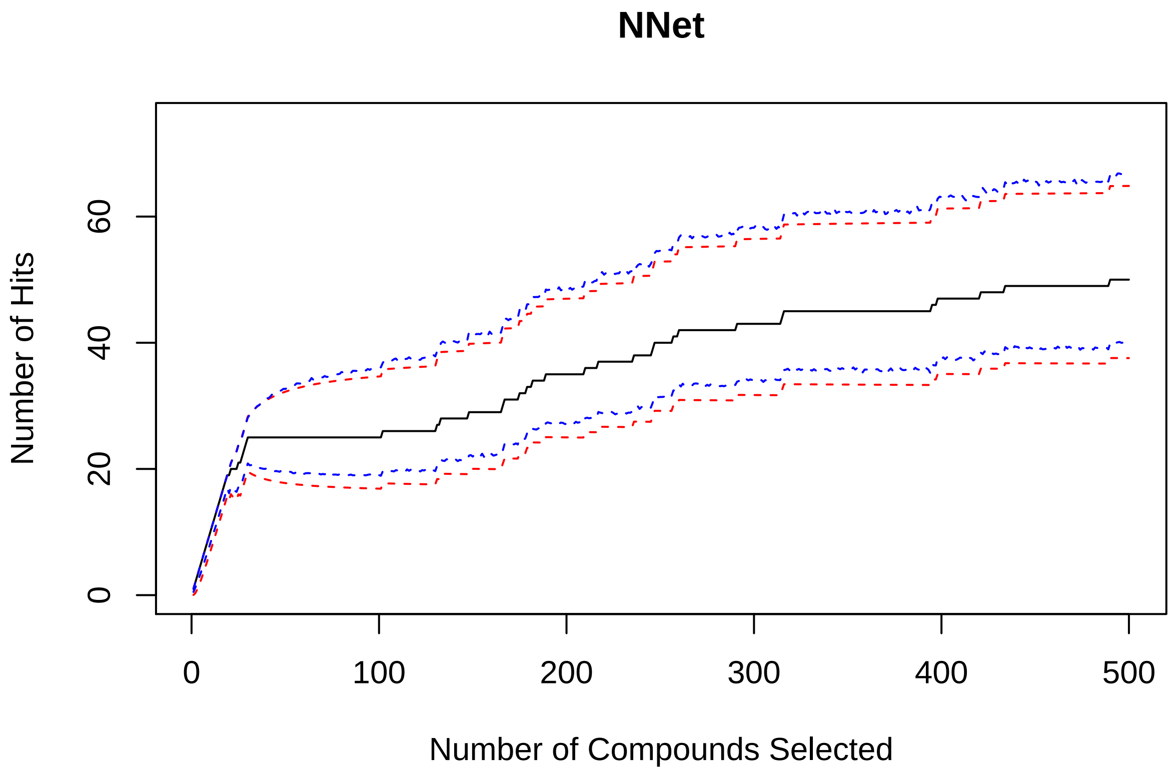
Early on in the curve I am now putting much higher probability that θ_{ij} is close to 1. Later on in the curve the sample size is so large that the prior doesn't have any influence on the posterior, so I am using close to the Jeffrey's prior at this point.

