

# Analyses for stickleback landscape parasitology study

Pascal I. Hablützel and Io S. Deflem

2022-11-27

## 1. Description

This R Markdown document describes the analyses performed for the manuscript entitled “Environmental pollution correlates with parasite infection across a riverine landscape” by Io S. Deflem, Seppe Marchand, Federico C.F. Calboli, Joost A.M. Raeymaekers, Filip A.M. Volckaert and Pascal I. Hablützel.

The analyses were run in R 4.2.2

## 2. Study area and sampling

Up to thirty 0+ three-spined sticklebacks were sampled at 37 locations in the Dijle and Demer basins in Belgium during autumn 2016 under a permit of the Flemish Agency Nature and Forest. Both basins together cover a continuous surface area of 3,624 km<sup>2</sup> with the furthest two sampling sites being located 117 km apart (distance measured along rivers). All locations included small and relatively slow flowing streams (drop off from highest to lowest point is 18 m) covering a wide range of ecological, hydromorphological, and physico-chemical characteristics. Fish were caught using a dip net.

## 3. Setting up working environment

```
# Empty environment
rm(list = ls())

# Print version of R
cat("This script was run with:", version[["version.string"]], "\n")

## This script was run with: R version 4.1.2 (2021-11-01)
# Set working directory to location where script is stored
setwd(dirname(rstudioapi::getActiveDocumentContext()$path)) # requires installation of package 'rstudioapi'

# Loading required libraries
require(BAS)
cat("BAS version", getNamespaceVersion("BAS"), "\n")

## BAS version 1.6.4

require(boral)
cat("boral version", getNamespaceVersion("boral"), "\n")

## boral version 2.0

require(car)
cat("car version", getNamespaceVersion("car"), "\n")

## car version 3.1-2
```

```

require(corrplot)
cat("corrplot version", getNamespaceVersion("corrplot"), "\n")

## corrplot version 0.92

require(ggplot2)
cat("ggplot2 version", getNamespaceVersion("ggplot2"), "\n")

## ggplot2 version 3.4.2

require(gplots)
cat("gplots version", getNamespaceVersion("gplots"), "\n")

## gplots version 3.1.3

require(vegan)
cat("vegan version", getNamespaceVersion("vegan"), "\n")

## vegan version 2.6-4

```

## 4 Loading and preparing host and parasite data

Fish were euthanized with a lethal dose of MS222 on the day of capture, following directions of the KU Leuven Animal Ethics Commission, and stored at -20 °C. Fish were kept in separate containers per site at all times. Lab based parasite screening of thawed fish involved placing individual fish in 5 or 10 ml cryo-tubes with 1 or 2 ml of distilled water. Following a vigorous shake of 10 s, the liquid was poured into a Petri dish and ectoparasites were identified and counted using a stereomicroscope. Fish were rinsed and checked again for the presence of ectoparasites on skin and fins. The intestines were examined for endoparasites. Before dissection, fish weight ( $\pm 0.1$  mg) and standard length ( $\pm 1$  mm) were recorded. To quantify body condition, we calculated the scaled mass index (SMI; Maceda-Veiga et al., 2014; Peig & Green, 2009). Sex was determined during dissection by inspection of gonad development. A total of 668 fish were dissected, which amounts to approximately 20 fish per location, with the exception of seven locations where only 10 fish were screened for the presence of macroparasites. Ecto- and endoparasites were morphologically identified to species level whenever possible.

```

# Parasite data
data <- read.csv("data_2016_2303.csv", sep = ";")
data$site <- as.factor(data$site)

# Calculate parasite parameters names(data) Parasite data is overdispersed
# (mostly so for Trichodina), if using average abundance data, species matrix
# needs to be transformed
dataao <- na.omit(data[, c(1, 22:24, 26:32)]) # Remove individual fish for which no parasites were counted

ddata <- dispweight(dataao[, -1]) # Correct for overdispersion of the parasite count data
avab <- aggregate(ddata, by = list(dataao[, "site"]), function(x) {
  mean(x, na.rm = T)
}) # Calculate average abundances per site
prev = aggregate(data[, c(22:24, 26:32)], by = list(data[, "site"]), function(x) {
  sum(x > 0, na.rm = T)/length(x)
}) # Calculate prevalence per site
medin = aggregate(data[, c(22:24, 26:32)], by = list(data[, "site"]), function(x) {
  median(x[x > 0], na.rm = T)
}) # Calculate median infection intensity per site
pa = aggregate(data[, c(22:24, 26:32)], by = list(data[, "site"]), function(x) {
  ifelse(mean(x, na.rm = T) > 0, 1, 0)
})

```

```

}) # Assess presence/absence per site

avab[is.na(avab)] <- 0
prev[is.na(prev)] <- 0
medin[is.na(medin)] <- 0

# Host condition data
avcondition <- aggregate(data$SMI, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2] # Calculate average host condition (SMI) per site
avlength <- aggregate(data$length, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2] # Calculate average host length per site

# Parasite index
sgyr <- 1:nrow(data)
stri <- 1:nrow(data)
sglu <- 1:nrow(data)
scon <- 1:nrow(data)
scysl <- 1:nrow(data)
spro <- 1:nrow(data)
saca <- 1:nrow(data)
scam <- 1:nrow(data)
sang <- 1:nrow(data)

for (j in 1:nrow(data)) {
  sgyr[j] <- data$Gyr[j]/sd(data$Gyr, na.rm = T)
  stri[j] <- data$Tri[j]/sd(data$Tri, na.rm = T)
  sglu[j] <- data$Glu[j]/sd(data$Glu, na.rm = T)
  scon[j] <- data$Con[j]/sd(data$Con, na.rm = T)
  scysl[j] <- data$CysL[j]/sd(data$CysL, na.rm = T)
  spro[j] <- data$Pro[j]/sd(data$Pro, na.rm = T)
  saca[j] <- data$Aca[j]/sd(data$Aca, na.rm = T)
  scam[j] <- data$Cam[j]/sd(data$Cam, na.rm = T)
  sang[j] <- data$Ang[j]/sd(data$Ang, na.rm = T)
}

PI <- 1:nrow(data)
for (j in 1:nrow(data)) {
  PI[j] <- 10/max(sgyr, na.rm = T) * sgyr[j] + 10/max(stri, na.rm = T) * stri[j] +
    10/max(sglu, na.rm = T) * sglu[j] + 10/max(scon, na.rm = T) * scon[j] + 10/max(scysl,
    na.rm = T) * scysl[j] + 10/max(spro, na.rm = T) * spro[j] + 10/max(saca,
    na.rm = T) * saca[j] + 10/max(scam, na.rm = T) * scam[j] + 10/max(sang, na.rm = T) *
    sang[j]
}

PI_ecto <- 1:nrow(data)
for (j in 1:nrow(data)) {
  PI_ecto[j] <- 10/max(sgyr, na.rm = T) * sgyr[j] + 10/max(stri, na.rm = T) * stri[j] +
    10/max(sglu, na.rm = T) * sglu[j]
}

PI_endo <- 1:nrow(data)

```

```

for (j in 1:nrow(data)) {
  PI_endo[j] <- 10/max(scon, na.rm = T) * scon[j] + 10/max(scysl, na.rm = T) *
    scysl[j] + 10/max(spro, na.rm = T) * spro[j] + 10/max(saca, na.rm = T) *
    saca[j] + 10/max(scam, na.rm = T) * scam[j] + 10/max(sang, na.rm = T) * sang[j]
}

avPI <- aggregate(PI, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2] # Calculate average parasite index per site
avPI_ecto <- aggregate(PI_ecto, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2] # Calculate average parasite index for ectoparasites per site
avPI_endo <- aggregate(PI_endo, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2] # Calculate average parasite index for endoparasites per site

```

## 5 Loading and preparing environmental and spatial data

Physico-chemical data was provided by the Flemish Environmental Agency (VMM). Each fish sampling site was chosen at or near an environmental monitoring site of VMM. Water parameters include water temperature, pH, conductivity, dissolved oxygen (O<sub>2</sub>), saturation with dissolved oxygen, and Biochemical and Chemical Oxygen Demand (BOD and COD). Nutrient related water parameters include measurements of nitrate (NO<sub>3</sub>-), nitrite (NO<sub>2</sub>-), Kjeldahl nitrogen (KjN), total nitrogen (Nt), ammonium (NH<sub>4</sub><sup>+</sup>), and total phosphorus (Pt). Following removal of strong collinear variables (significant correlation with P < 0.05 and Pearson correlation coefficient > |0.6|; Dormann et al., 2013), six environmental physico-chemical variables were retained (temperature, conductivity, COD, saturation with dissolved oxygen, ammonium, and total nitrogen), representing different aspects of water quality. For each water parameter, the average value of the year before sampling was calculated based on monthly monitoring data. Additionally, two hydromorphological variables were included: the presence of a pool-riffle pattern and meanders were noted during field sampling and these parameters were included as binary variables (presence/absence) for a representation of abiotic habitat structure. Spatial (waterway) distances were calculated using the Network Analyst toolbox in ArcGIS. Upstream distance was defined as the maximal upstream distance from the sampling location. Network peripherality was calculated as the average waterway distance of a sampling location to all other locations. Hence, a total of eight environmental and two spatial variables were included in the statistical analysis.

```

# Environmental data (VMM)
environment <- read.csv("Environment_update.csv", sep = ";")
spavar <- read.csv("space2.csv", sep = ";") # spatial variables: network centrality and upstream distance
# plot(spavar$netcen); plot(density(spavar$netcen)) plot(spavar$updist);
# plot(density(spavar$updist))

# Environmental data (from field observations)
field_data <- read.csv("field_data.csv", sep = ",")
environment2 <- cbind(environment[, c(1, 49, 52:53, 55, 57, 60, 63)], field_data[-c(8,
  10, 25, 27, 37), 33:34], spavar[, c(2, 3)])
environment2$pool_riffle <- as.factor(environment2$pool_riffle)
environment2$meander <- as.factor(environment2$meander)

netcen <- spavar$netcen
updist <- spavar$updist

```

We used univariate generalized linear models to investigate how landscape-level effects modify infection patterns of individual parasite taxa, host size and condition. We kept the statistical models linear (as opposed to polynomial) and only considered main effects (i.e. no interaction terms) because we had no prior

information from this study system that more complex models were necessary and because the study design with (only) 37 sampling sites was not intended for non-linear models. Ten explanatory variables (temperature, conductivity, COD, saturation with dissolved oxygen, ammonium, total nitrogen, the presence of pool-riffle patterns and meanders, upstream distance, and network peripherality) were included.

## 6. Univariate analysis using Bayesian Model Averaging

Univariate analyses - We used generalized linear models in a BMA approach to understand how infection with individual parasite taxa relate to host characteristics (length and condition), environmental conditions as well as spatial distance among sampling sites. Parasite infection was calculated in three ways at the host population level: average abundance (mean parasites per host), prevalence (percentage of infected hosts) and median infection intensity (median number of parasites in infected hosts). We calculated the individual parasitisation index (IPI) following Kalbe et al. (2002) as a measurement for total parasite abundance and species richness for each individual fish. This index was calculated for all parasite species combined, and for ecto- and endoparasite species separately. For these models, we assumed a normal error distribution (which appeared to be justified, see Supplementary Figures S1-S2) and applied a Jeffrey-Zellner-Siow prior. Model assumptions (homoscedasticity of the variances and normal distribution of the errors) were assessed using the generic model plot function in R and did only not clearly deviate in any of the models for rare parasites. We followed a normal distribution, and not Poisson or negative binomial, for the parasite data for the common species (Trichodina sp. and Gyrodactylus spp.) and the individual parasitisation index as the parameters used are deviates from count data. Rare parasites (Glugea, Contracaecum, Anguillicoloides, and unidentified cysts) were excluded from the univariate analysis because there was not enough data to obtain a good fit of the models. For rare parasites (Contracaecum sp. and Anguillicoloides crassus), we used population-level presence-absence data assuming a binomial error distribution and a uniformly distributed BIC prior. Due to low prevalences, the other parasites were not included in the species-specific models. Explanatory variables were considered important when they had a posterior inclusion probability (PIP) of 0.5. To account for overdispersion in the parasite counts, we transformed the data by downweighting overdispersed taxa following Clarke et al. (2006) using the dispweight function in the R package vegan v2.5.6 (Oksanen et al., 2013).

### 6.0.1 Figure 2

```
# Make a matrix for PIP (Posterior Inclusion Probability)
PIP <- matrix(nrow = 12, ncol = 14)
rownames(PIP) <- c("Host condition", "Host length", "Temperature", "Oxygen saturation",
  "Conductivity", "COD", "Ammonium", "Total nitrogen", "Pool riffle pattern", "Meander",
  "Network peripherality", "Upstream distance")
colnames(PIP) <- c("Host condition", "Host size", "Gyrodactylus abundance", "Gyrodactylus prevalence",
  "Gyrodactylus infection intensity", "Trichodina abundance", "Trichodina prevalence",
  "Trichodina infection intensity", "Glugea", "Contracaecum", "Aguillicola", "PI",
  "PI ecto", "PI endo")

# Condition
bas.model <- bas.lm(avcondition ~ T_av + O2_sat_av + Con_av + COD_av + NH4_.av +
  Nt_av + pool_riffle + meander + netcen + updist, data = environment2, prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(3:12), 1] <- pip[2:11, 1] * sign(coef.model$postmean[2:11])

# Length
bas.model <- bas.lm(avlength ~ T_av + O2_sat_av + Con_av + COD_av + NH4_.av + Nt_av +
  pool_riffle + meander + netcen + updist, data = environment2, prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(3:12), 2] <- pip[2:11, 1] * sign(coef.model$postmean[2:11])
```

```

# Gyrodactylus abundance
bas.model <- bas.lm(avab$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Gyrodactylus prevalence
bas.model <- bas.lm(prev$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 4] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Gyrodactylus infection intensity
bas.model <- bas.lm(medin$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 5] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Trichodina abundance
bas.model <- bas.lm(avab$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 6] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Trichodina prevalence
bas.model <- bas.lm(prev$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 7] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Trichodina infection intensity
bas.model <- bas.lm(medin$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 8] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Glugea
bas.model <- bas.glm(pa$Glu ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  betaprior = g.prior(100), family = binomial)
coef.model <- coef(bas.model)

```

```

pip <- summary(bas.model)
PIP[c(1:12), 9] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Contracaecum
bas.model <- bas.glm(pa$Con ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  betaprior = g.prior(100), family = binomial)
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 10] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# Anguillicola
bas.model <- bas.glm(pa$Ang ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  betaprior = g.prior(100), family = binomial)
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 11] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# PI
bas.model <- bas.lm(avPI ~ avlength + avcondition + T_av + O2_sat_av + Con_av + COD_av +
  NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 12] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# PI ecto
bas.model <- bas.lm(avPI_ecto ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 13] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

# PI endo
bas.model <- bas.lm(avPI_endo ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
coef.model <- coef(bas.model)
pip <- summary(bas.model)
PIP[c(1:12), 14] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

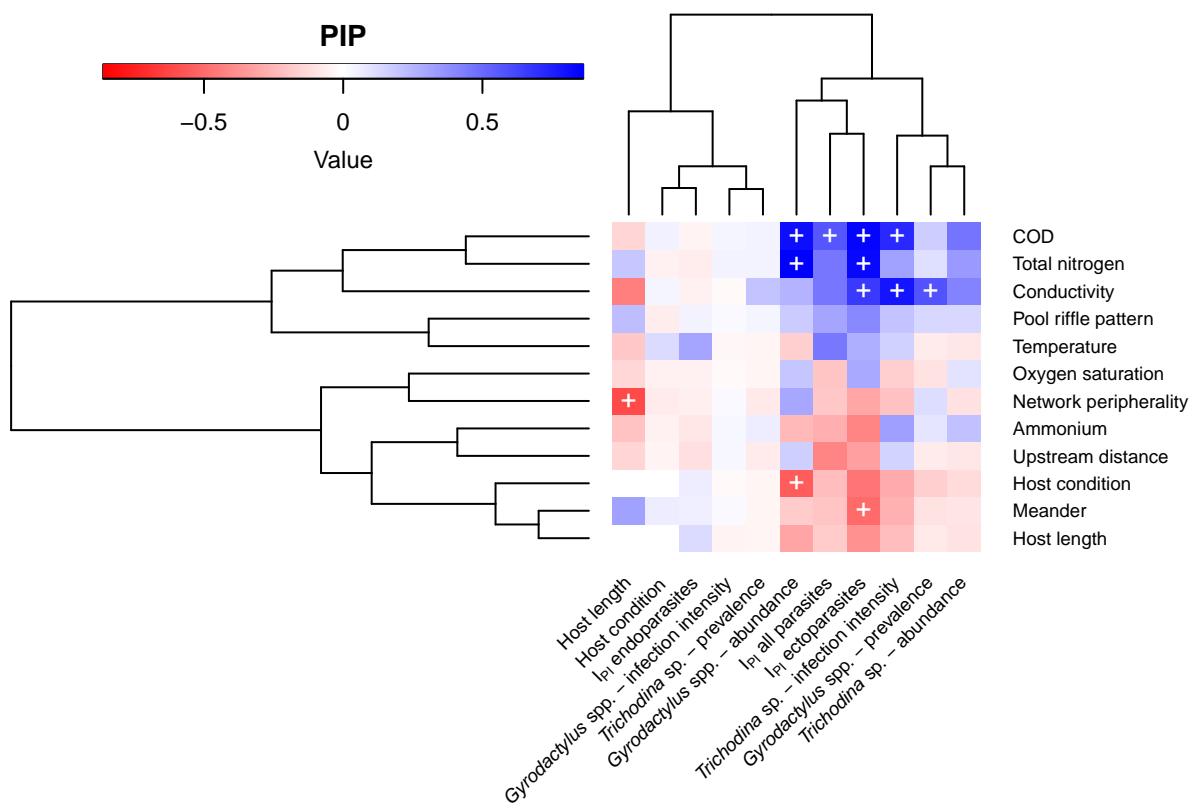
x = round(PIP, digits=2)
x[abs(PIP)<0.5] <- ""
x[abs(PIP)>0.5] <- "+"
heatmap.2(PIP[,-c(9,10,11)],
  cellnote = x[,-c(9,10,11)],
  #main = "Correlation",
  notece=1,
  notecol="white",
  density.info="none",
  trace="none",
  margins =c(10,8),

```

```

col=redblue(256),
dendrogram="both",
cexRow = 0.7,
cexCol = 0.7,
key.title = "PIP",
lhei = c(1,3),
lwid = c(0.5, 0.5),
srtCol = 45,
labCol = c("Host condition", "Host length", expression(paste(italic("Gyrodactylus"), " spp.")),
#Colv="NA"
)

```



```

pdf("Figure2.pdf", width = 7.29, height = 4.5)
heatmap.2(PIP[,-c(9,10,11)],
  cellnote = x[,-c(9,10,11)],
  #main = "Correlation",
  notece=1,
  notecol="white",
  density.info="none",
  trace="none",
  margins =c(10,8),
  col=redblue(256),
  dendrogram="both",
  cexRow = 0.7,
  cexCol = 0.7,
  key.title = "PIP",

```

```

lhei = c(1,3),
lwid = c(0.5, 0.5),
srtCol = 45,
labCol = c("Host condition", "Host length", expression(paste(italic("Gyrodactylus")), " spp.")),
#Colv="NA"
)

dev.off()

## pdf
## 2

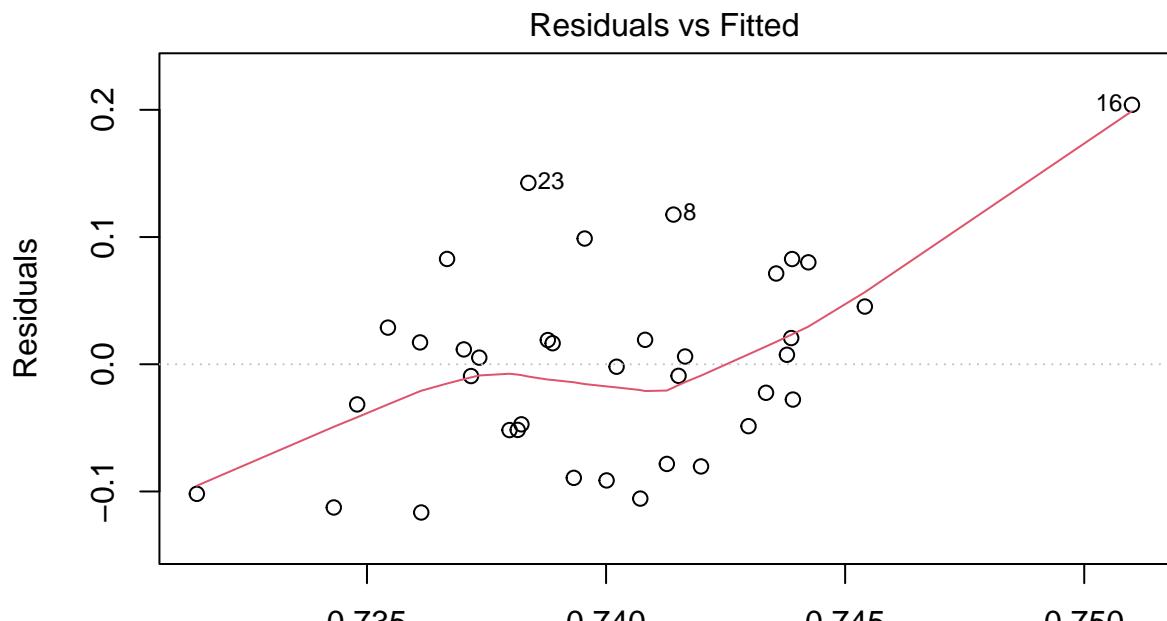
```

## 6.1 Variation in host condition

```

bas.model <- bas.lm(avcondition ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av +
                     Nt_av + pool_riffle + meander + netcen + updist, data = environment2, prior = "JZS")
plot(bas.model)

```

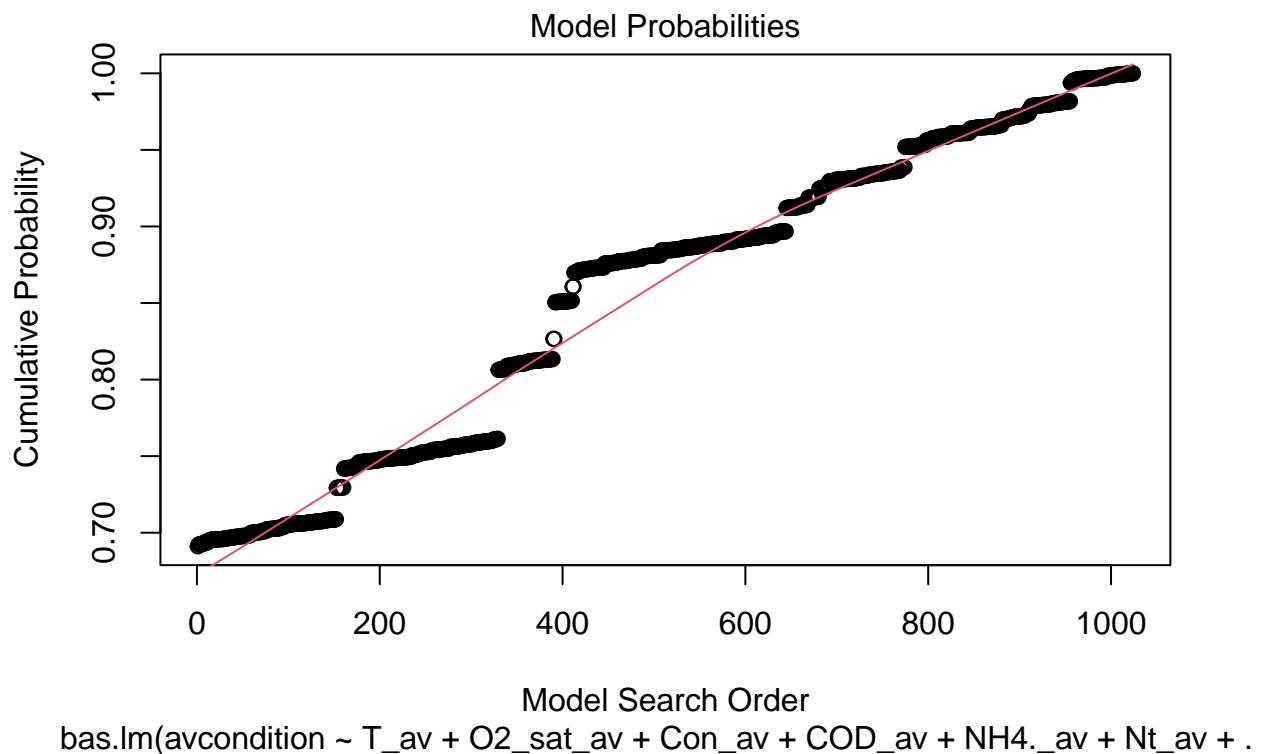


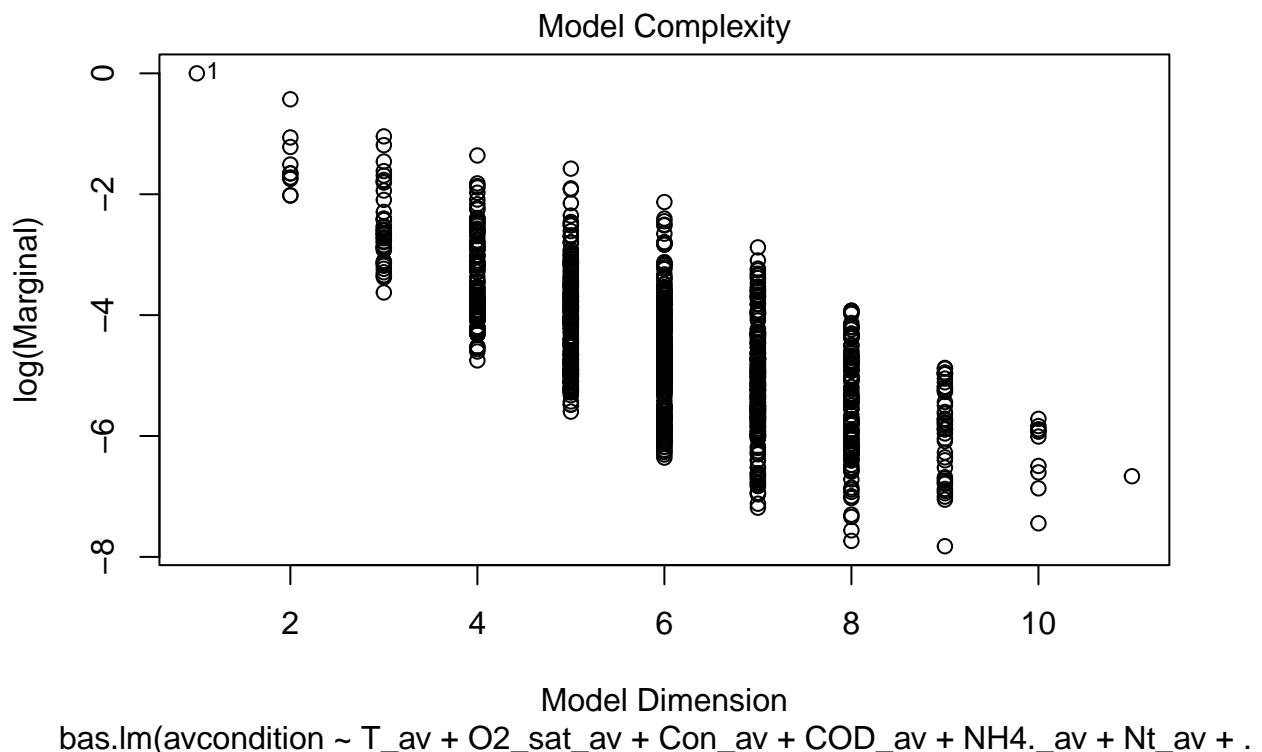
Predictions under BMA

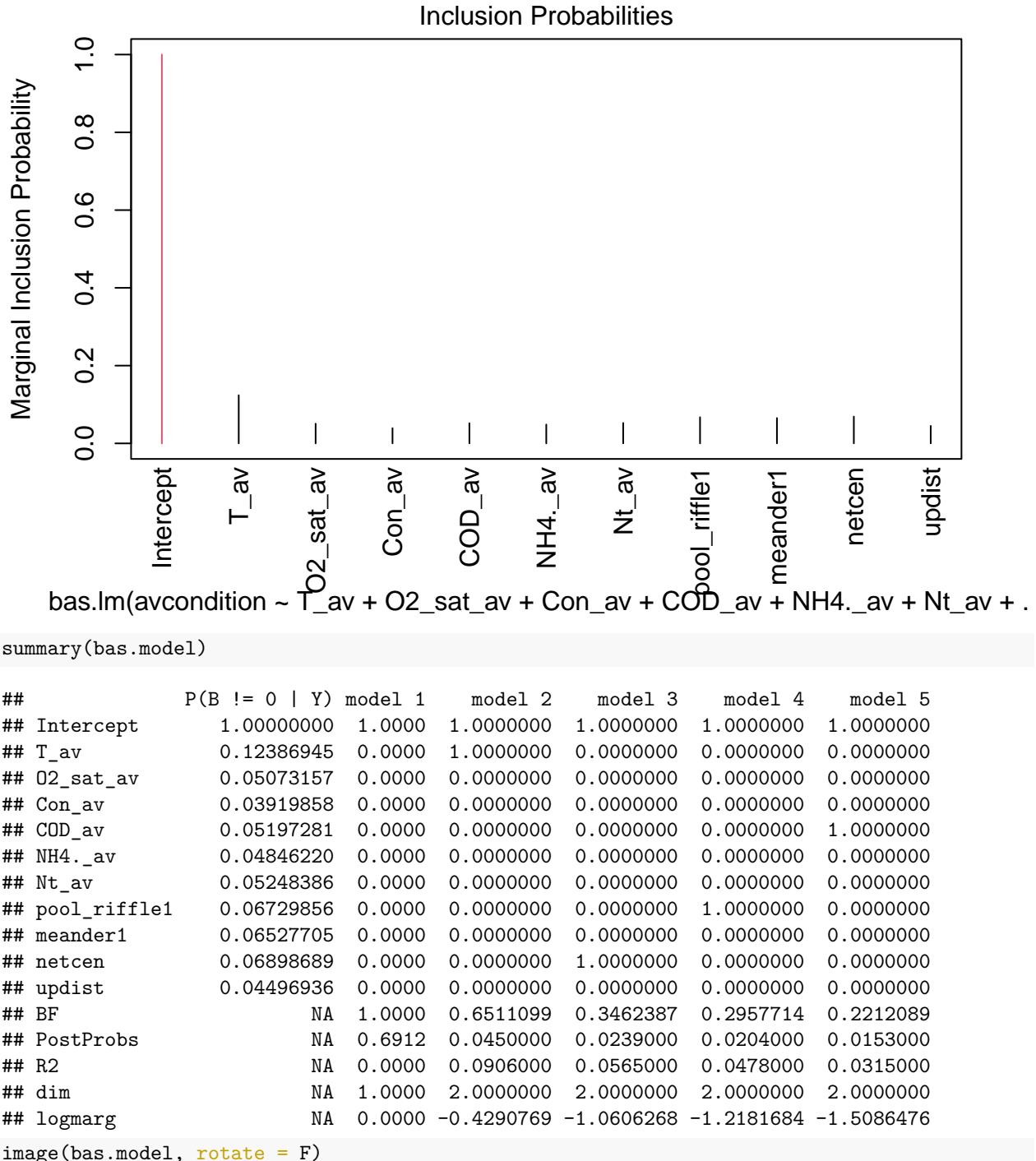
```

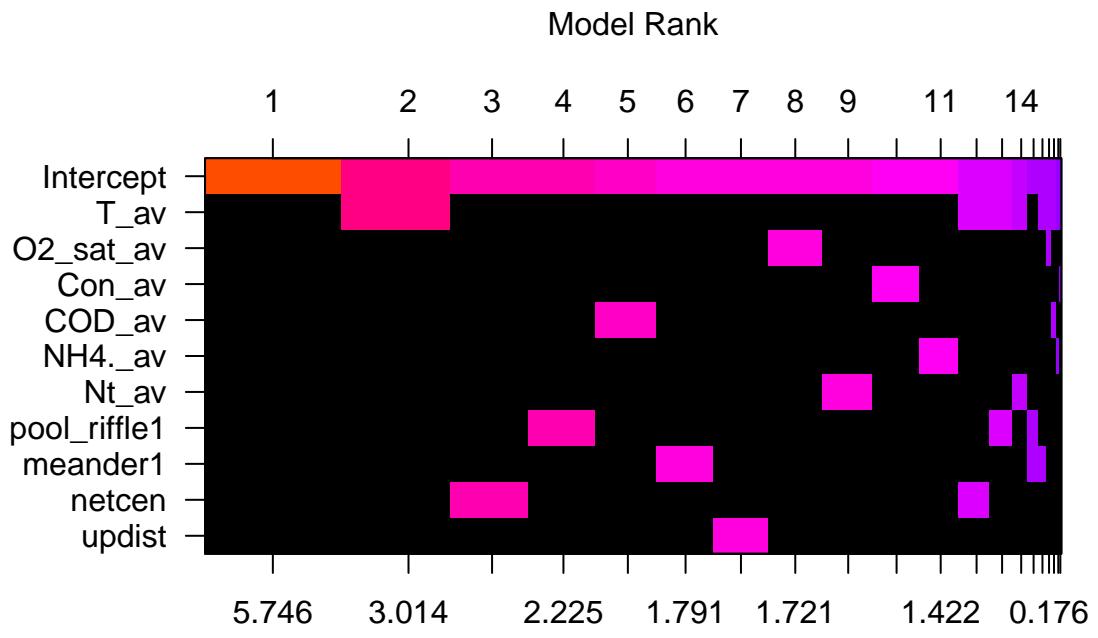
bas.lm(avcondition ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + .

```









### Log Posterior Odds

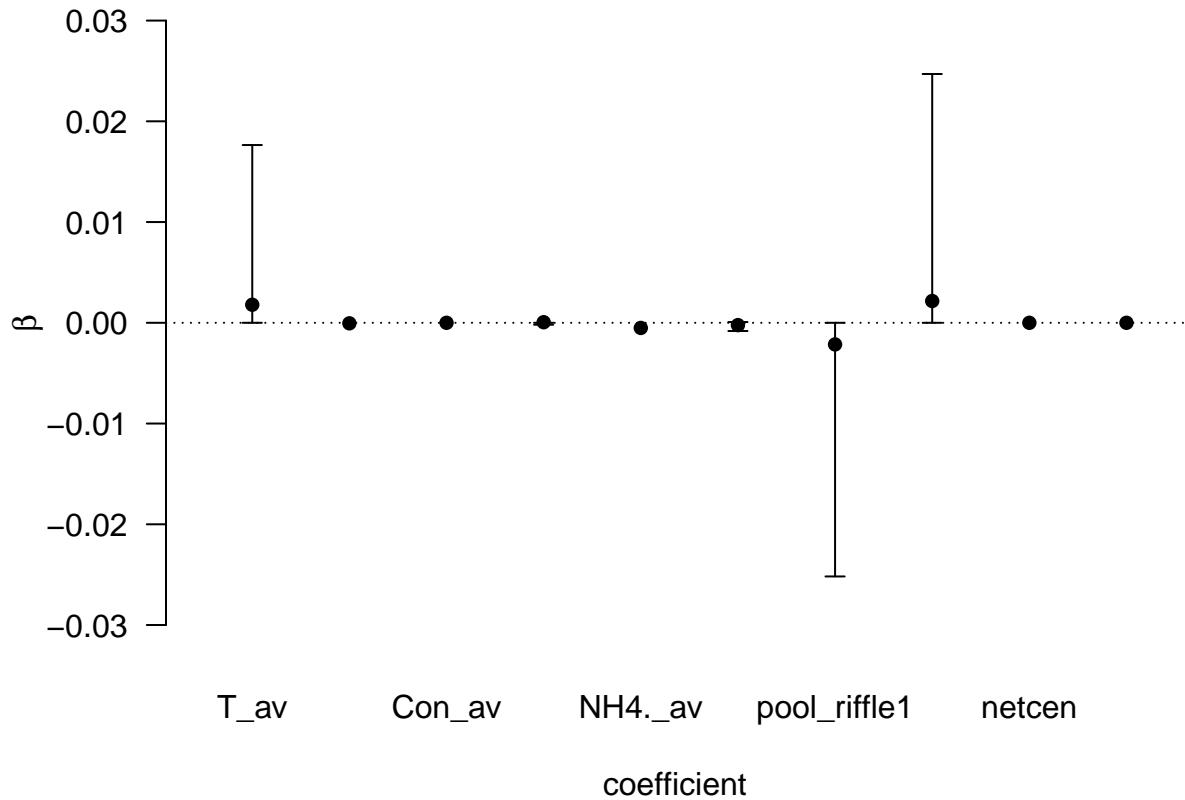
```

coef.model <- coef(bas.model)
# abs(coef.model$postmean)-2*coef.model$postsd > 0
plot(confint(coef.model, parm = 2:11))

## Warning in arrows(x[not.deg], ci[not.deg, 1], x[not.deg], ci[not.deg, 2], :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x[not.deg], ci[not.deg, 1], x[not.deg], ci[not.deg, 2], :
## zero-length arrow is of indeterminate angle and so skipped

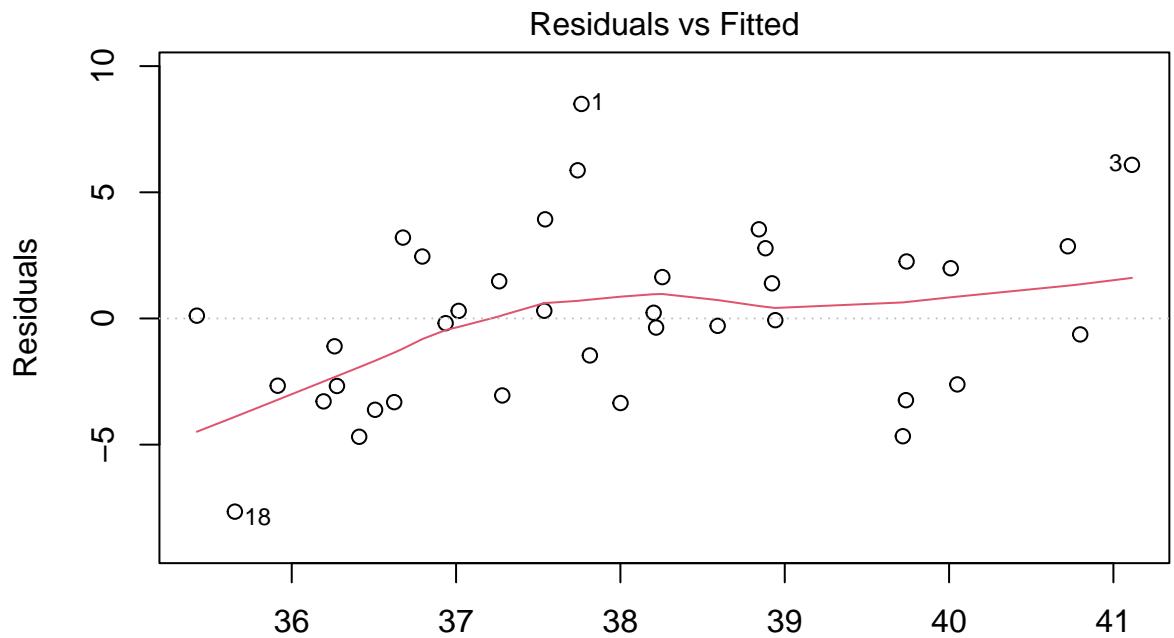
```



```
## NULL
confint <- confint(coef.model, parm = 2:11)
write.table(confint, "condition.txt", sep = "\t")
pip <- summary(bas.model)
PIP[c(3:12), 1] <- pip[2:11, 1] * sign(coef.model$postmean[2:11])
# coef.model$postmean[2:11]
```

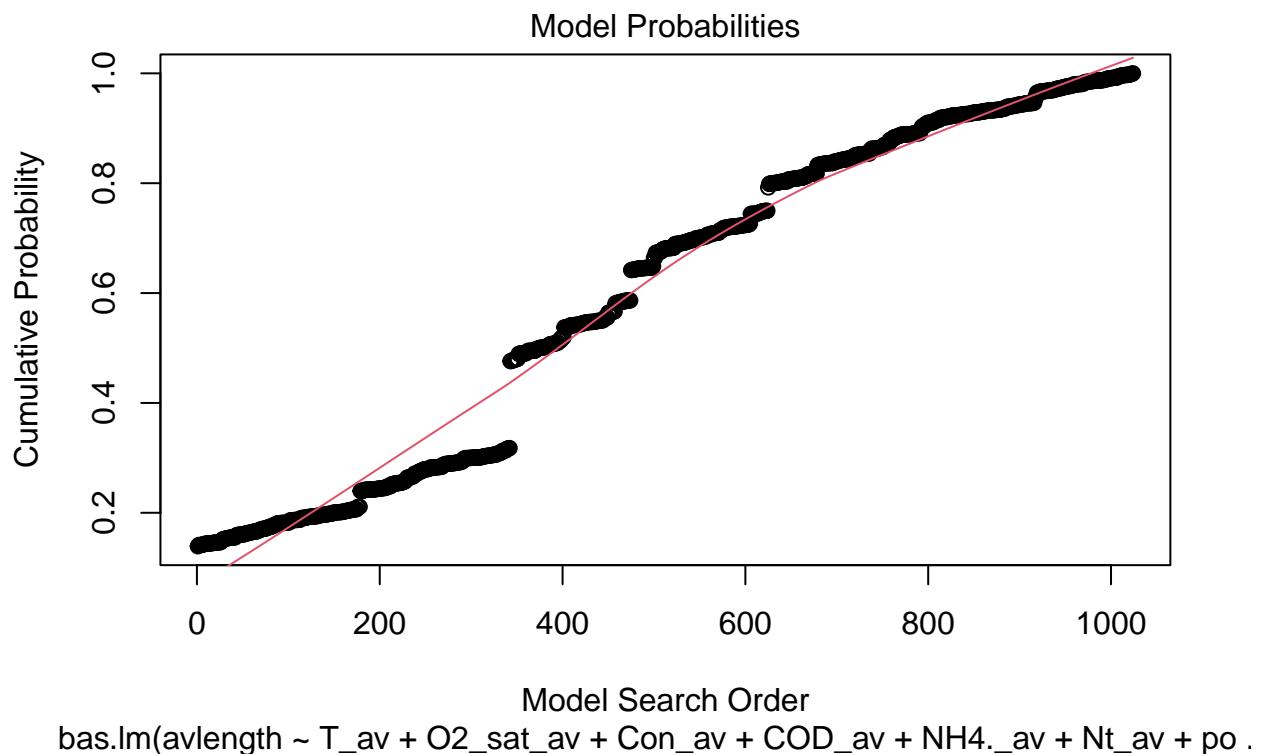
## 6.2 Variation in host length

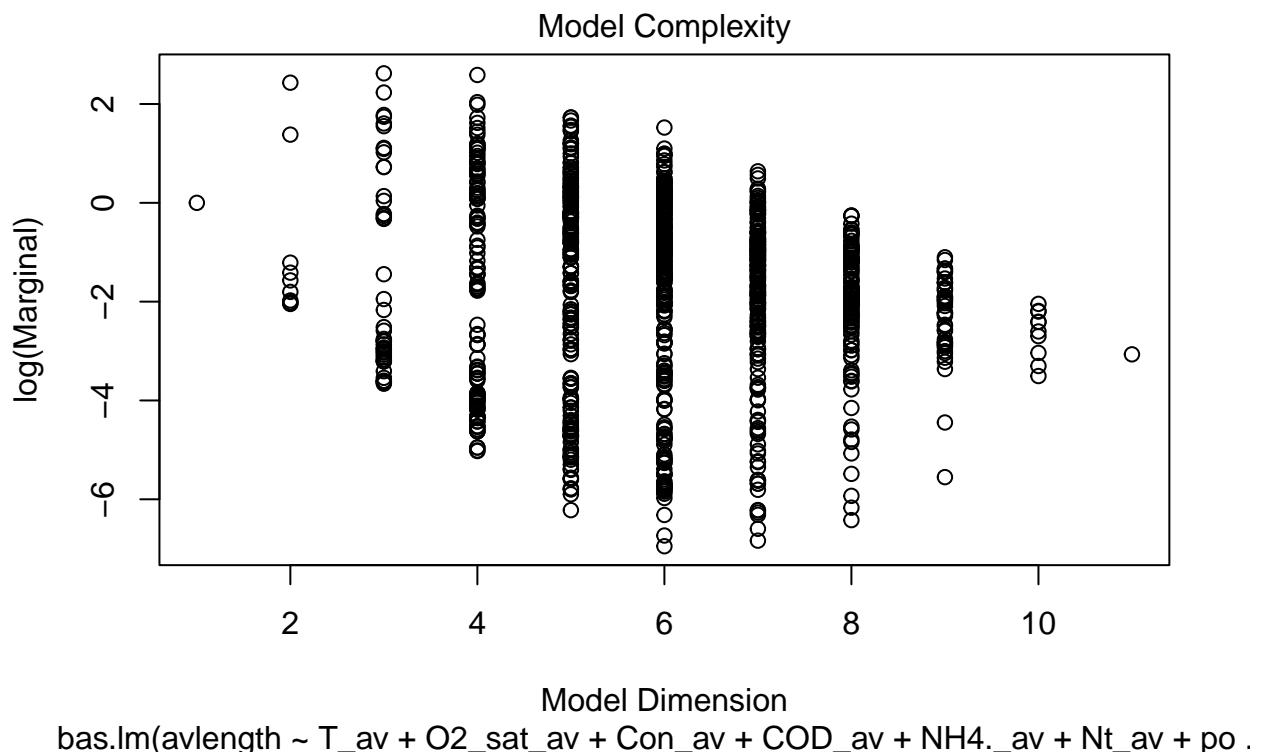
```
bas.model <- bas.lm(avlength ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av +
  pool_riffle + meander + netcen + updist, data = environment2, prior = "JZS")
plot(bas.model)
```

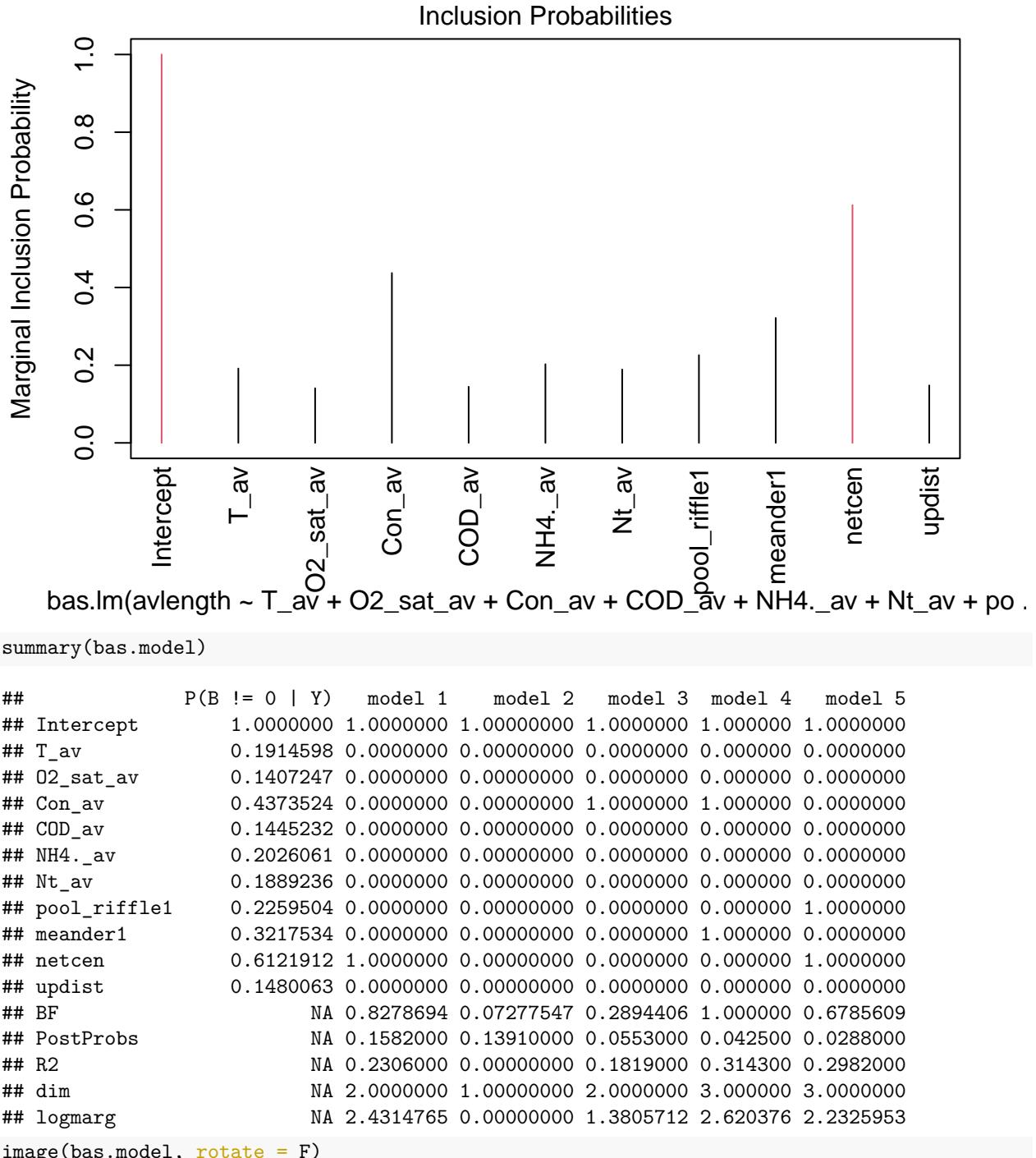


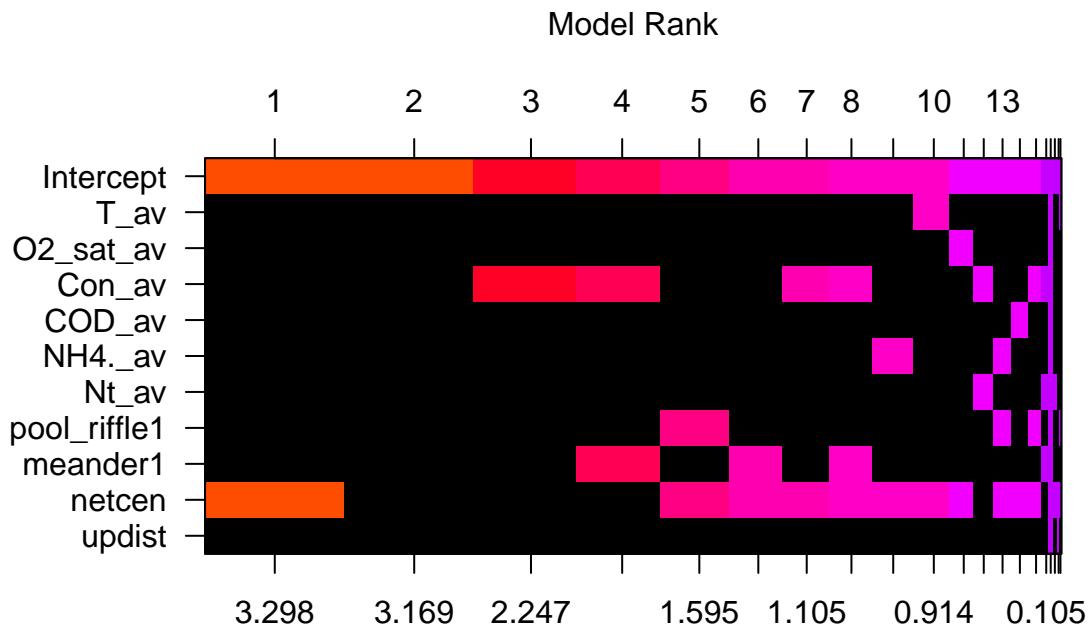
Predictions under BMA

bas.lm(avlength ~ T\_av + O2\_sat\_av + Con\_av + COD\_av + NH4.\_av + Nt\_av + po .