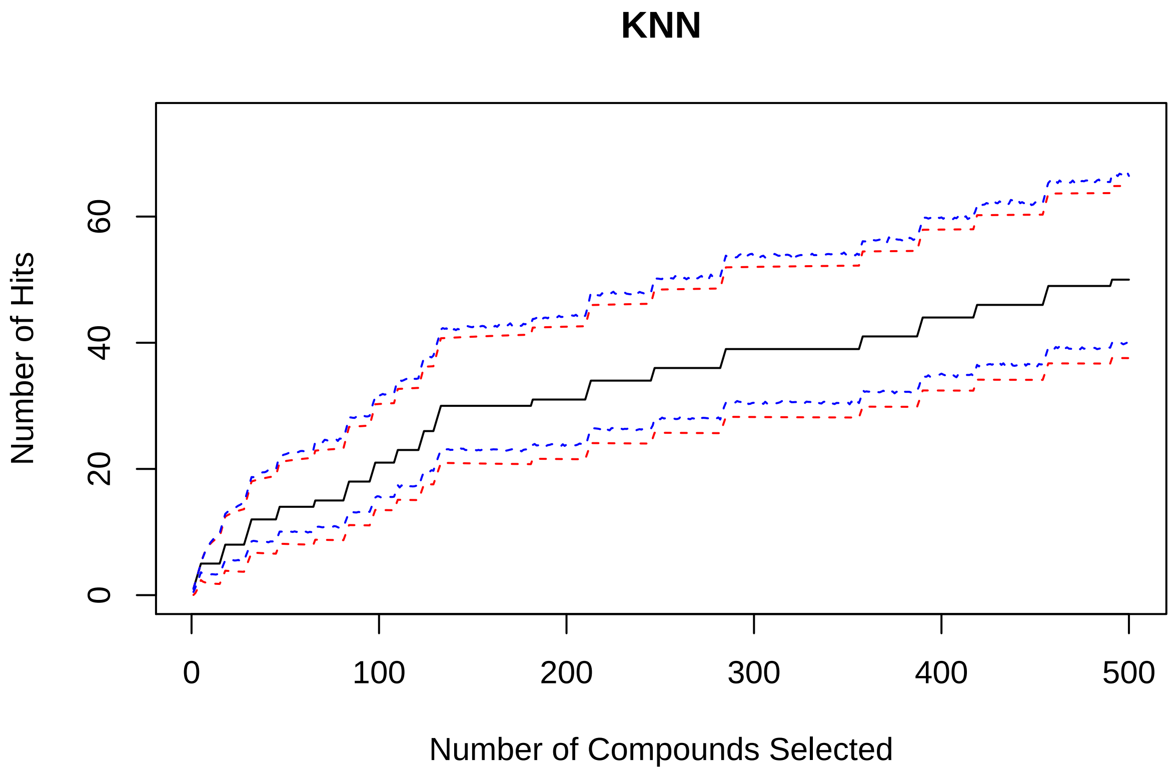
Blue CIs are Bayesian, Red are Frequentist.

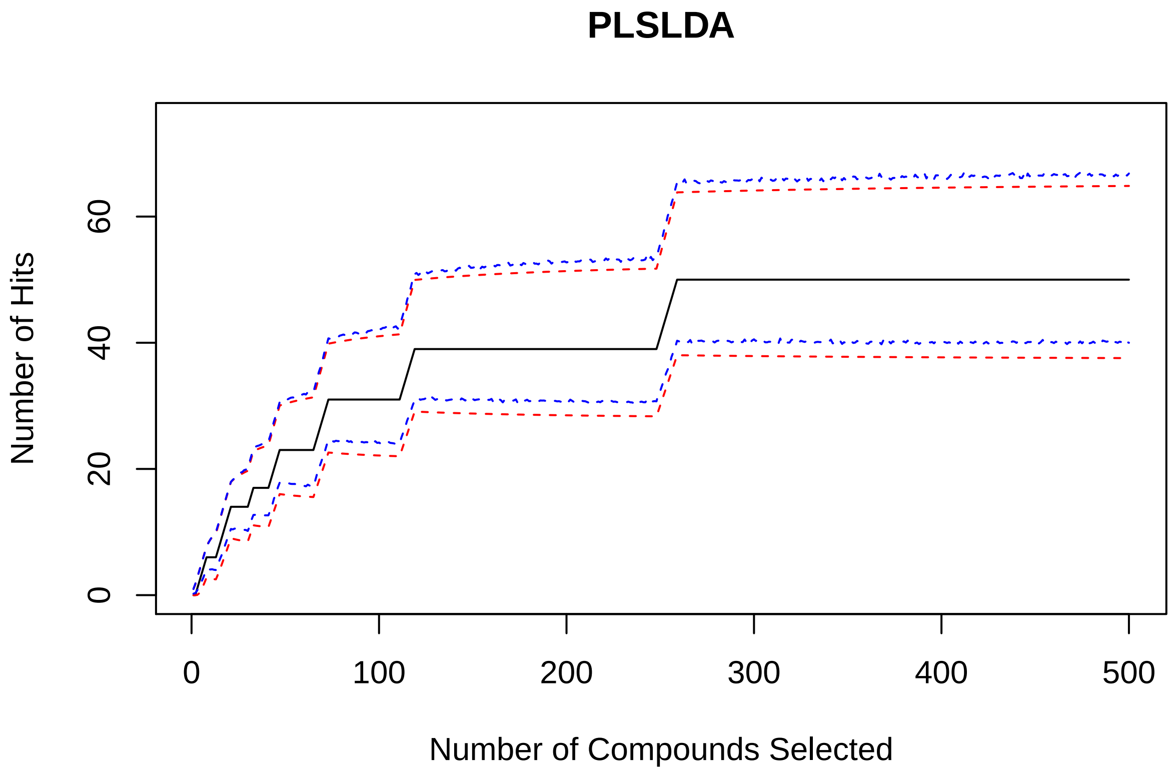
```
modelType <- c("Tree", "RF", "SVM", "NNet", "KNN", "PLSLDA")

u <- 1:500
i <- 2
for (i in 2:7) {

  hit.vec <- probs.Burd$Observed
  probs <- probs.Burd[, i]

  order.idx <- order(probs, decreasing = T)
  probs <- probs[order.idx]
  hit.vec <- hit.vec[order.idx]

  m <- length(probs)
  int.mat <- matrix(ncol = 3, nrow = m)
  colnames(int.mat) <- c("NHits", "LB", "UB")

  a <- vector(length = 500)
  b <- vector(length = 500)
  for(j in 1:500) {
    a[j] <- sum(hit.vec[1:j]) + .5 + .005 * (500-i)
    b[j] <- j - sum(hit.vec[1:j]) + .5
  }

  sum.samp <- vector(length = 10000)
  for(j in 1:500) {
    int.mat[j, 1] <- sum(hit.vec[1:j])
    sum.samp <- rbeta(10000, a[j], b[j]) * j
    int.mat[j, 2] <- quantile(sum.samp, probs = .025)
    int.mat[j, 3] <- quantile(sum.samp, probs = .975)
  }

  plot(int.list.burd[[i-1]][, 1], type = "l", ylim = c(0, 75),
       main = colnames(probs.Burd)[i+1], ylab = "Number of Hits",
       xlab = "Number of Compounds Selected")
  lines(int.list.burd[[i-1]][, 2], type = "l", lty = "dashed", col = "red")
  lines(int.list.burd[[i-1]][, 3], type = "l", lty = "dashed", col = "red")

  lines(int.mat[, 2], type = "l", lty = "dashed", col = "blue")
  lines(int.mat[, 3], type = "l", lty = "dashed", col = "blue")

}
```