### Log Posterior Odds

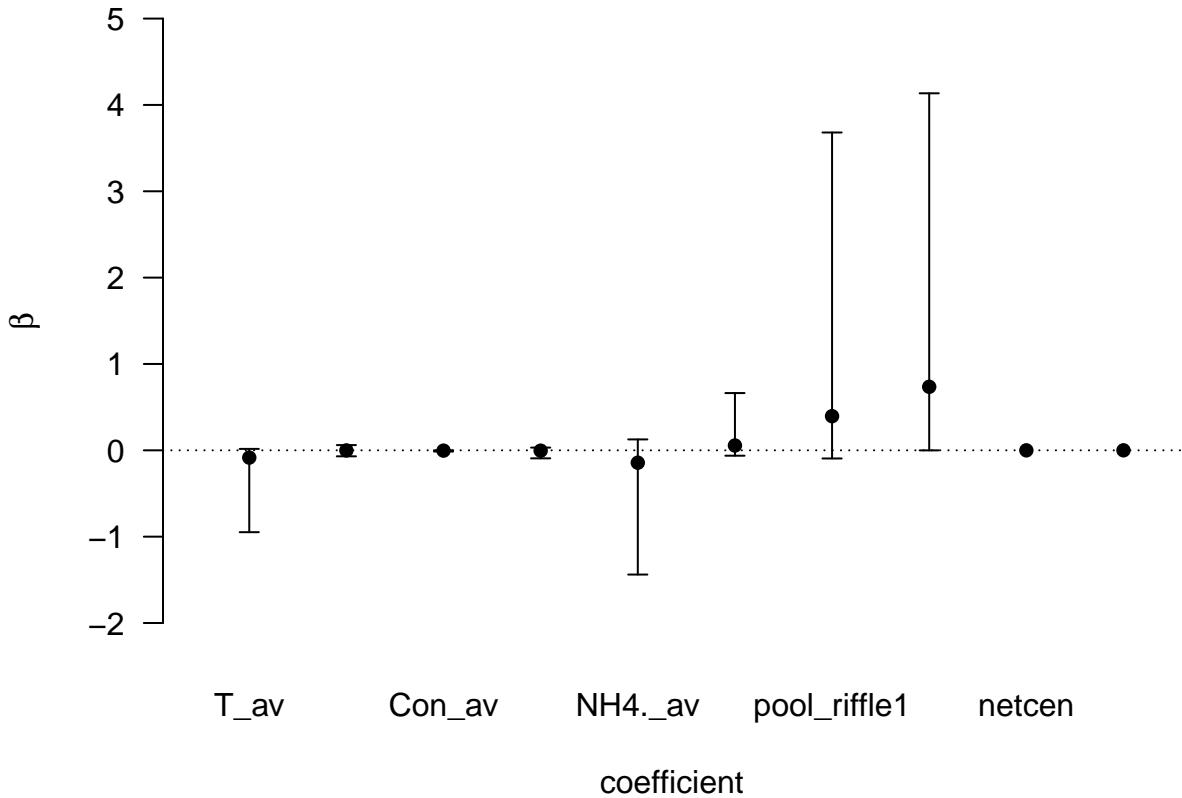
```

coef.model <- coef(bas.model)
# abs(coef.model$postmean)-2*coef.model$postsd > 0
plot(confint(coef.model, parm = 2:11))

## Warning in arrows(x[not.deg], ci[not.deg, 1], x[not.deg], ci[not.deg, 2], :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x[not.deg], ci[not.deg, 1], x[not.deg], ci[not.deg, 2], :
## zero-length arrow is of indeterminate angle and so skipped

```



```

## NULL

confint <- confint(coef.model, parm = 2:11)
write.table(confint, "length.txt", sep = "\t")
pip <- summary(bas.model)
PIP[c(3:12), 1] <- pip[2:11, 1] * sign(coef.model$postmean[2:11])
# coef.model$postmean[2:11]

# Prediction plot
newdata = as.data.frame(cbind(rep(mean(environment2$T_av), 37), rep(mean(environment2$O2_sat_av),
  37), rep(mean(environment2$Con_av), 37), rep(mean(environment2$COD_av), 37),
  rep(mean(environment2$NH4._av), 37), rep(mean(environment2$Nt_av), 37), rep(1,
  37), rep(1, 37), rep(mean(netcen), 37), rep(mean(updist), 37)))
colnames(newdata) <- c("T_av", "O2_sat_av", "Con_av", "COD_av", "NH4._av", "Nt_av",
  "pool_riffle", "meander", "netcen", "updist")
newdata[, "pool_riffle"] <- as.factor(newdata[, "pool_riffle"])
newdata[, "meander"] <- as.factor(newdata[, "meander"])
newdata1 <- newdata
newdata1[, "netcen"] <- netcen
BMA_avlength_netcen <- predict(bas.model, newdata = newdata1, estimator = "BMA",
  se.fit = TRUE)

figure3j <- ggplot(environment2, aes(netcen, BMA_avlength_netcen$fit)) + theme_bw() +
  geom_line(color = "red", size = 1) + geom_ribbon(aes(ymin = (BMA_avlength_netcen$fit -
  BMA_avlength_netcen$se.bma.fit), ymax = (BMA_avlength_netcen$fit + BMA_avlength_netcen$se.bma.fit)),
  alpha = 0.1) + geom_point(data = environment2, aes(x = netcen, y = avlength)) +
  labs(x = expression("Network peripherality [m]"), y = expression("Average host length [mm]"))

```

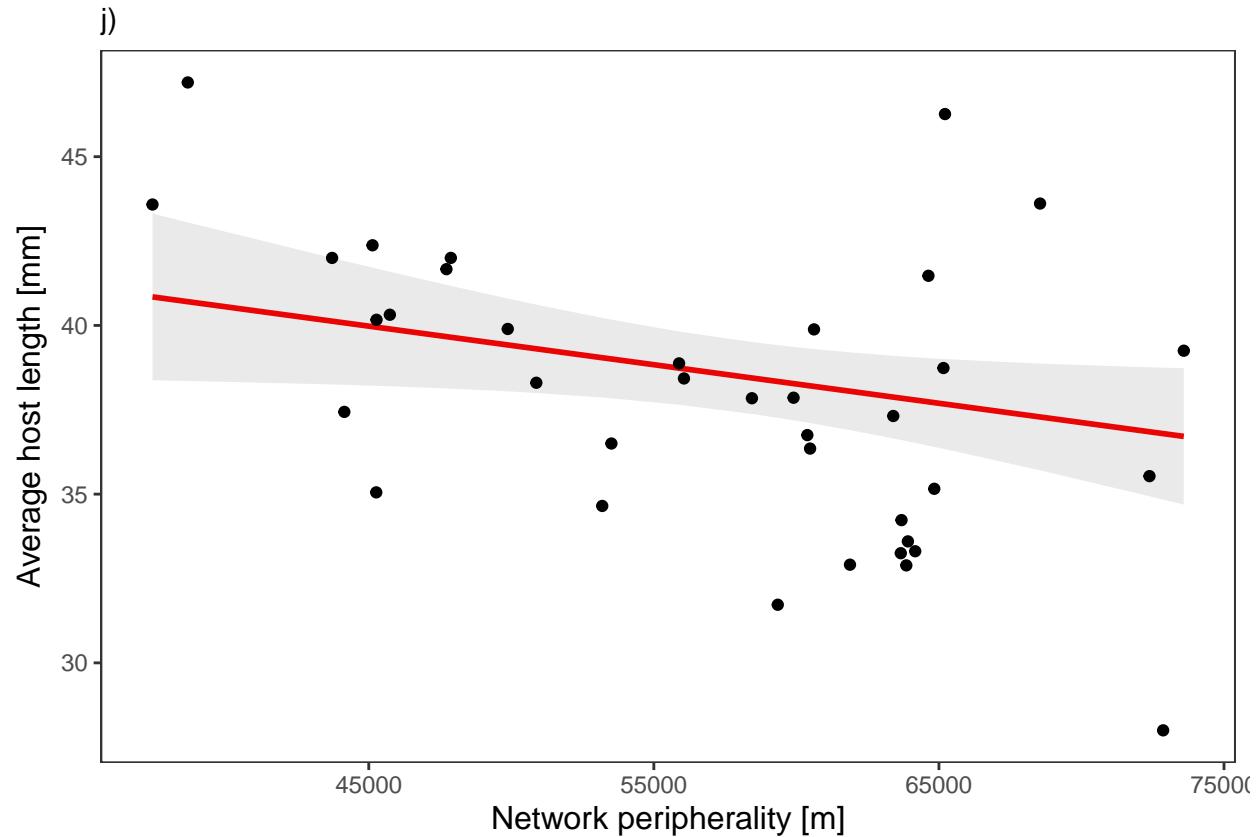
```

theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "j)")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

figure3j

```



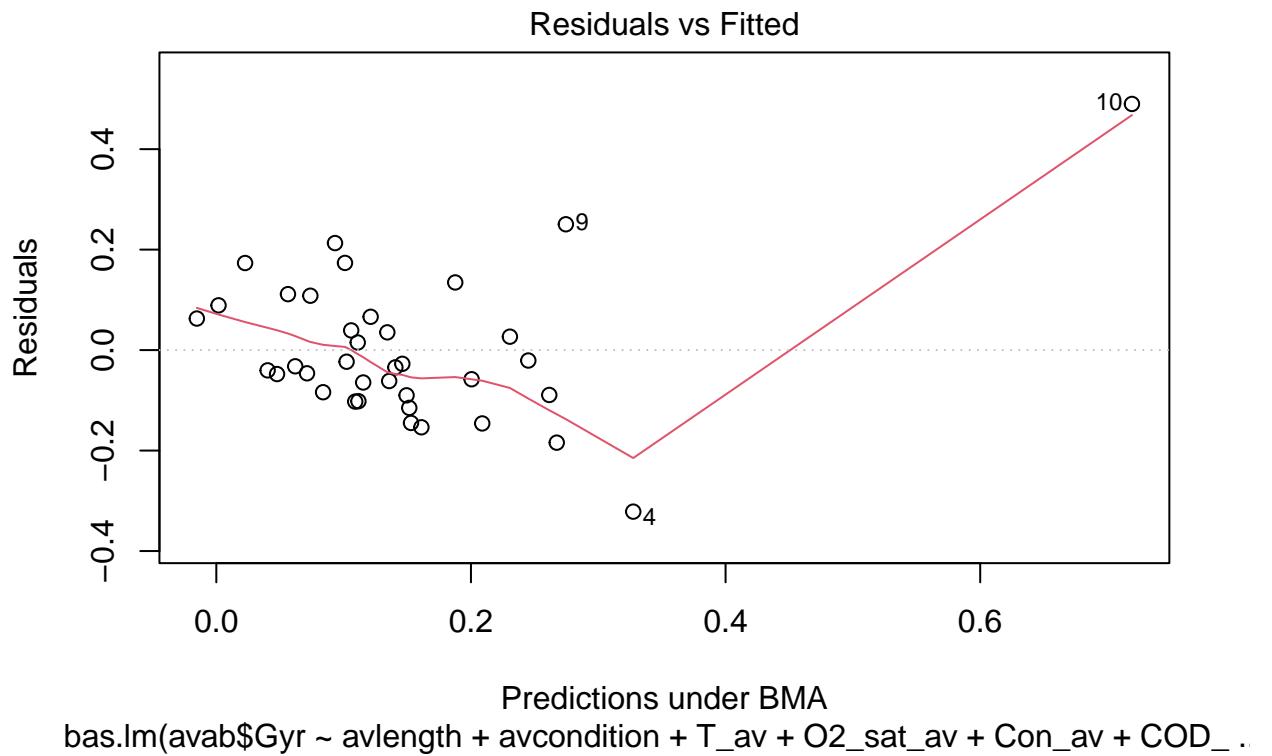
## 6.3 Variation in Gyrodactylus infection

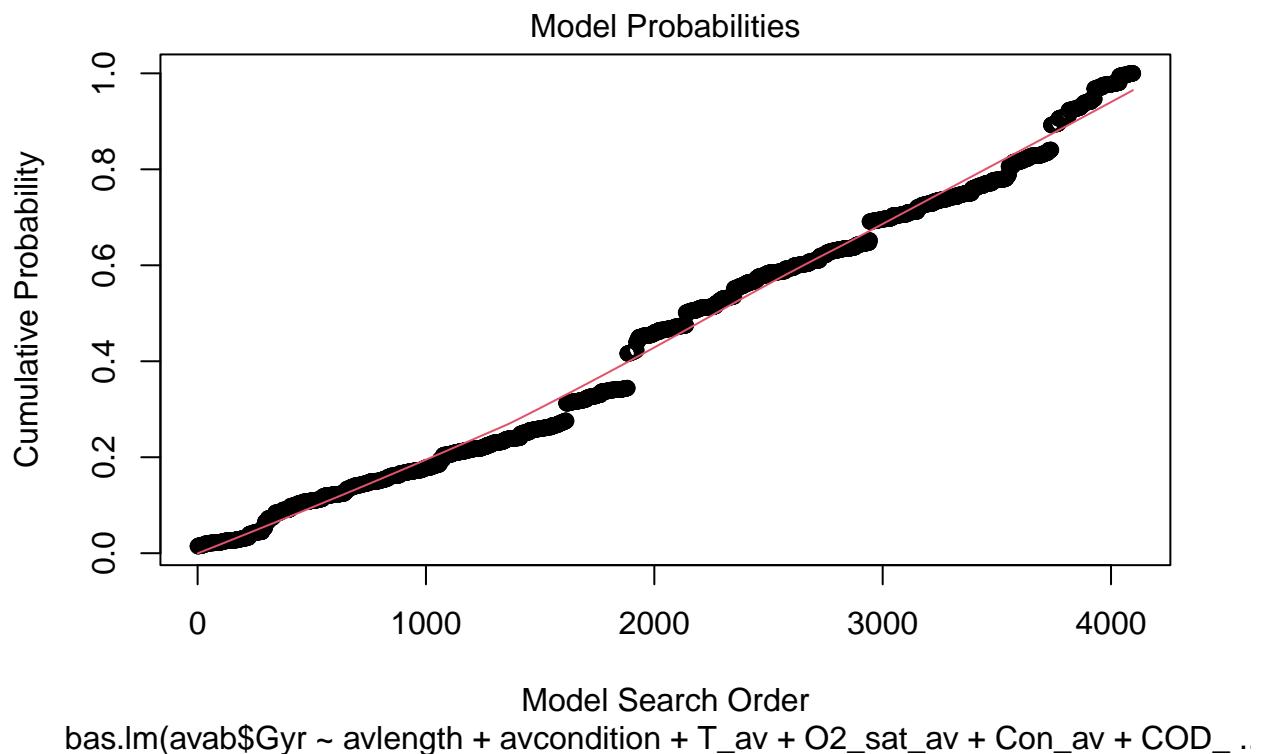
### 6.3.1 Mean abundance

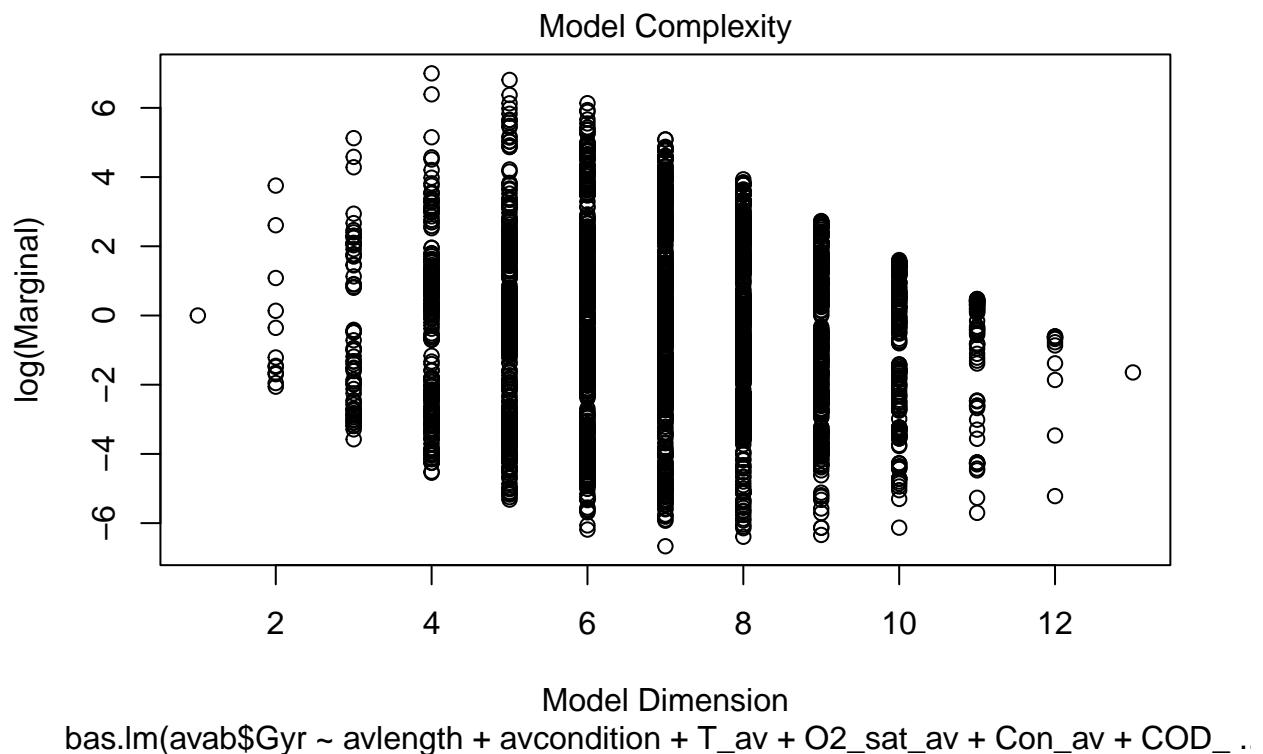
```

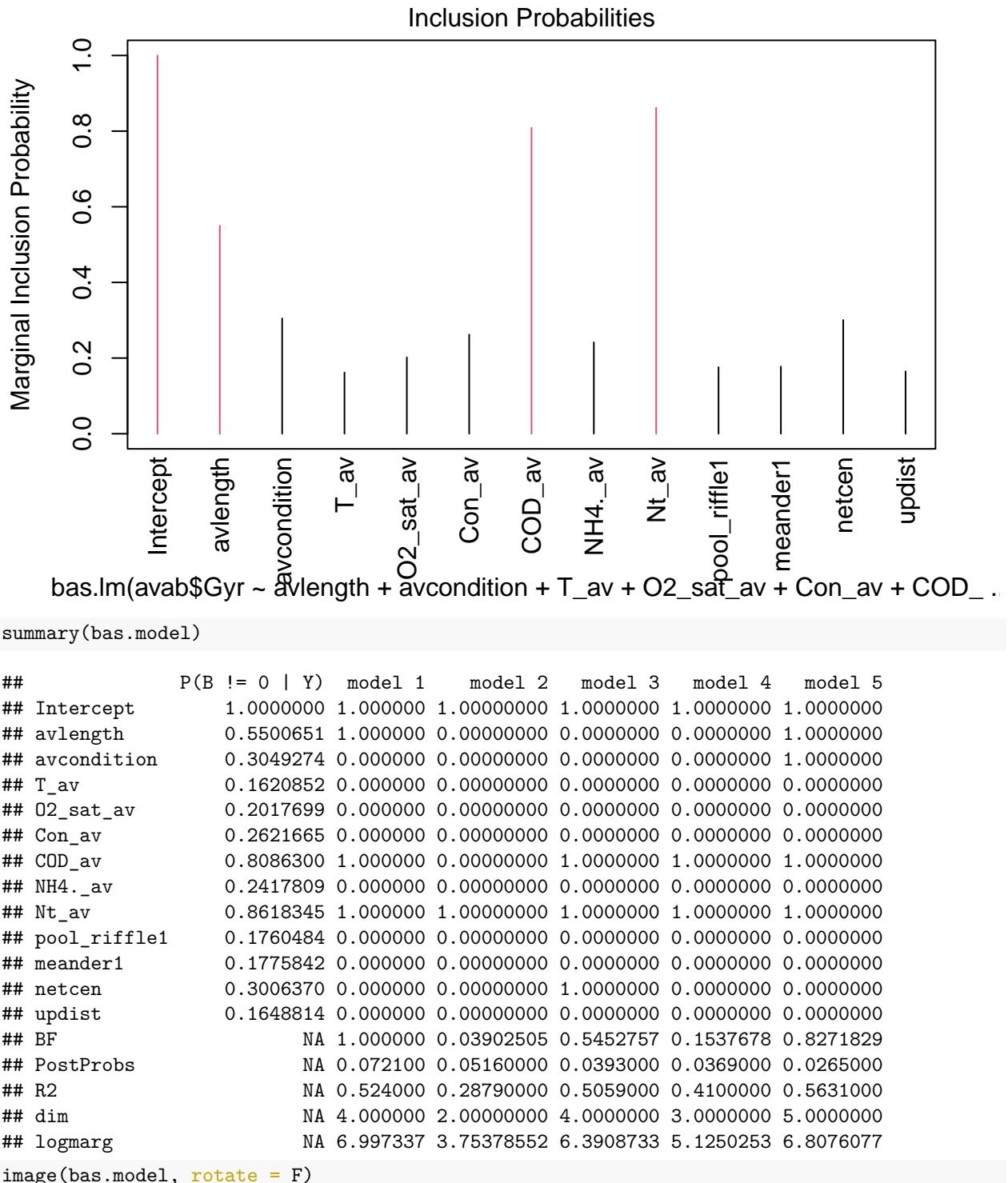
bas.model <- bas.lm(avab$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)

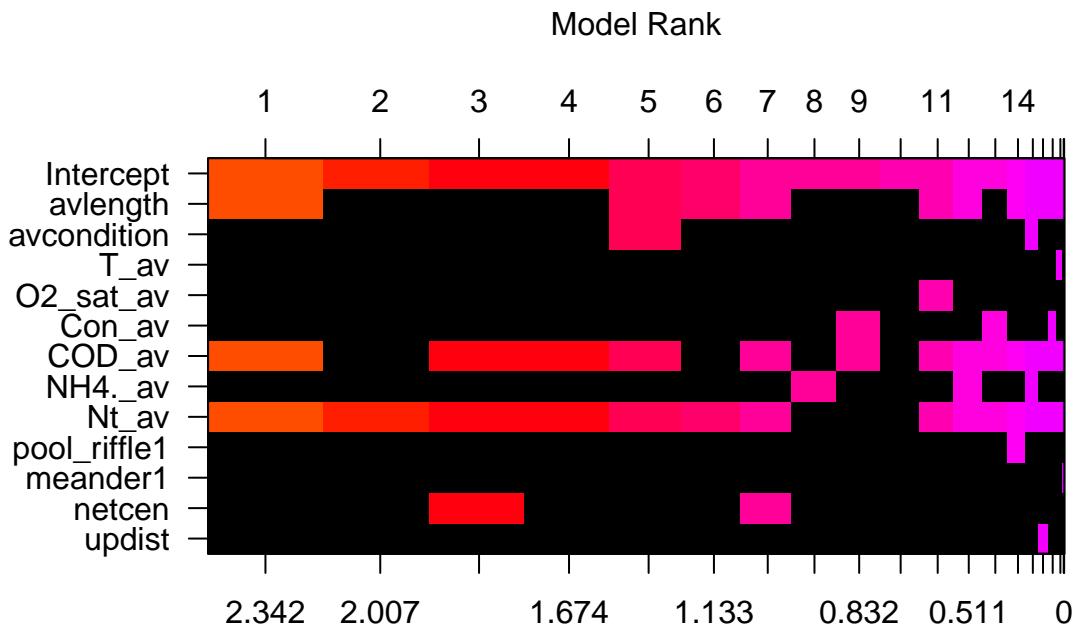
```











### Log Posterior Odds

```

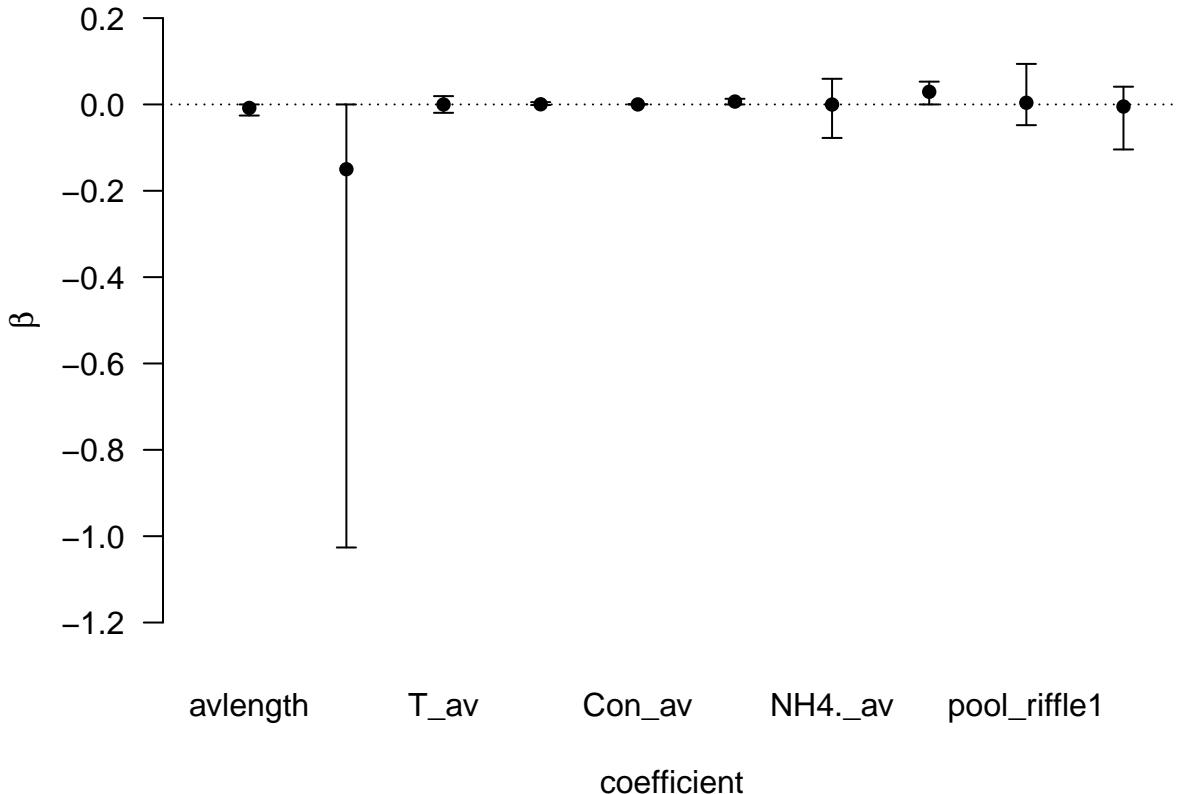
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE
confint(coef.model)

##              2.5%      97.5%       beta
## Intercept  9.368127e-02 2.026373e-01 1.489134e-01
## avlength   -2.506959e-02 0.000000e+00 -8.122675e-03
## avcondition -1.004059e+00 1.814248e-02 -1.500326e-01
## T_av        -2.150885e-02 1.781368e-02 -1.493132e-04
## O2_sat_av   -1.215569e-03 5.315800e-03 3.794809e-04
## Con_av      -6.200201e-05 5.698393e-04 5.726233e-05
## COD_av      0.000000e+00 1.304648e-02 6.652983e-03
## NH4_av      -6.317018e-02 7.265640e-02 -1.888173e-04
## Nt_av        0.000000e+00 5.261888e-02 2.930918e-02
## pool_riffle1 -5.710073e-02 9.309714e-02 3.966550e-03
## meander1    -9.632646e-02 4.166008e-02 -4.806259e-03
## netcen       -4.040786e-07 1.020782e-05 1.332021e-06
## updist      -1.164647e-06 2.073127e-06 4.618571e-08
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

```
plot(confint(coef.model, parm = 2:11))
```



```
## NULL

confint <- confint(coef.model, parm = 2:11)
write.table(confint, "GyroAA.txt", sep = "\t")
pip <- summary(bas.model)
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

### 6.3.1.1 Prediction plot for marginal effect of host condition on average Gyrodactylus infection

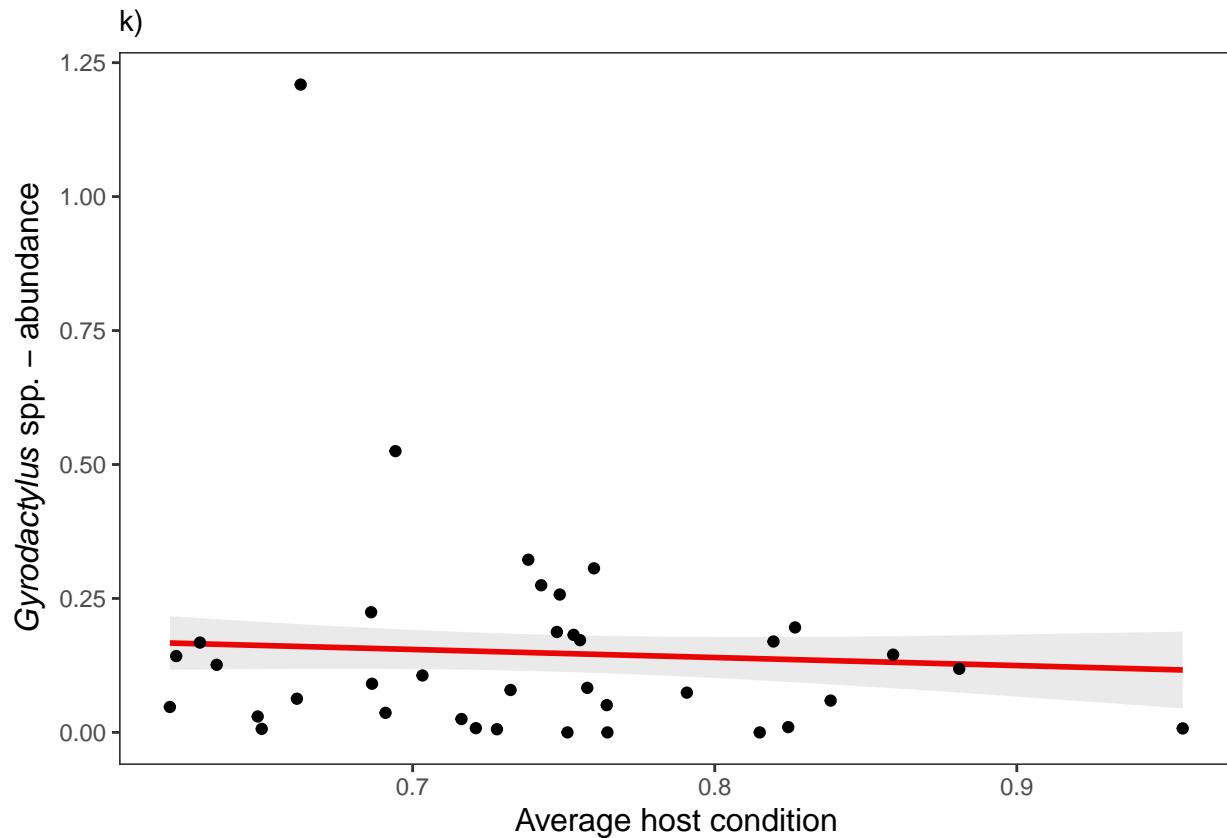
```
# Prediction plot
newdata = as.data.frame(cbind(rep(mean(avlength), 37), rep(mean(avcondition), 37),
  rep(mean(environment2$T_av), 37), rep(mean(environment2$O2_sat_av), 37), rep(mean(environment2$Con_av),
  37), rep(mean(environment2$COD_av), 37), rep(mean(environment2$NH4._av),
  37), rep(mean(environment2$Nt_av), 37), rep(1, 37), rep(1, 37), rep(mean(netcen),
  37), rep(mean(updist), 37)))
colnames(newdata) <- c("avlength", "avcondition", "T_av", "O2_sat_av", "Con_av",
  "COD_av", "NH4._av", "Nt_av", "pool_riffle", "meander", "netcen", "updist")
newdata[, "pool_riffle"] <- as.factor(newdata[, "pool_riffle"])
newdata[, "meander"] <- as.factor(newdata[, "meander"])
newdata1 <- newdata
newdata1[, "avcondition"] <- avcondition
BMA_Gyr_avcond <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
```

```

figure3k = ggplot(environment2, aes(avcondition, BMA_Gyr_avcond$fit)) + theme_bw() +
  geom_line(color = "red", size = 1) + geom_ribbon(aes(ymin = (BMA_Gyr_avcond$fit -
  BMA_Gyr_avcond$se.bma.fit), ymax = (BMA_Gyr_avcond$fit + BMA_Gyr_avcond$se.bma.fit)),
  alpha = 0.1) + geom_point(data = environment2, aes(x = avcondition, y = avab$Gyr)) +
  labs(x = expression("Average host condition"), y = expression(paste(italic("Gyrodactylus"),
  " spp. - abundance"))) + theme(axis.title.x = element_text(size = 12), axis.title.y = element_t
  them(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "k)")

figure3k

```



### 6.3.1.2 Prediction plot for marginal effect of COD on average Gyrodactylus infection

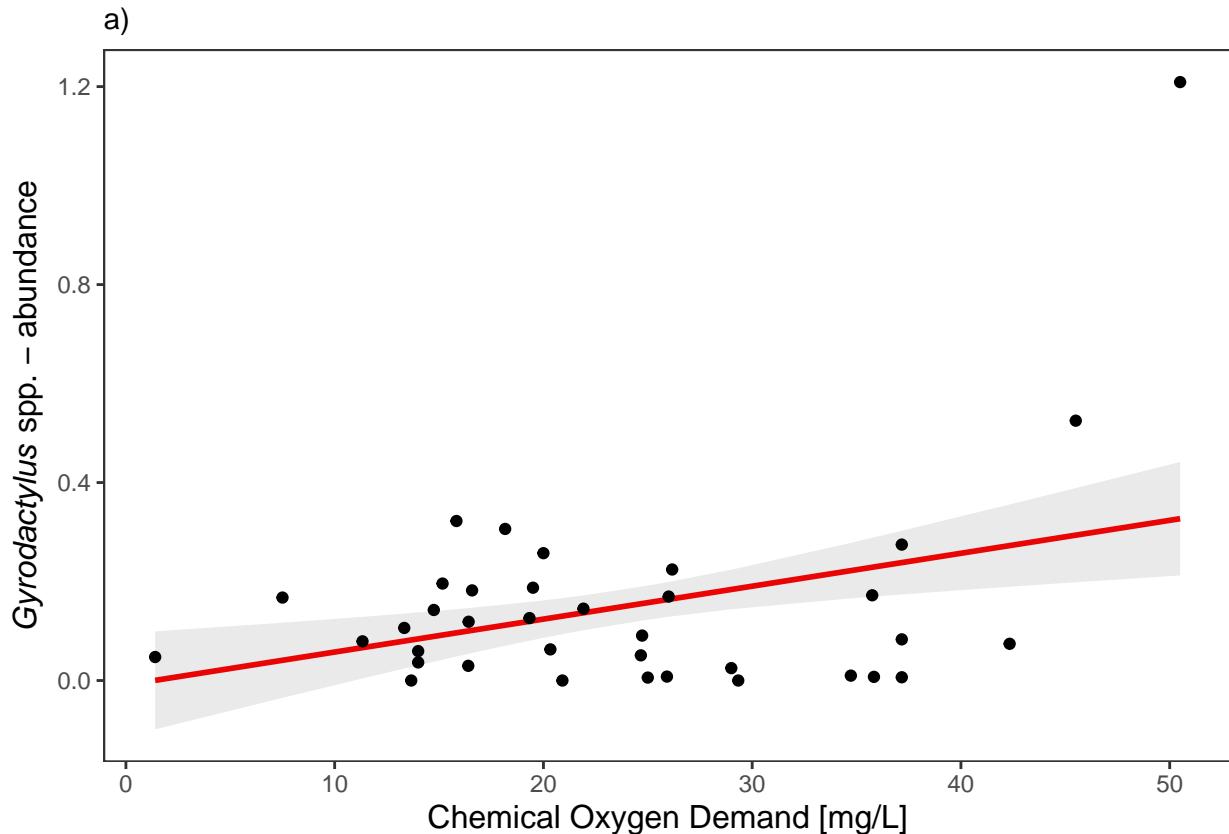
```

newdata1 <- newdata
newdata1[, "COD_av"] <- environment2$COD_av
BMA_Gyr_COD_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3a = ggplot(environment2, aes(COD_av, BMA_Gyr_COD_av$fit)) + theme_bw() + geom_line(color = "red",
  size = 1) + geom_ribbon(aes(ymin = (BMA_Gyr_COD_av$fit - BMA_Gyr_COD_av$se.bma.fit),
  ymax = (BMA_Gyr_COD_av$fit + BMA_Gyr_COD_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environm
  aes(x = COD_av, y = avab$Gyr)) + labs(x = expression("Chemical Oxygen Demand [mg/L]"),
  y = expression(paste(italic("Gyrodactylus"), " spp. - abundance"))) + theme(axis.title.x = element_
  axis.title.y = element_text(size = 12)) + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) + labs(subtitle = "a)")

figure3a

```

figure3a

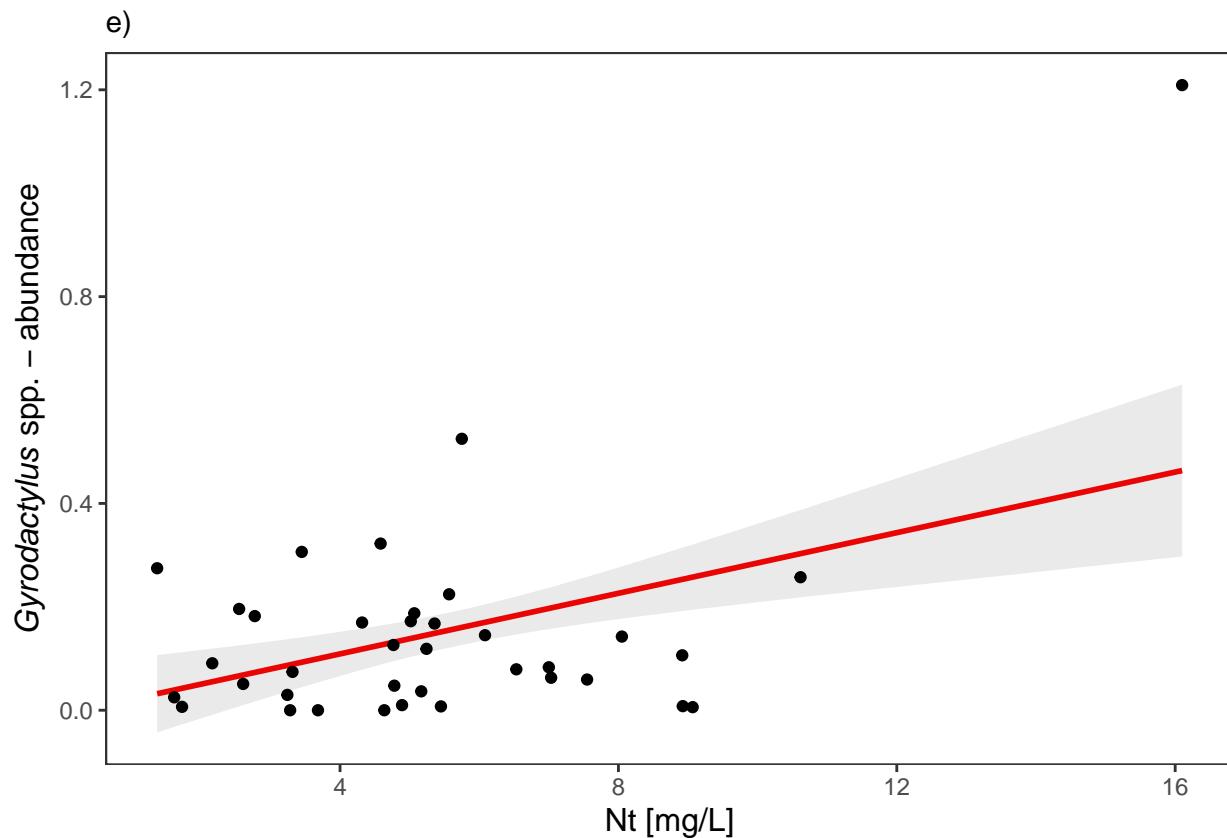


### 6.3.2.3 Prediction plot for marginal effect of total nitrogen on average Gyrodactylus infection

```

newdata1 <- newdata
newdata1[, "Nt_av"] <- environment2$Nt_av
BMA_Gyr_Nt_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3e = ggplot(environment2, aes(Nt_av, BMA_Gyr_Nt_av$fit)) + theme_bw() + geom_line(color = "red",
  size = 1) + geom_ribbon(aes(ymin = (BMA_Gyr_Nt_av$fit - BMA_Gyr_Nt_av$se.bma.fit),
  ymax = (BMA_Gyr_Nt_av$fit + BMA_Gyr_Nt_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environment2,
  aes(x = Nt_av, y = avab$Gyr)) + labs(x = expression("Nt [mg/L]"), y = expression(paste(italic("Gyro-
  " spp. - abundance")))) + theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(
  size = 12), panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "e")
figure3e

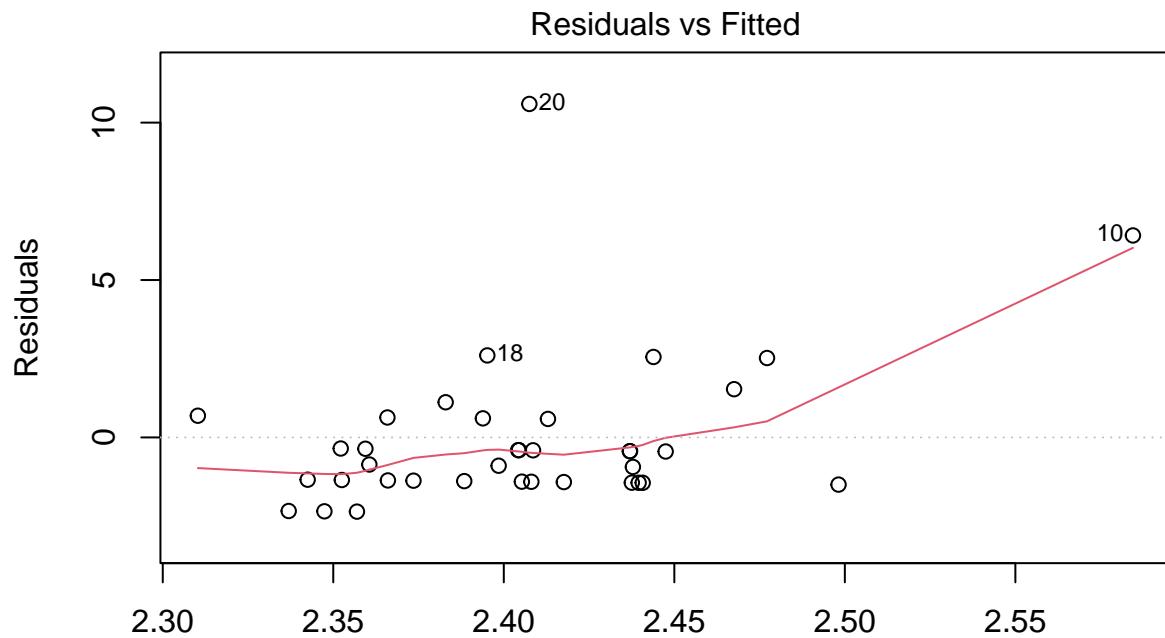
```