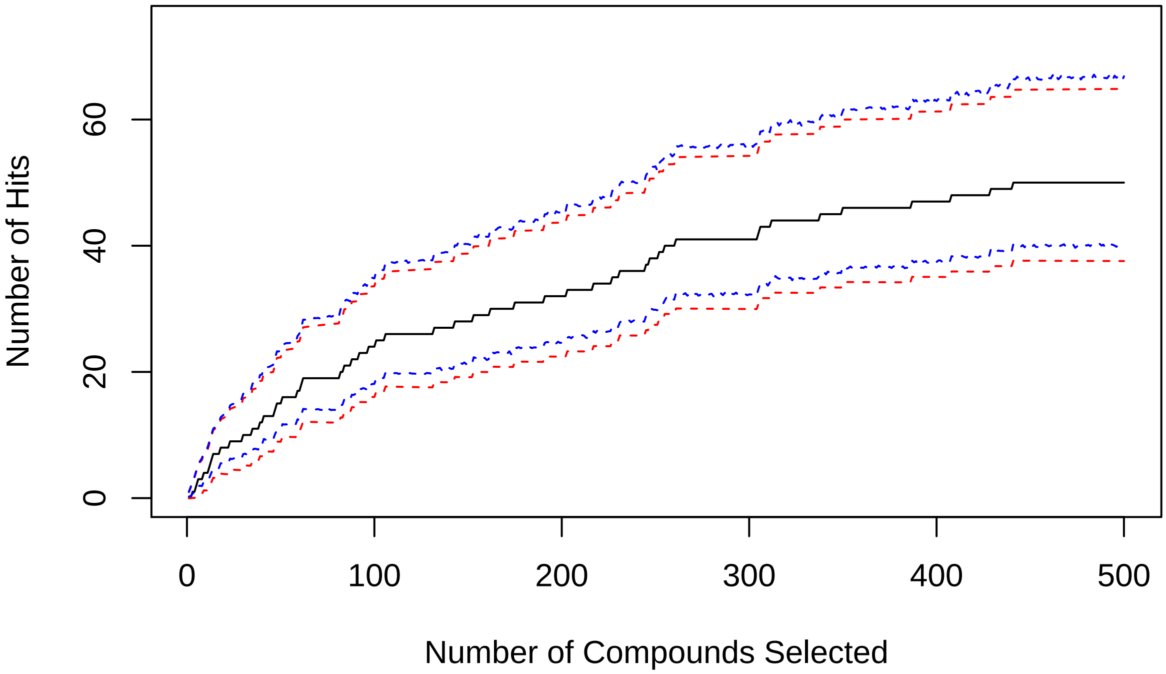



SVM
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


NNet
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


KNN
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


PLSLDA
Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected


Number of Hits
0
20
40
60
0
100
200
300
400
500
Number of Compounds Selected