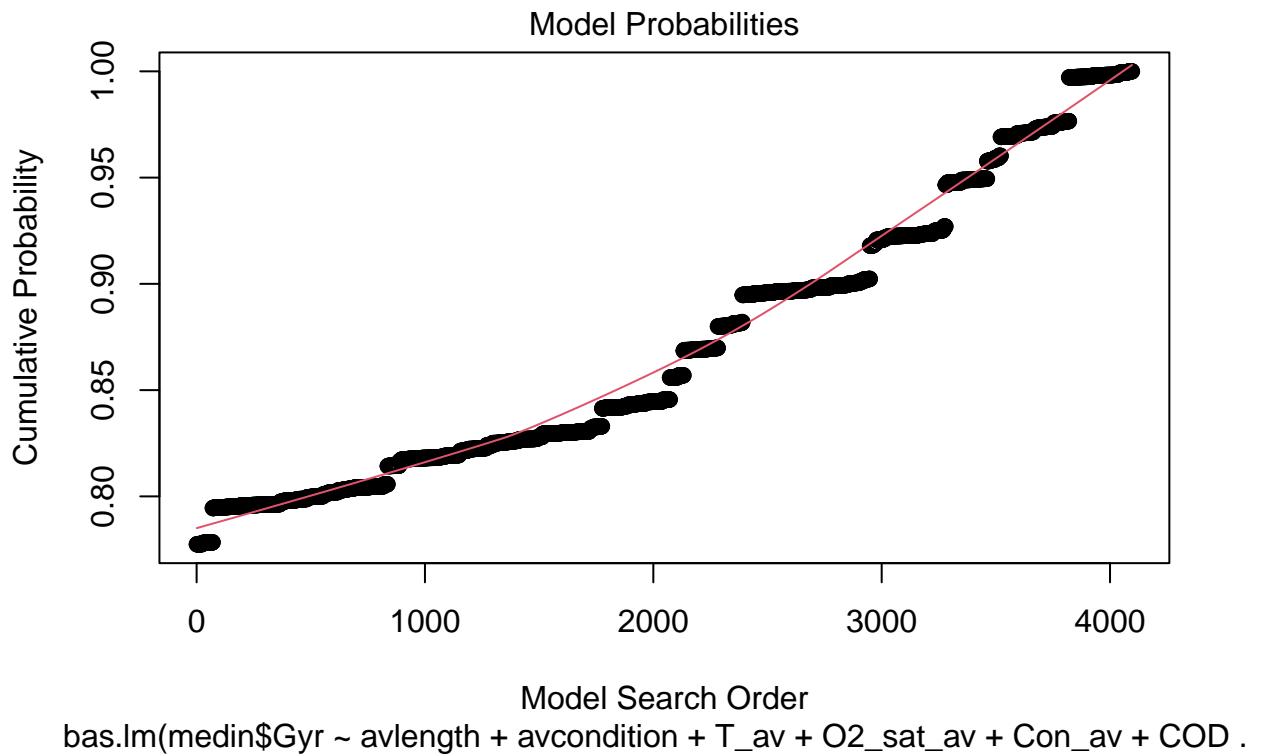
### 6.3.2 Median infection intensity

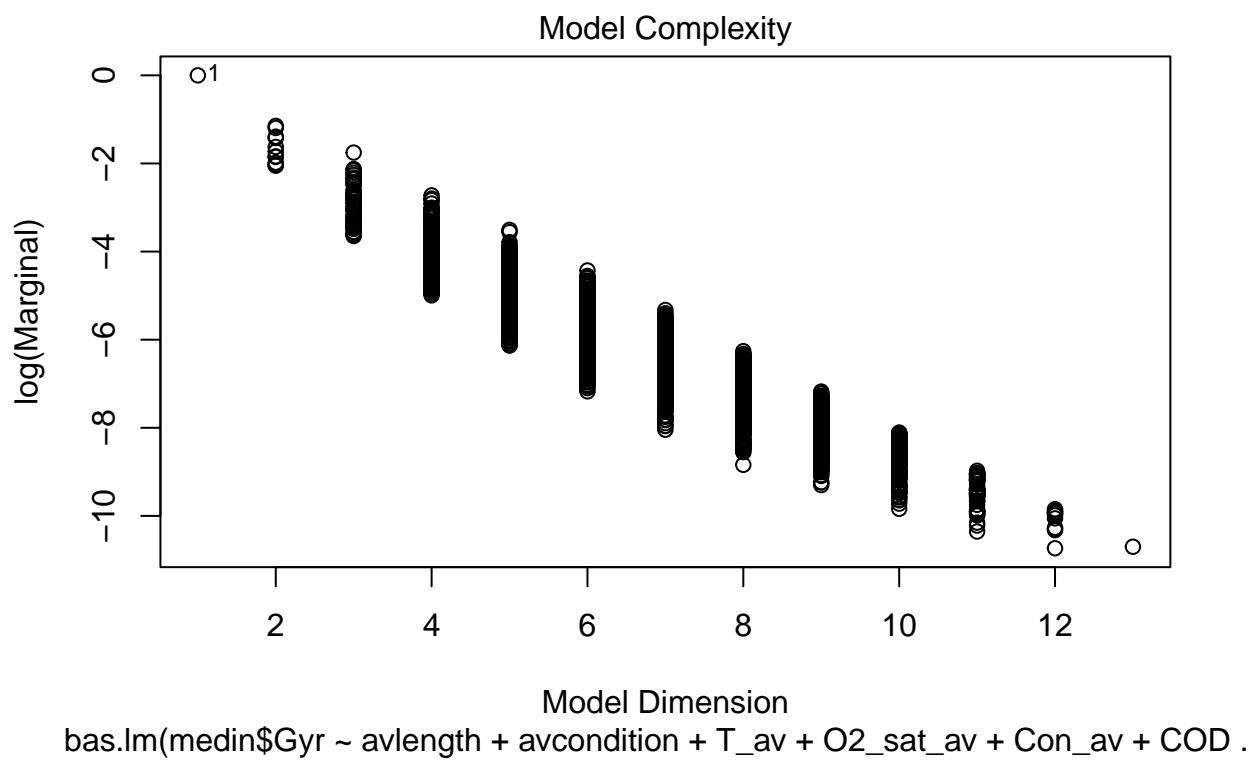
```
bas.model <- bas.lm(medin$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals

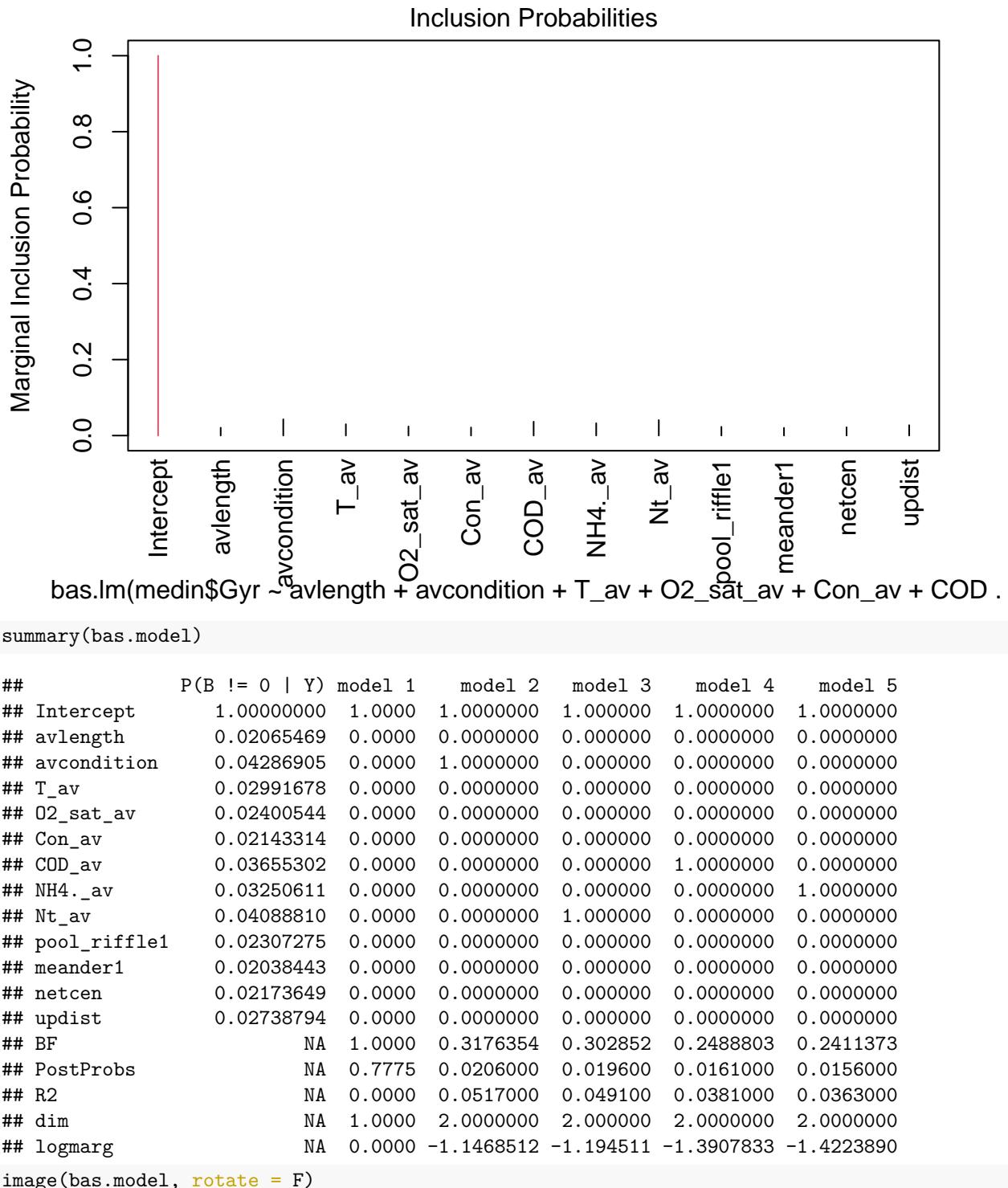
plot(bas.model)
```

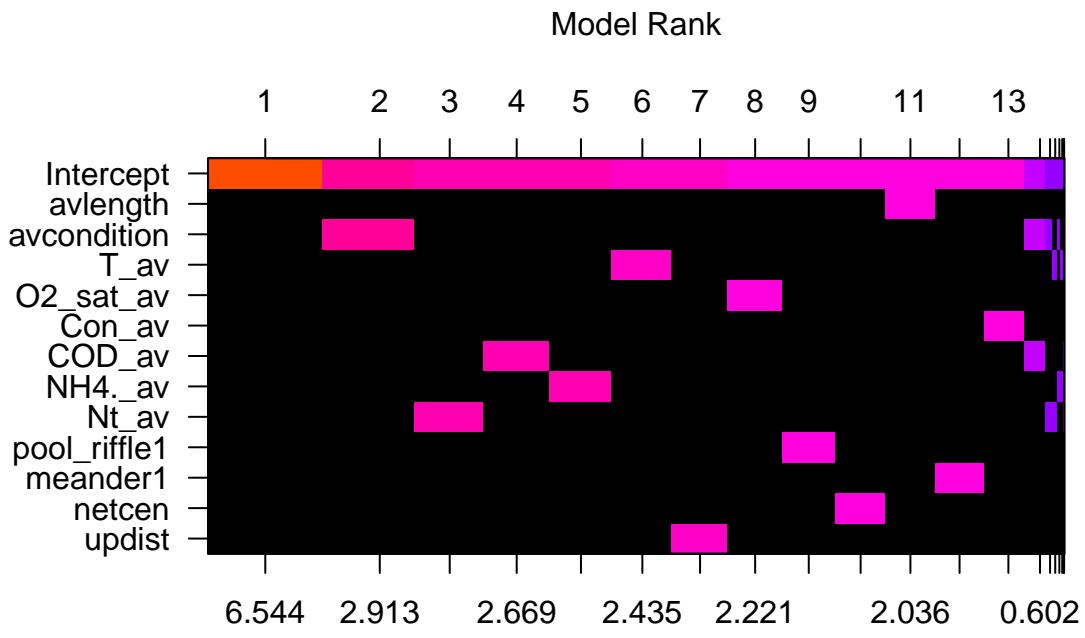


Predictions under BMA  
bas.lm(medin\$Gyr ~ avlength + avcondition + T\_av + O2\_sat\_av + Con\_av + COD .









### Log Posterior Odds

```

coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

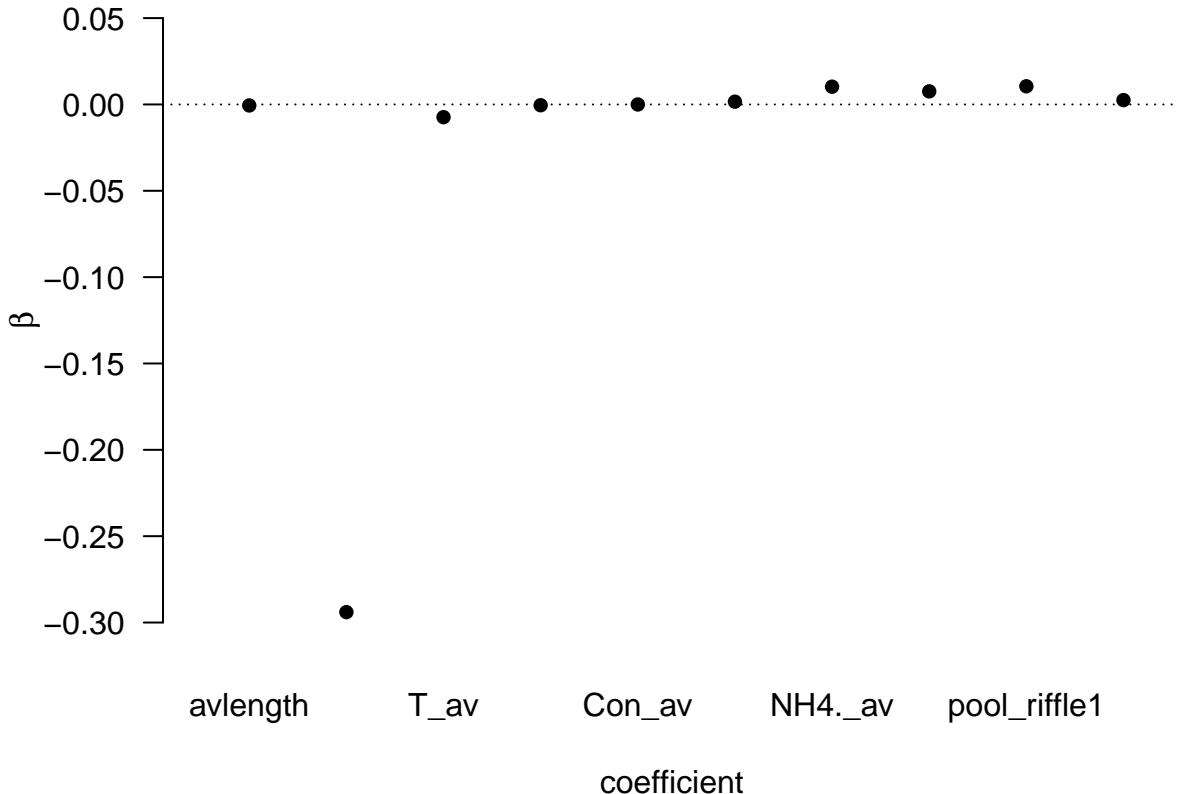
## [1] TRUE FALSE FALSE
## [13] FALSE

confint(coef.model)

##              2.5%    97.5%      beta
## Intercept  1.554818 3.237484 2.405405e+00
## avlength   0.000000 0.000000 -5.938396e-04
## avcondition 0.000000 0.000000 -2.939823e-01
## T_av       0.000000 0.000000 -7.368692e-03
## O2_sat_av  0.000000 0.000000 -4.581964e-04
## Con_av     0.000000 0.000000 -1.594570e-05
## COD_av     0.000000 0.000000 1.609035e-03
## NH4_av     0.000000 0.000000 1.027436e-02
## Nt_av      0.000000 0.000000 7.570653e-03
## pool_riffle1 0.000000 0.000000 1.051497e-02
## meander1   0.000000 0.000000 2.511911e-03
## netcen     0.000000 0.000000 3.990331e-07
## updist     0.000000 0.000000 4.560731e-07
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

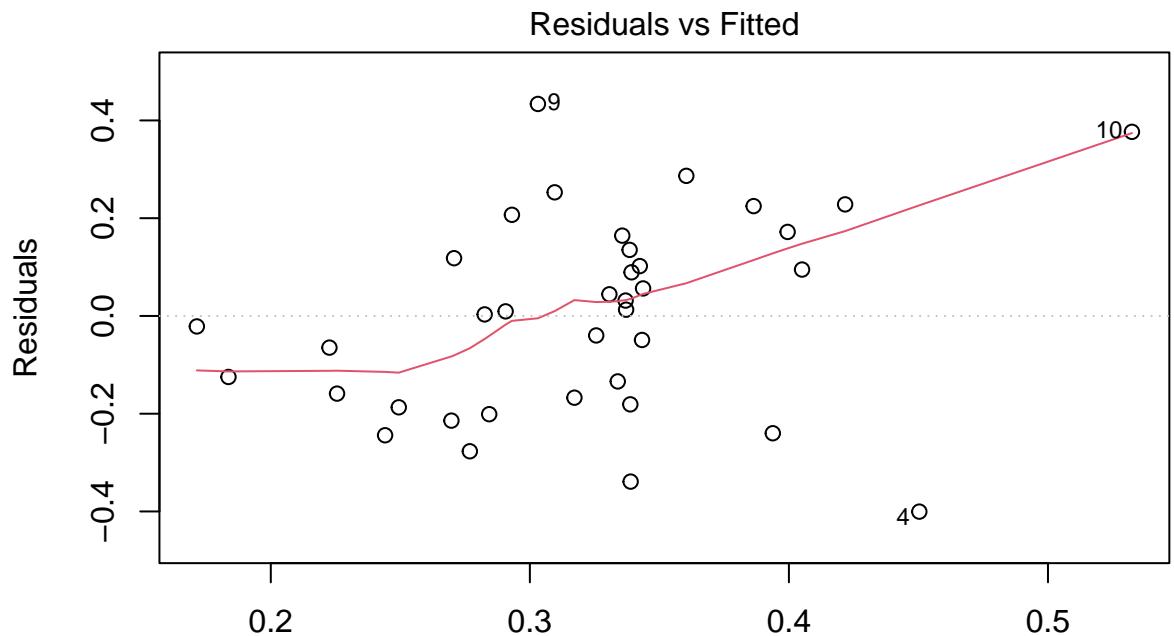
```
plot(confint(coef.model, parm = 2:11))
```



```
## NULL  
confint <- confint(coef.model, parm = 2:11)  
write.table(confint, "GyroAA.txt", sep = "\t")  
pip <- summary(bas.model)  
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

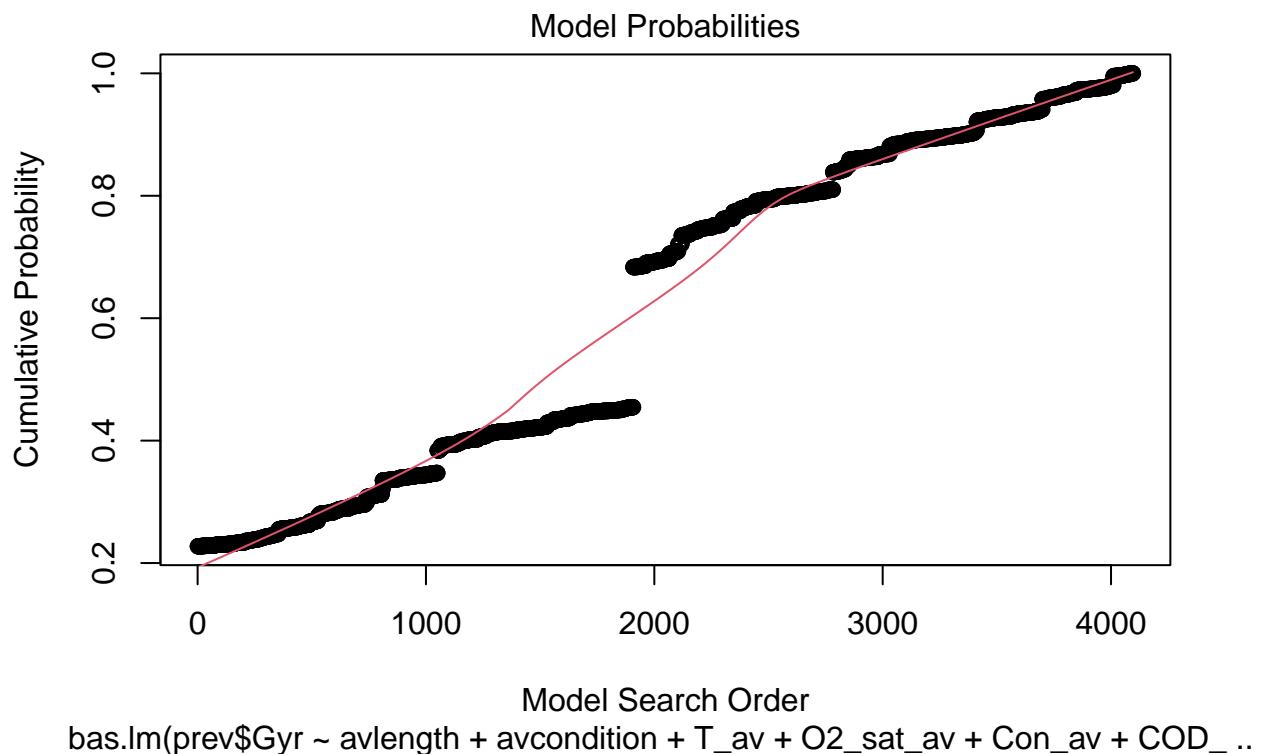
### 6.3.2 Prevalence

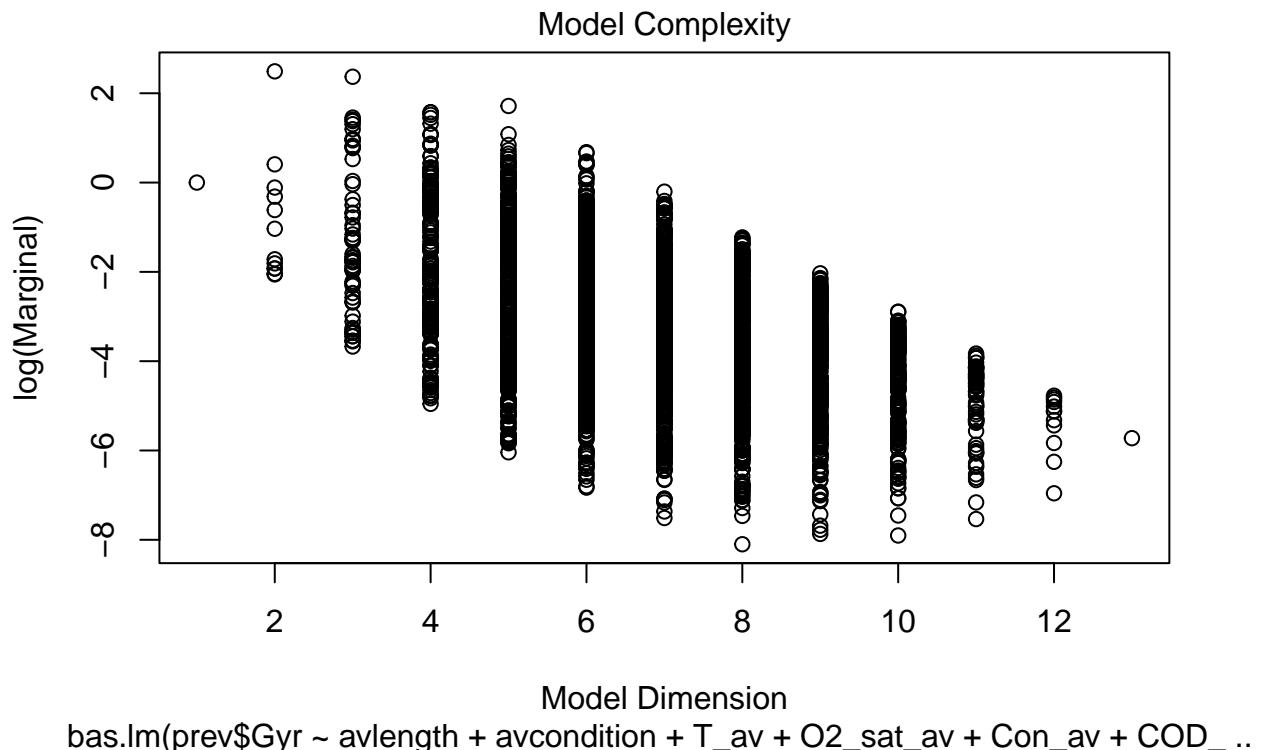
```
bas.model <- bas.lm(prev$Gyr ~ avlength + avcondition + T_av + O2_sat_av + Con_av +  
COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,  
prior = "JZS")  
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA  
r = bas.model$Y - yhat #these are the model residuals  
  
plot(bas.model)
```

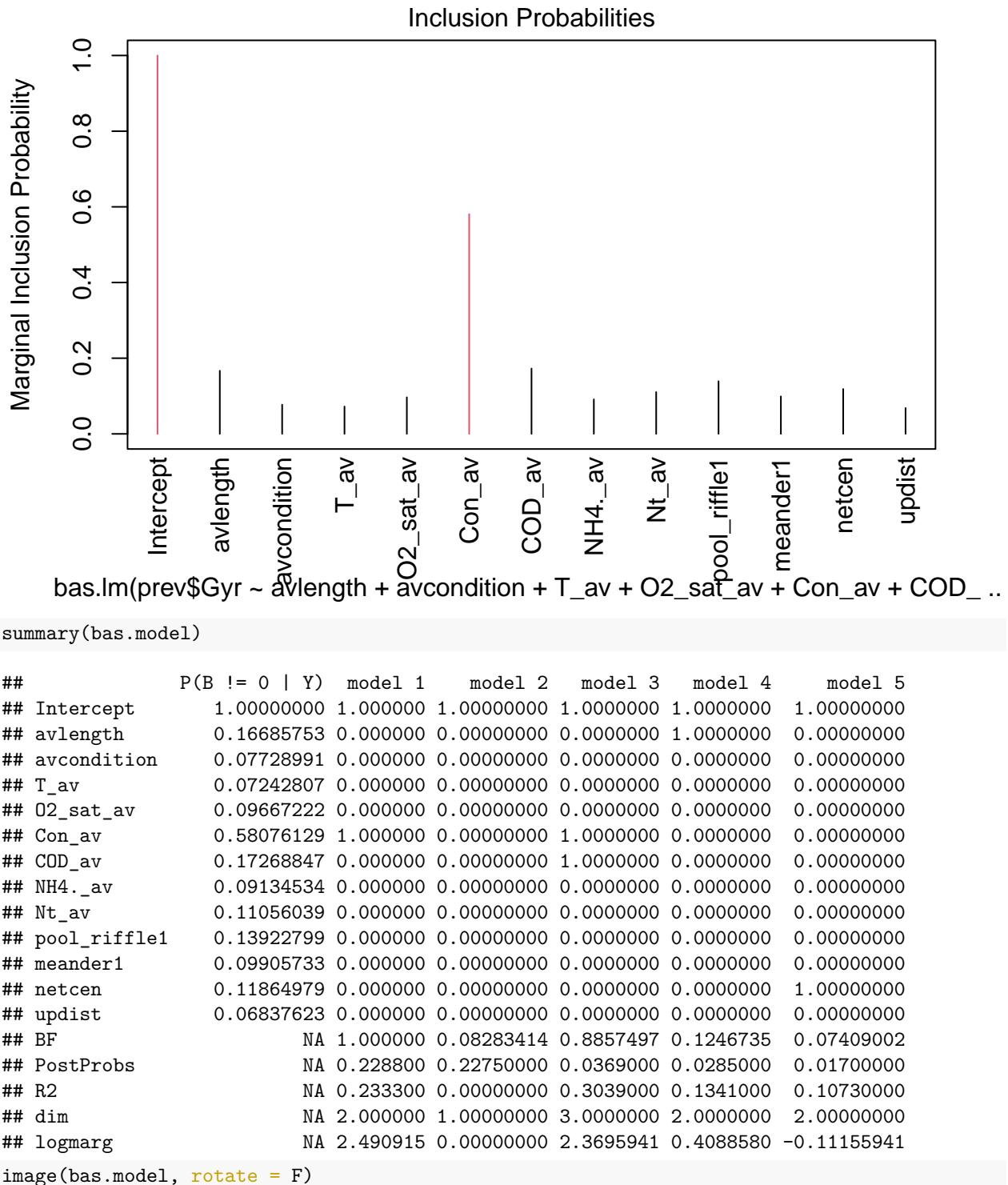


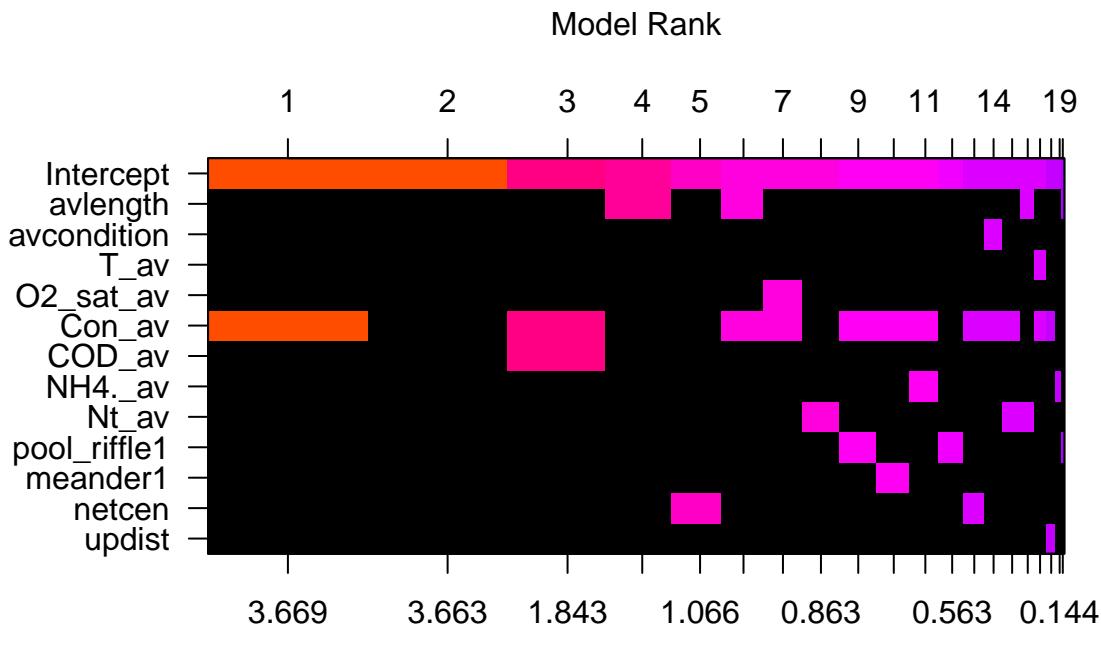
Predictions under BMA

bas.lm(prev\$Gyr ~ avlength + avcondition + T\_av + O2\_sat\_av + Con\_av + COD\_ ..









```

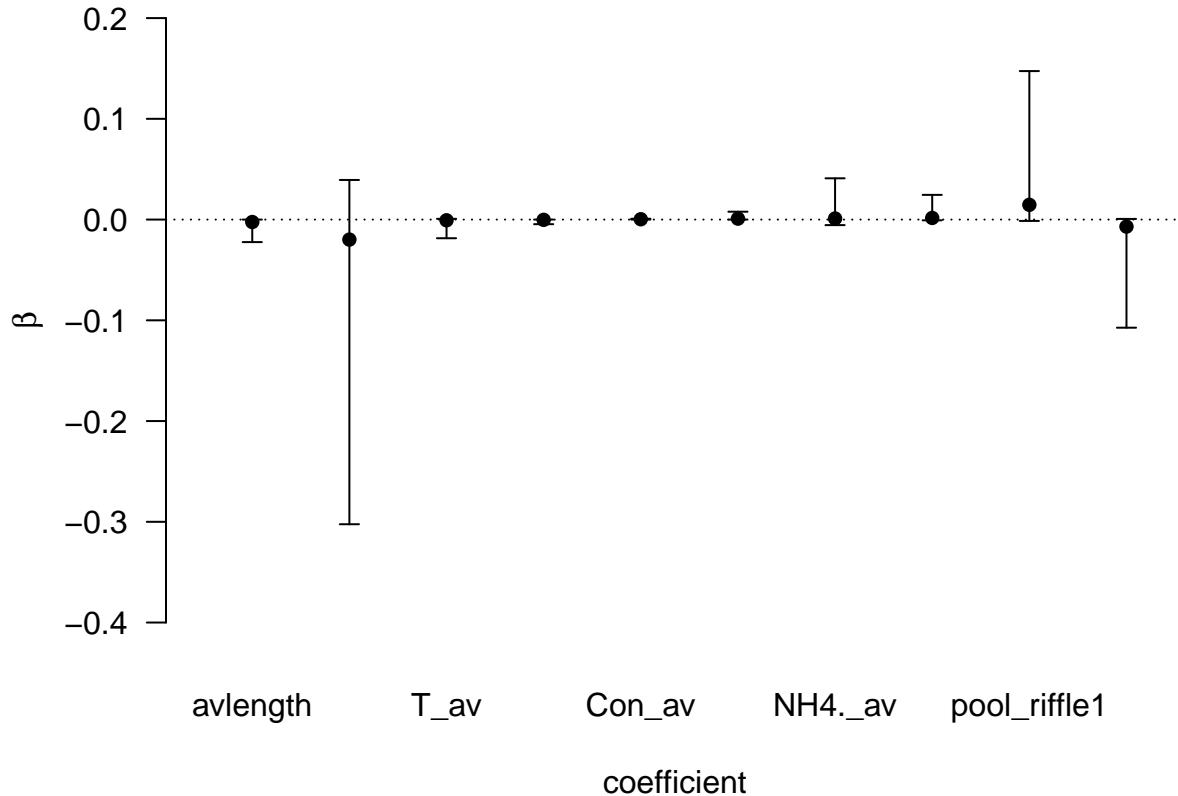
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept    2.504121e-01 3.942108e-01 3.224082e-01
## avlength     -2.251760e-02 5.740352e-05 -2.431148e-03
## avcondition  -4.323417e-01 7.238452e-04 -1.986363e-02
## T_av         -1.222674e-02 6.635477e-03 -7.123130e-04
## O2_sat_av   -4.479545e-03 0.000000e+00 -2.662986e-04
## Con_av        0.000000e+00 8.007537e-04 3.112530e-04
## COD_av       -9.015984e-05 7.979574e-03 9.376253e-04
## NH4_av        -2.806788e-04 4.792229e-02 1.064409e-03
## Nt_av         -2.623335e-06 2.542809e-02 1.612509e-03
## pool_riffle1 0.000000e+00 1.507804e-01 1.465705e-02
## meander1     -1.080379e-01 9.591179e-04 -6.966751e-03
## netcen        0.000000e+00 8.149340e-06 5.765326e-07
## updist       -1.036460e-06 1.759374e-07 -6.922933e-09
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

```
plot(confint(coef.model, parm = 2:11))
```

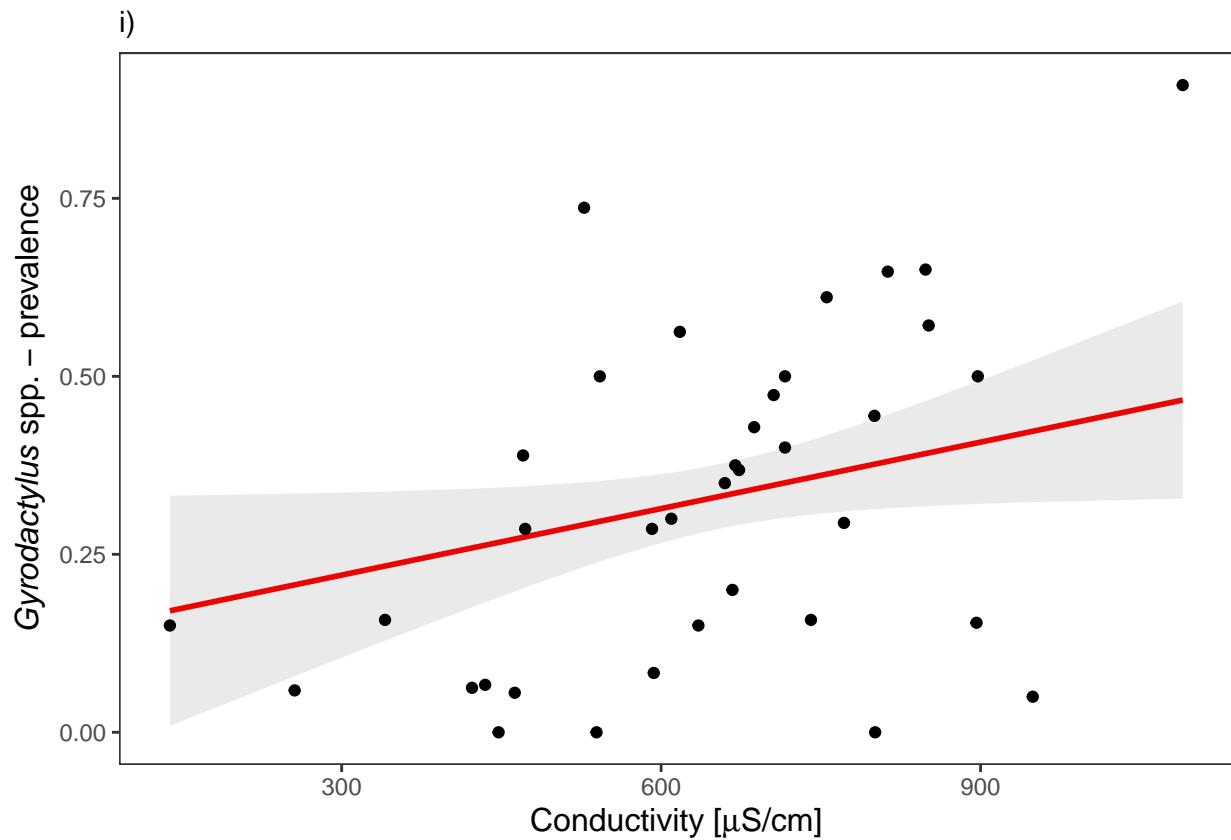


```
## NULL
confint <- confint(coef.model, parm = 2:11)
write.table(confint, "GyroAA.txt", sep = "\t")
pip <- summary(bas.model)
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

### 6.3.2.3 Prediction plot for marginal effect of conductivity on average Gyrodactylus infection

```
newdata1 <- newdata
newdata1[, "Con_av"] <- environment2$Con_av
BMA_Gyr_Con_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3i = ggplot(environment2, aes(Con_av, BMA_Gyr_Con_av$fit)) + theme_bw() + geom_line(color = "red",
size = 1) + geom_ribbon(aes(ymin = (BMA_Gyr_Con_av$fit - BMA_Gyr_Con_av$se.bma.fit),
ymax = (BMA_Gyr_Con_av$fit + BMA_Gyr_Con_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environment2,
aes(x = Con_av, y = prev$Gyr)) + labs(x = expression(paste("Conductivity [",
mu, "S/cm]")), y = expression(paste(italic("Gyrodactylus"), " spp. - prevalence")))) +
theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
labs(subtitle = "i")
```

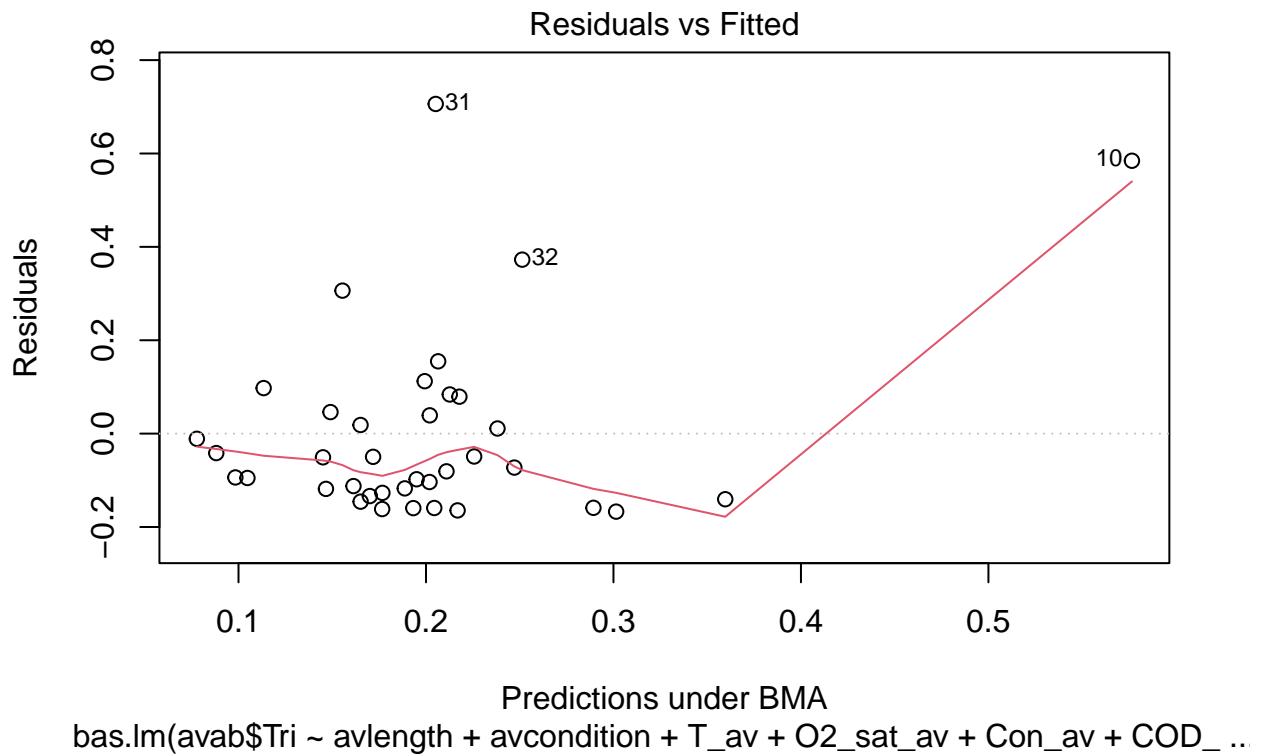
figure3i

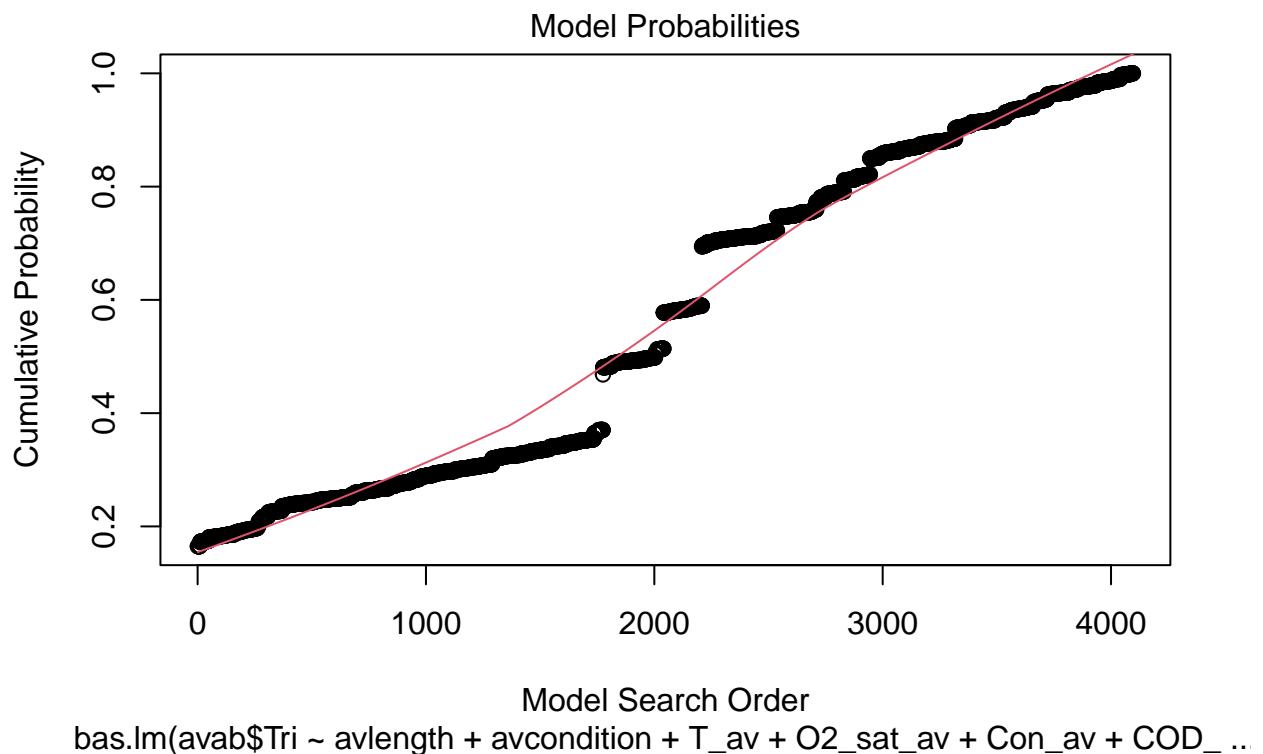


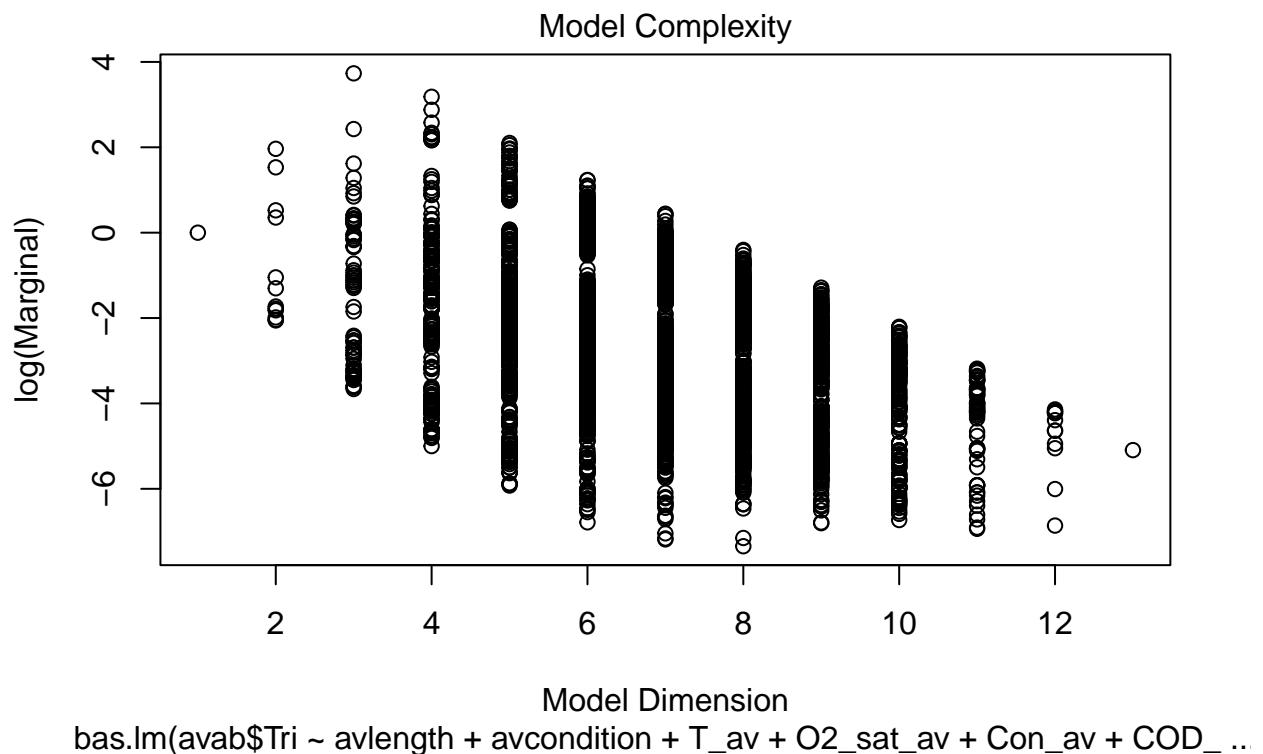
## 6.4 Variation in Trichodina infection

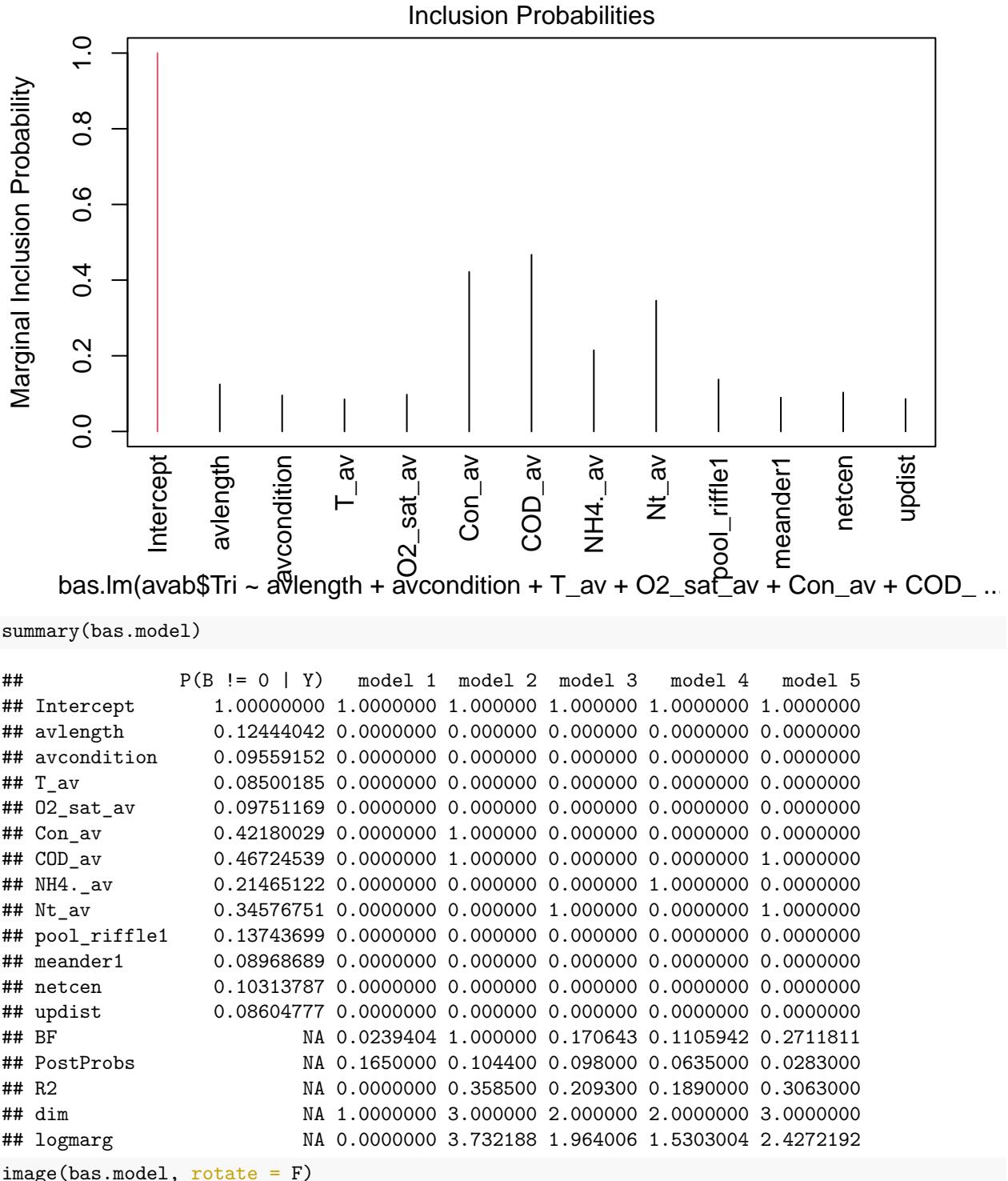
### 6.4.1 Mean abundance

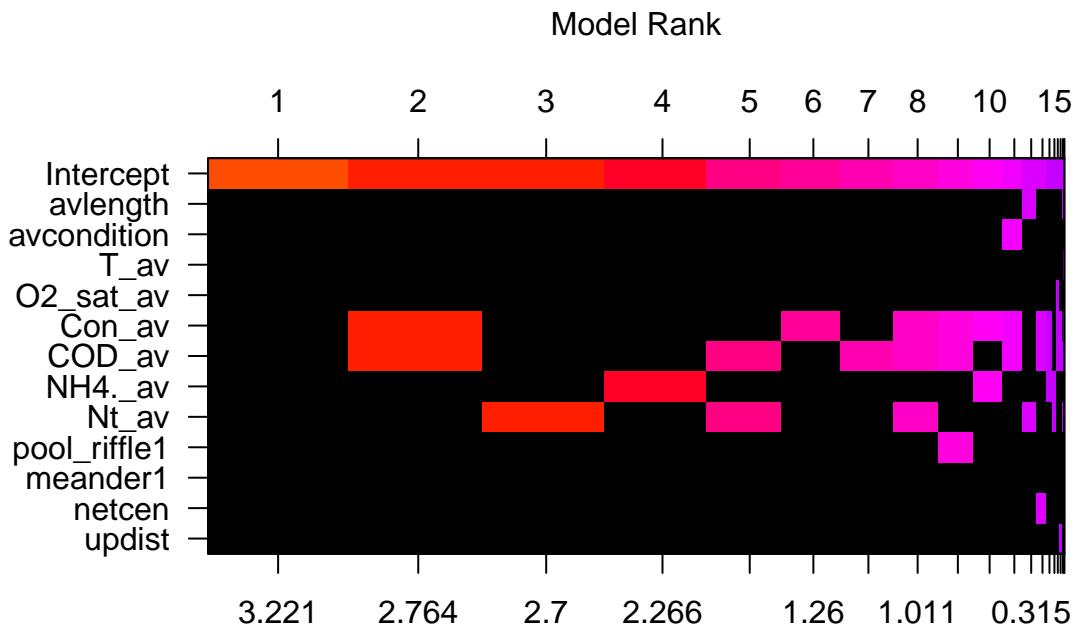
```
bas.model <- bas.lm(avab$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```











### Log Posterior Odds

```

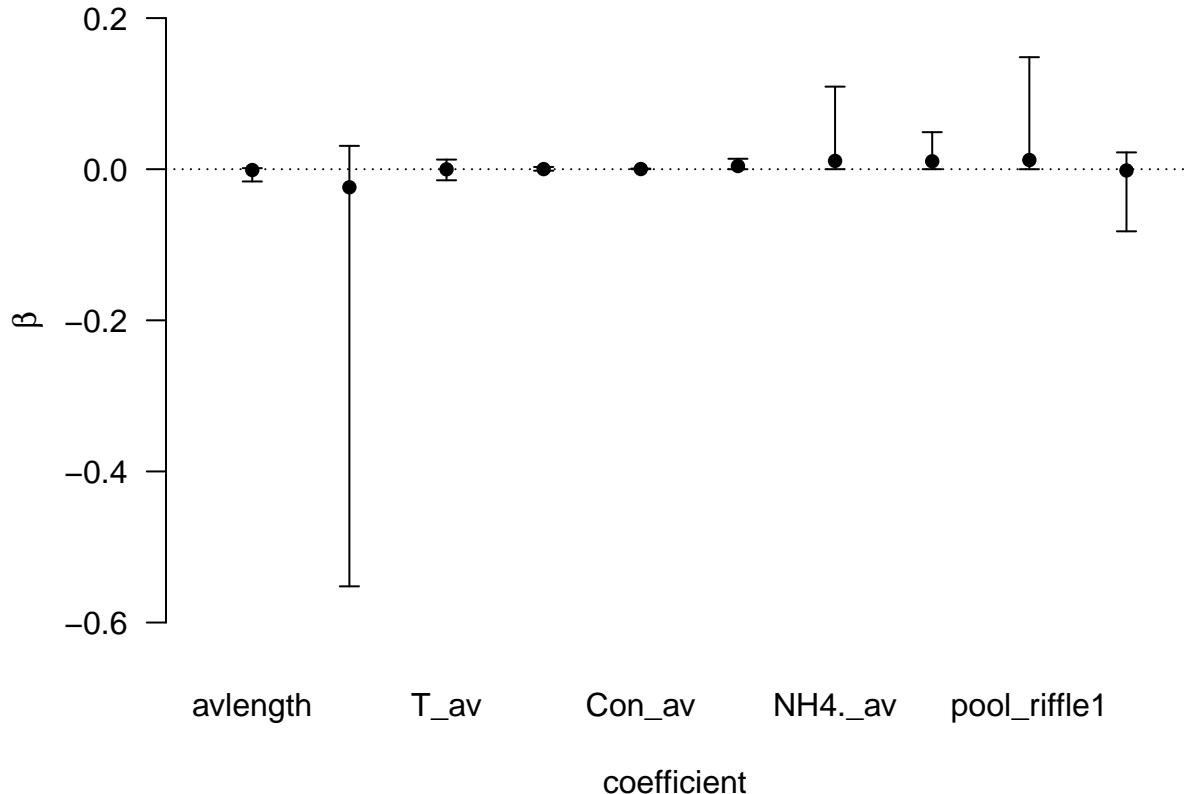
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept     1.282053e-01 2.732163e-01 2.002413e-01
## avlength     -1.660924e-02 6.920726e-04 -1.114390e-03
## avcondition  -5.922113e-01 4.148719e-02 -2.394139e-02
## T_av         -1.540283e-02 1.304251e-02 -1.479738e-04
## O2_sat_av   -1.682859e-03 3.136282e-03 1.132458e-04
## Con_av        0.000000e+00 7.760180e-04 2.109076e-04
## COD_av        0.000000e+00 1.380053e-02 4.203136e-03
## NH4._av      0.000000e+00 1.061486e-01 1.096330e-02
## Nt_av         0.000000e+00 4.954751e-02 1.044647e-02
## pool_riffle1 -5.691153e-04 1.431913e-01 1.212839e-02
## meander1     -7.487026e-02 2.205462e-02 -1.678368e-03
## netcen        -6.088326e-06 1.357290e-06 -2.846749e-07
## updist       -9.831469e-07 7.876887e-07 -1.915234e-08
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

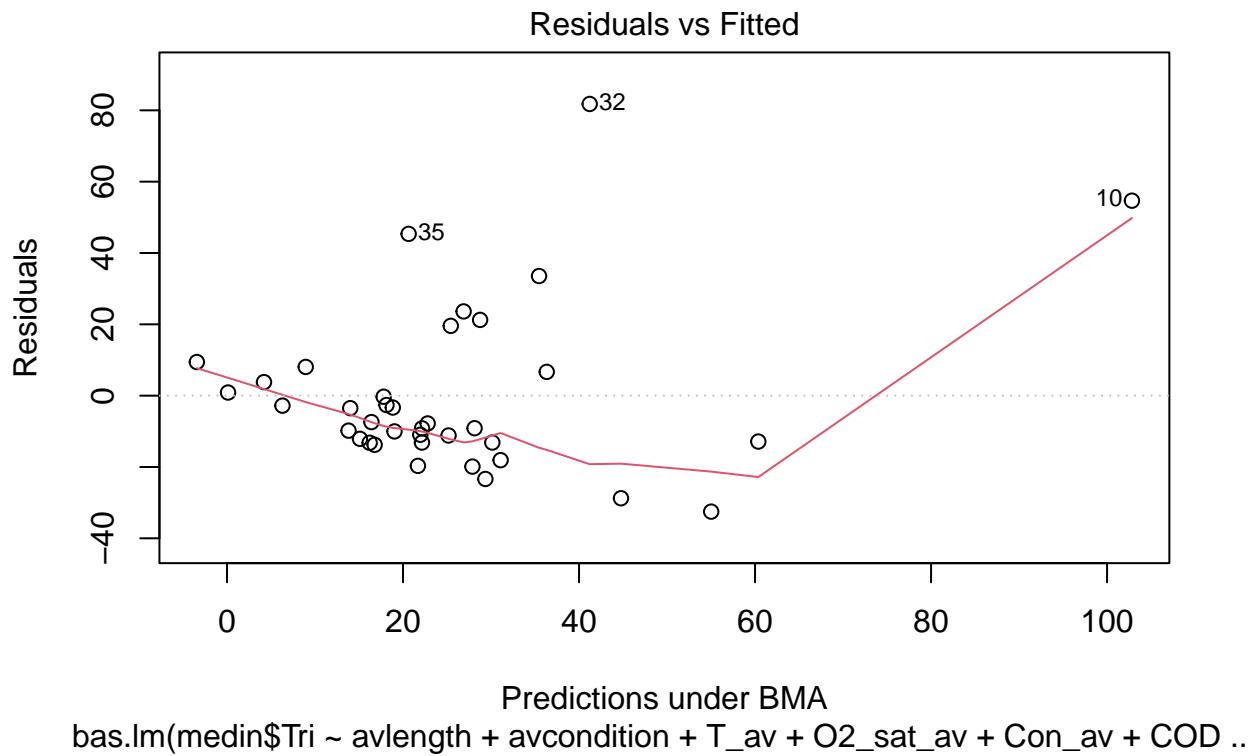
```
plot(confint(coef.model, parm = 2:11))
```

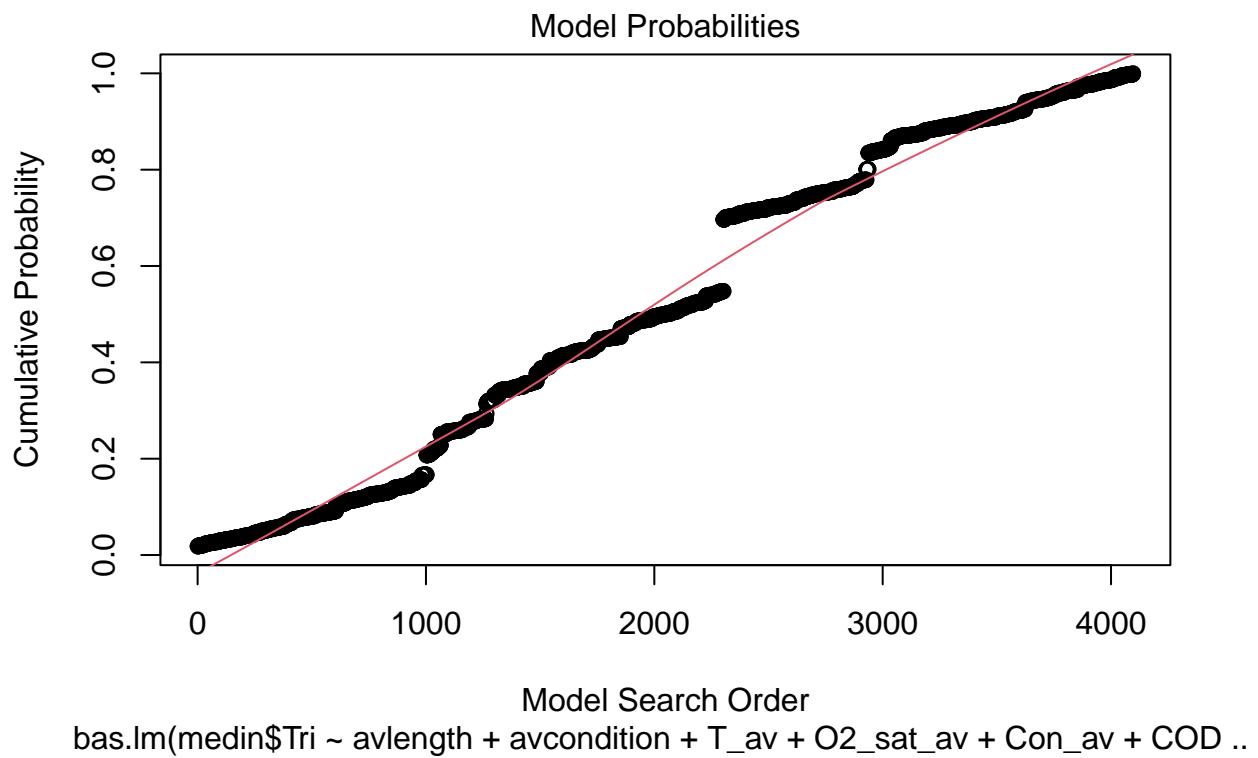


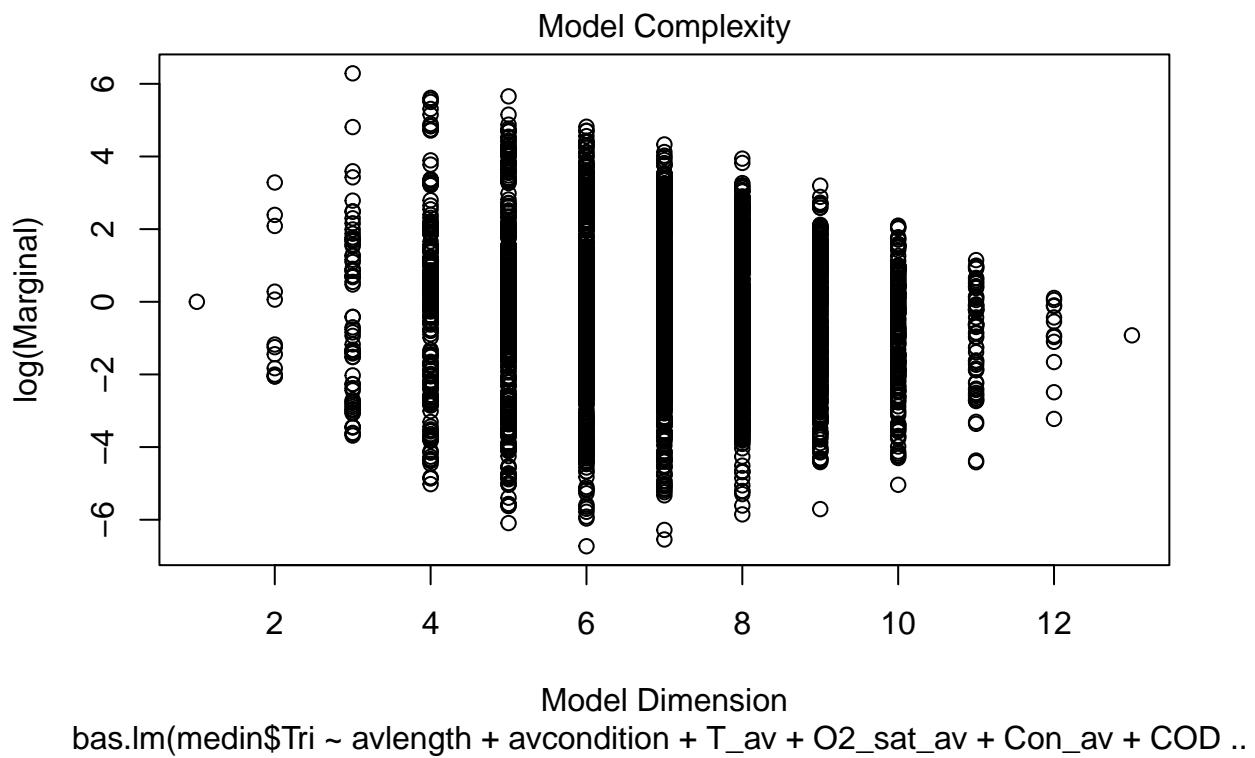
```
## NULL  
confint <- confint(coef.model, parm = 2:11)  
write.table(confint, "GyroAA.txt", sep = "\t")  
pip <- summary(bas.model)  
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

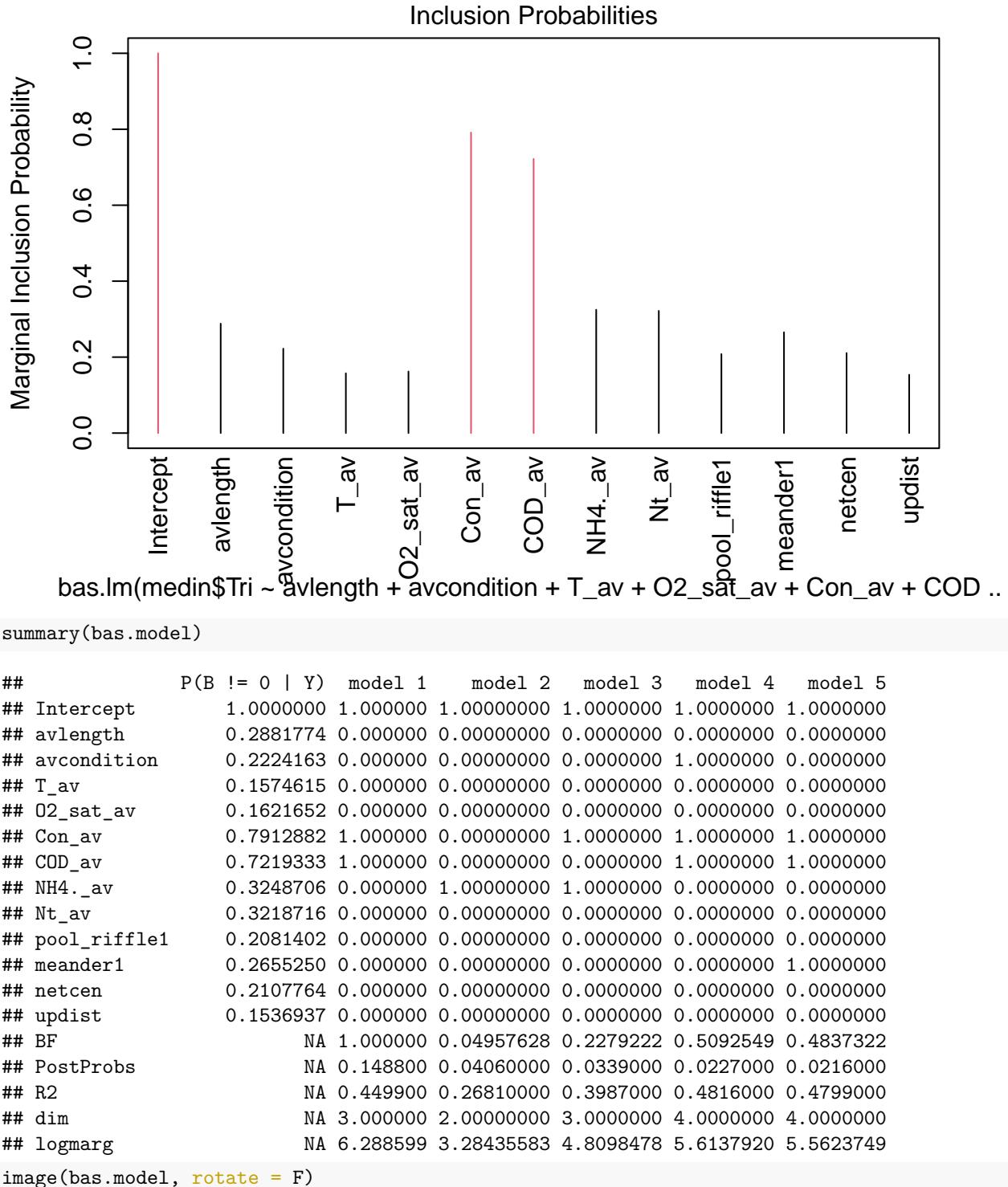
#### 6.4.1 Median infection intensity

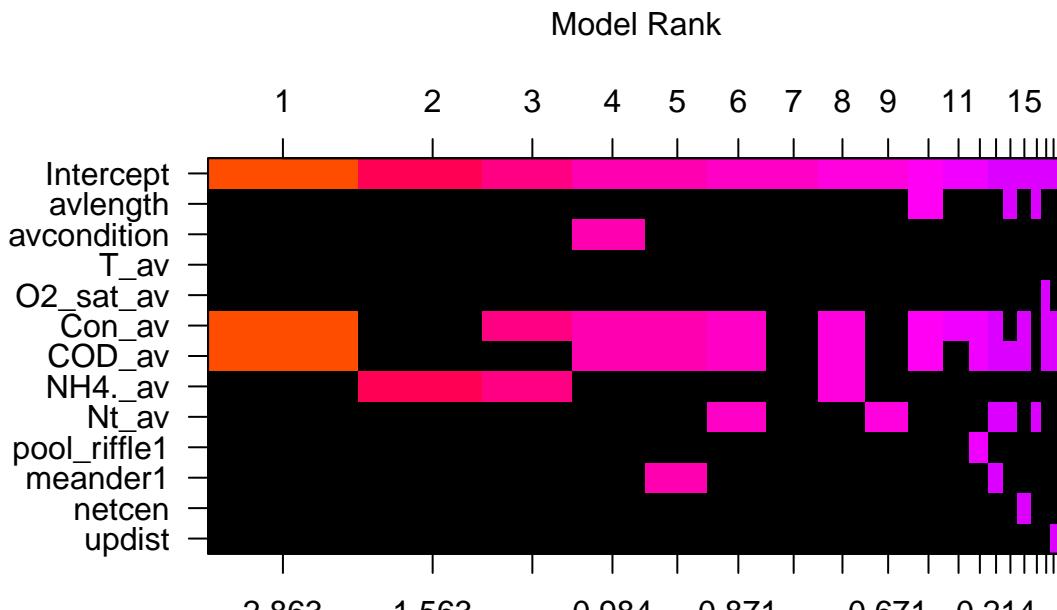
```
bas.model <- bas.lm(medin$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +  
COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,  
prior = "JZS")  
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA  
r = bas.model$Y - yhat #these are the model residuals  
plot(bas.model)
```











```

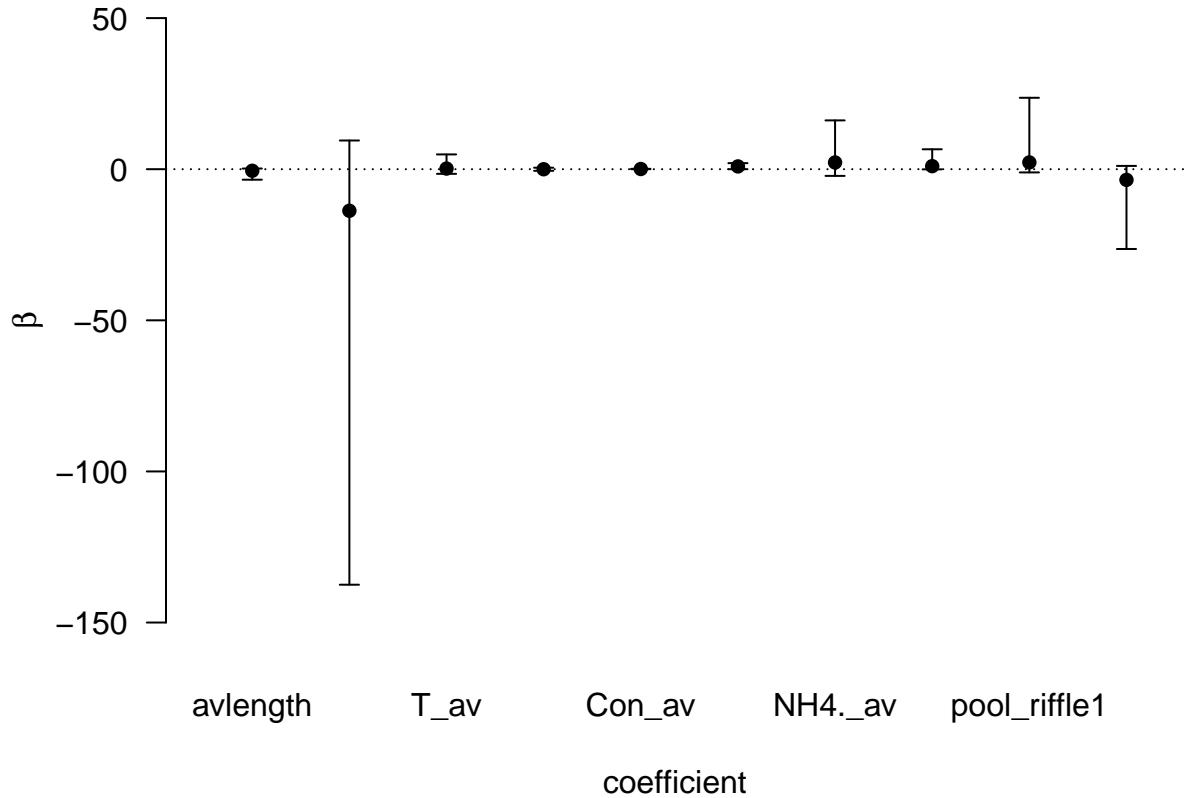
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept     1.633240e+01 3.399173e+01 2.547297e+01
## avlength     -3.616474e+00 9.245391e-02 -5.158998e-01
## avcondition   -1.406368e+02 9.354321e+00 -1.375058e+01
## T_av          -1.784648e+00 4.421906e+00 2.065105e-01
## O2_sat_av    -5.101547e-01 4.984247e-01 -1.498026e-02
## Con_av         0.000000e+00 1.274043e-01 6.585727e-02
## COD_av         0.000000e+00 1.986147e+00 9.195418e-01
## NH4._av       -2.499706e+00 1.546575e+01 2.222211e+00
## Nt_av          -6.514337e-02 6.447830e+00 1.012166e+00
## pool_riffle1  -8.297595e-01 2.475500e+01 2.247962e+00
## meander1      -2.599753e+01 1.738938e+00 -3.523477e+00
## netcen        -1.455727e-03 1.759921e-04 -1.366046e-04
## updist        -2.309218e-04 2.835031e-04 5.653370e-06
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

```
plot(confint(coef.model, parm = 2:11))
```

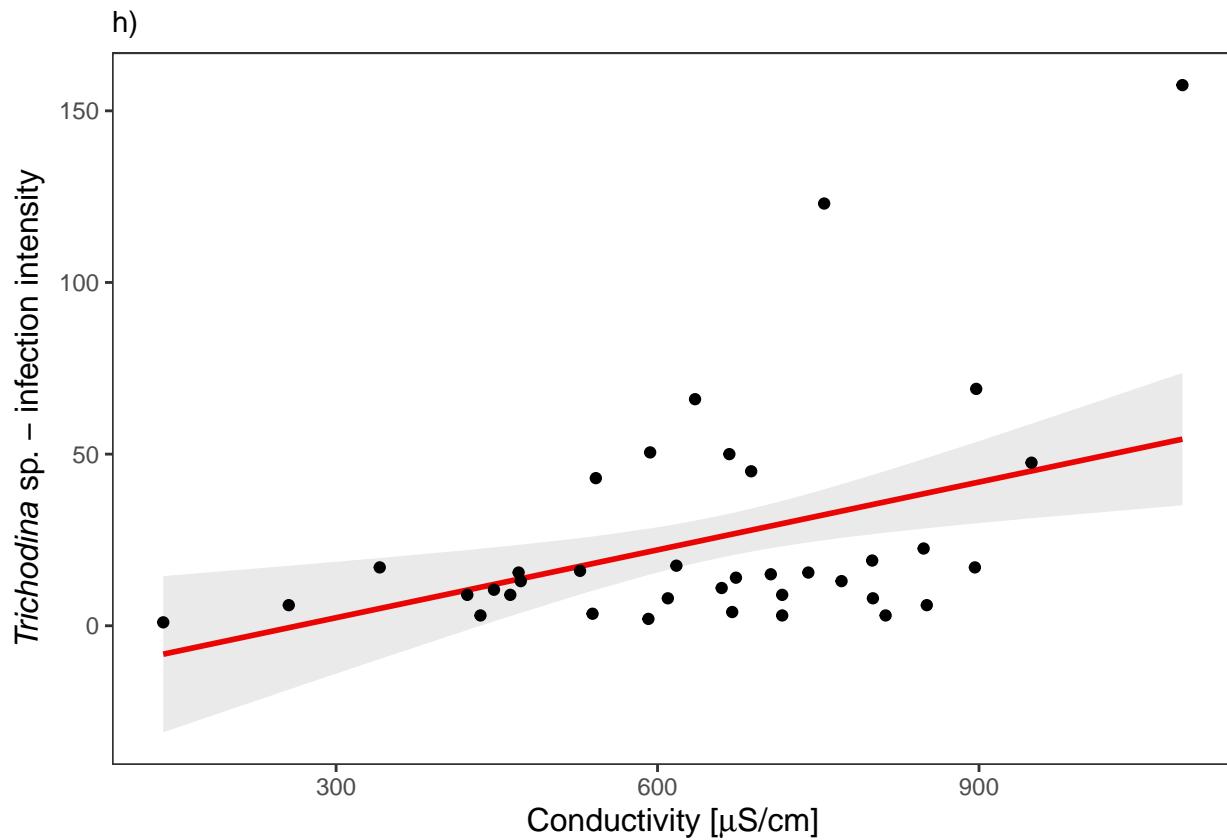


```
## NULL
confint <- confint(coef.model, parm = 2:11)
write.table(confint, "GyroAA.txt", sep = "\t")
pip <- summary(bas.model)
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

### 6.3.2.3 Prediction plot for marginal effect of conductivity on median infection intensity with Trichodina

```
newdata1 <- newdata
newdata1[, "Con_av"] <- environment2$Con_av
BMA_Tri_Con_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3h = ggplot(environment2, aes(Con_av, BMA_Tri_Con_av$fit)) + theme_bw() + geom_line(color = "red",
size = 1) + geom_ribbon(aes(ymin = (BMA_Tri_Con_av$fit - BMA_Tri_Con_av$se.bma.fit),
ymax = (BMA_Tri_Con_av$fit + BMA_Tri_Con_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environment2,
aes(x = Con_av, y = medin$Tri)) + labs(x = expression(paste("Conductivity [",
mu, "S/cm]")), y = expression(paste(italic("Trichodina"), " sp. - infection intensity")))) +
theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
labs(subtitle = "h")
```

figure3h

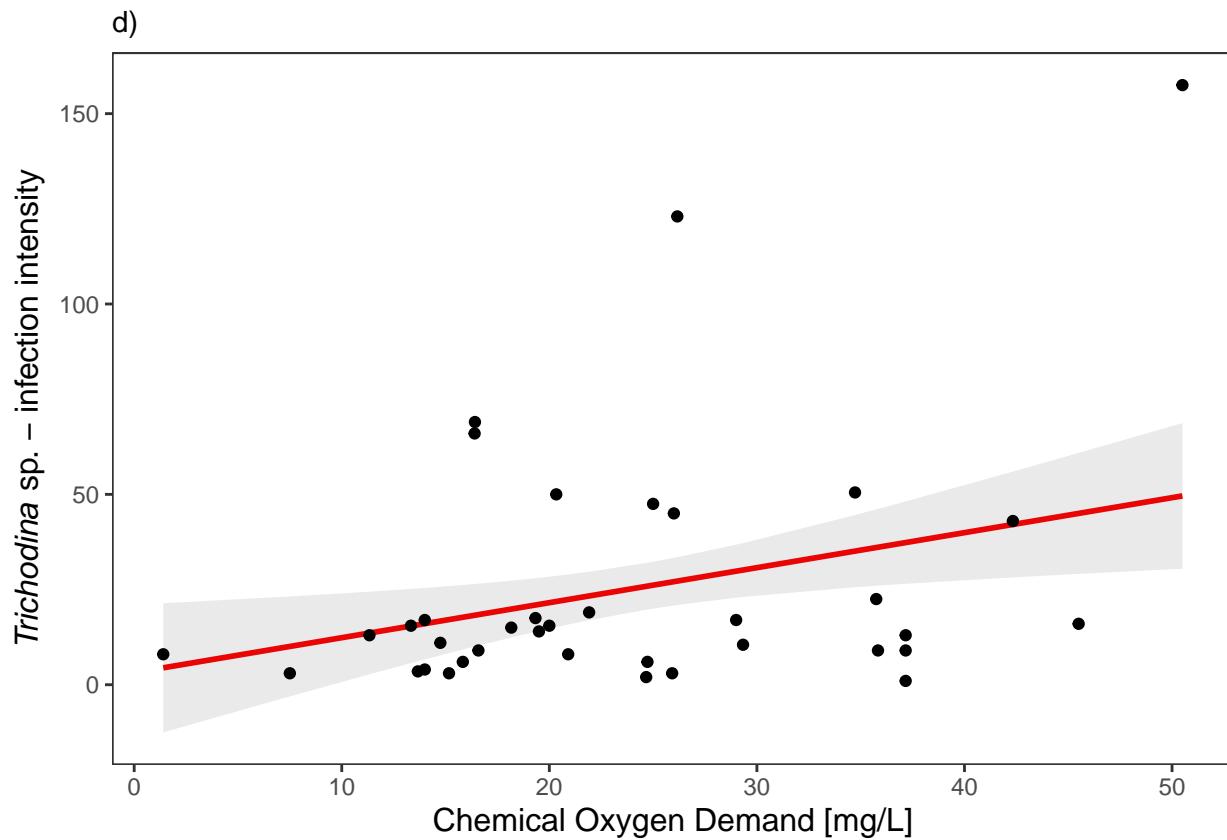


### 6.3.2.3 Prediction plot for marginal effect of COD on median infection intensity with *Trichodina*

```

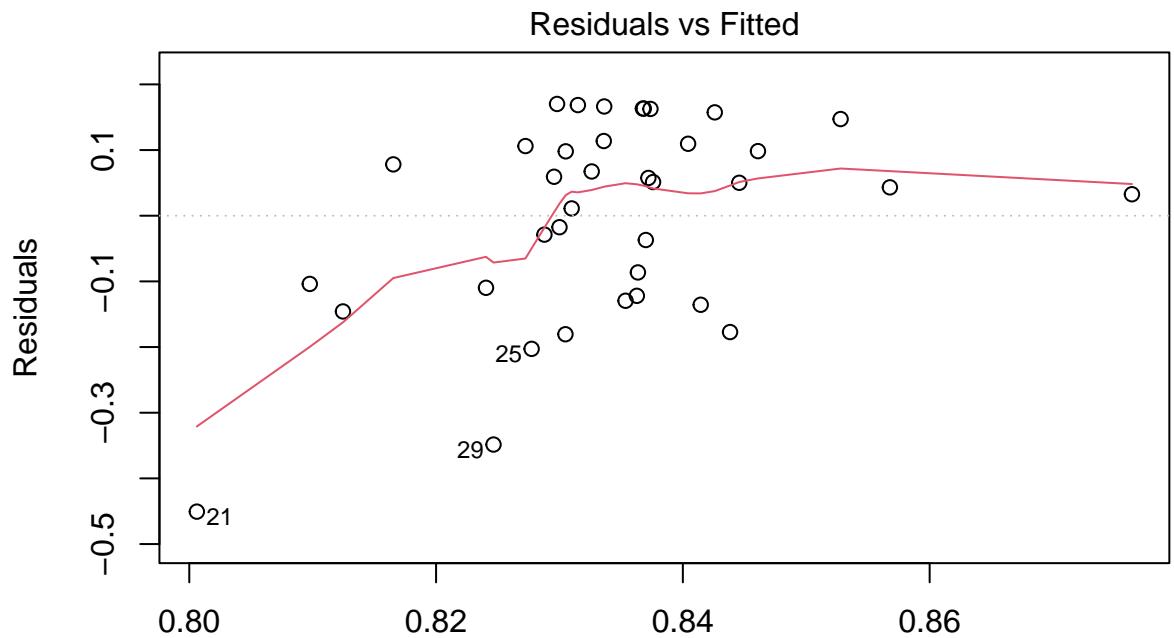
newdata1 <- newdata
newdata1[, "COD_av"] <- environment2$COD_av
BMA_Tri_COD_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3d = ggplot(environment2, aes(COD_av, BMA_Tri_COD_av$fit)) + theme_bw() + geom_line(color = "red",
  size = 1) + geom_ribbon(aes(ymin = (BMA_Tri_COD_av$fit - BMA_Tri_COD_av$se.bma.fit),
  ymax = (BMA_Tri_COD_av$fit + BMA_Tri_COD_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environment2,
  aes(x = COD_av, y = medin$Tri)) + labs(x = expression(paste("Chemical Oxygen Demand [mg/L]")),
  y = expression(paste(italic("Trichodina"), " sp. - infection intensity"))) +
  theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "d"))
figure3d

```



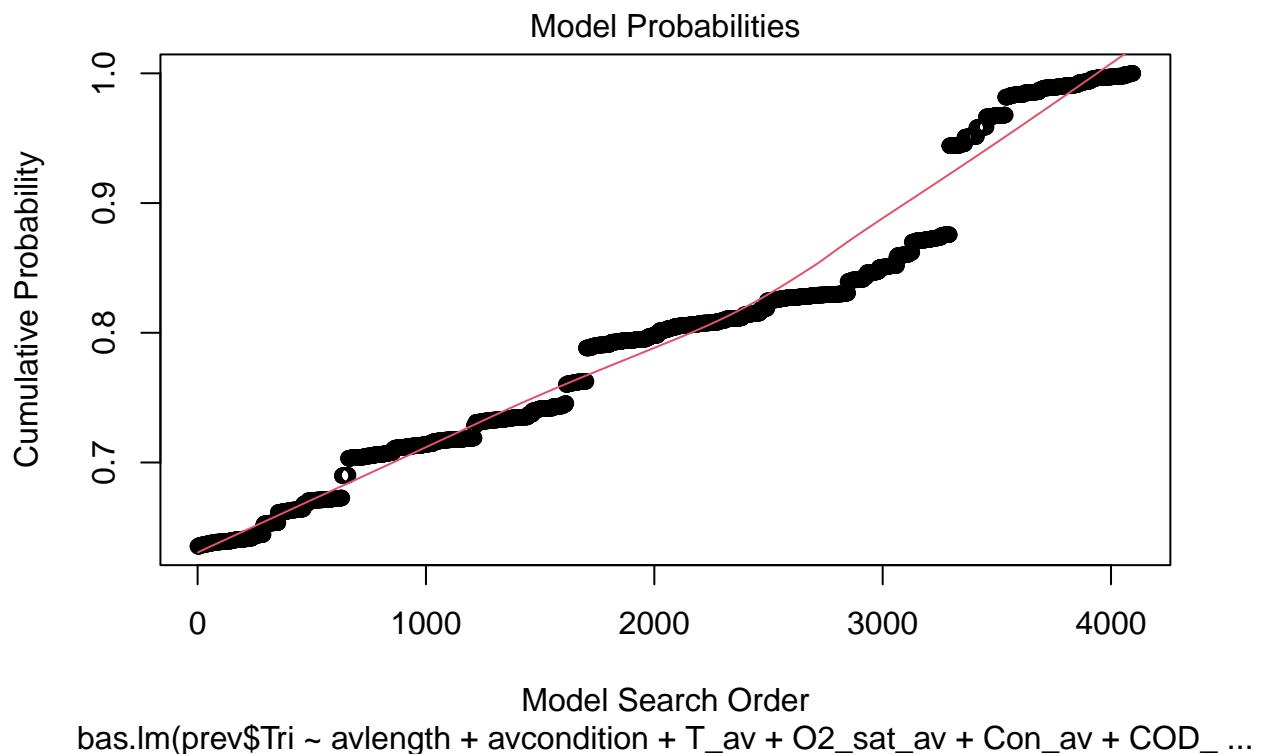
#### 6.4.1 Prevalence

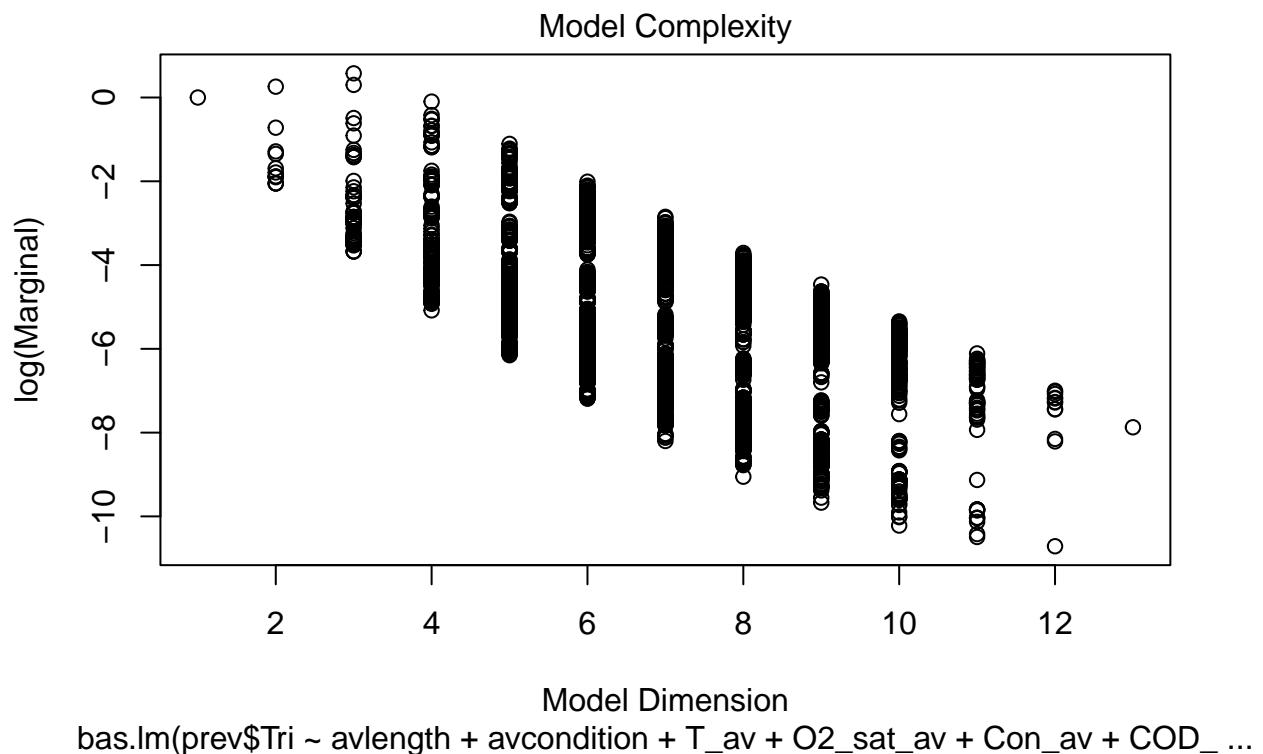
```
bas.model <- bas.lm(prev$Tri ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```

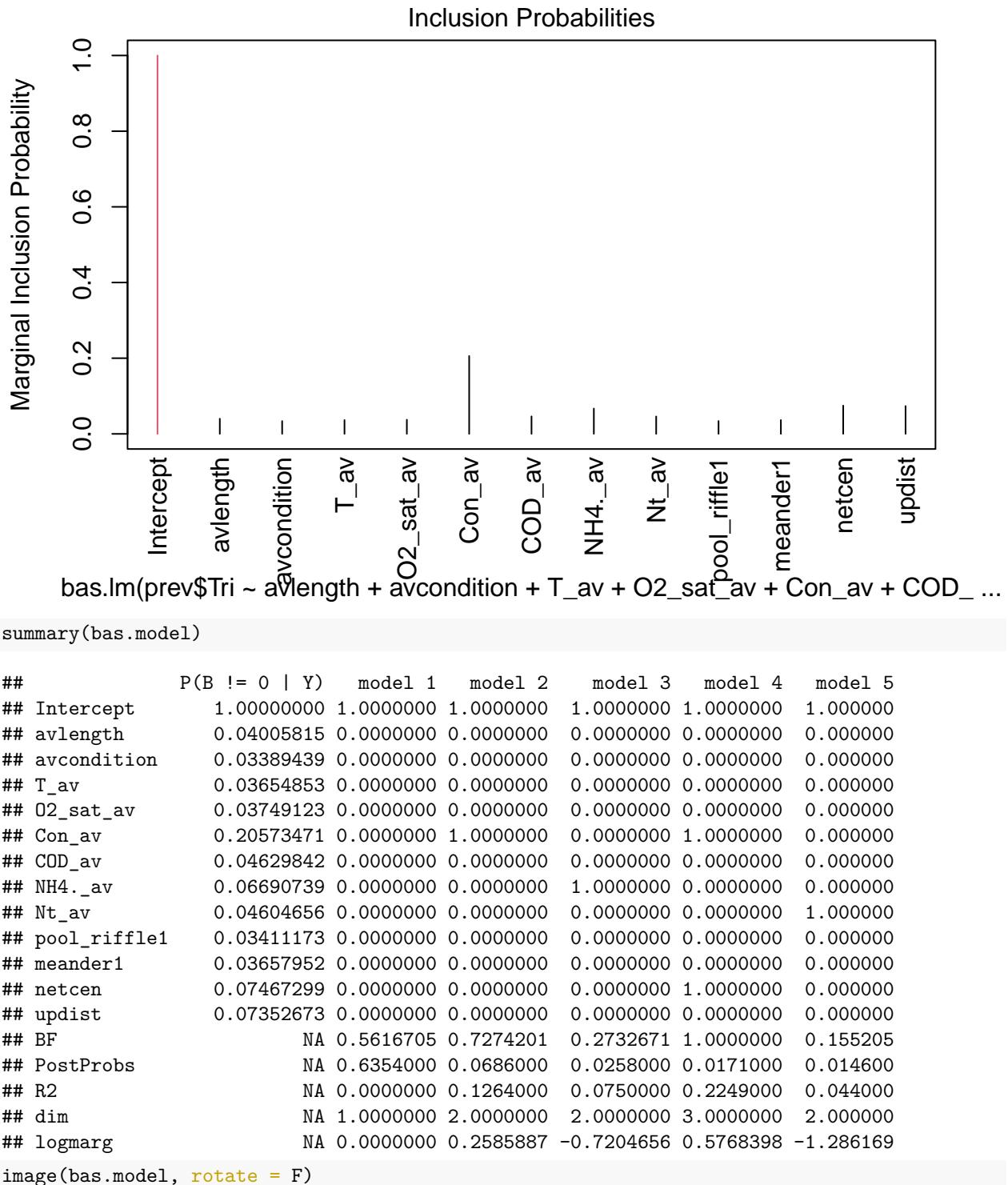


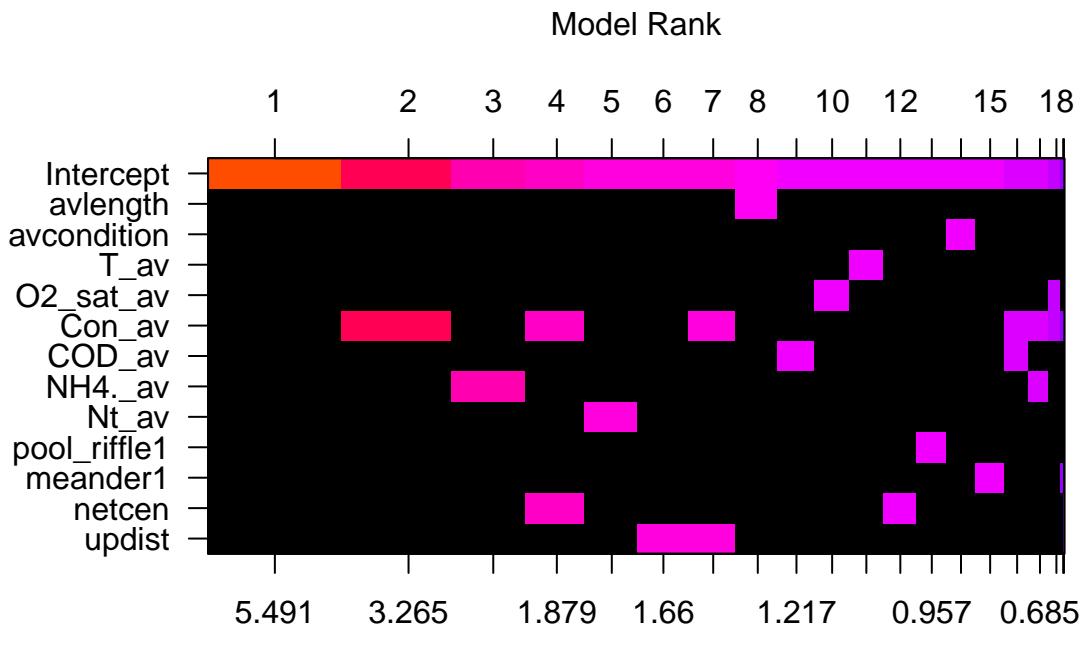
Predictions under BMA

bas.lm(prev\$Tri ~ avlength + avcondition + T\_av + O2\_sat\_av + Con\_av + COD\_ ...)









```

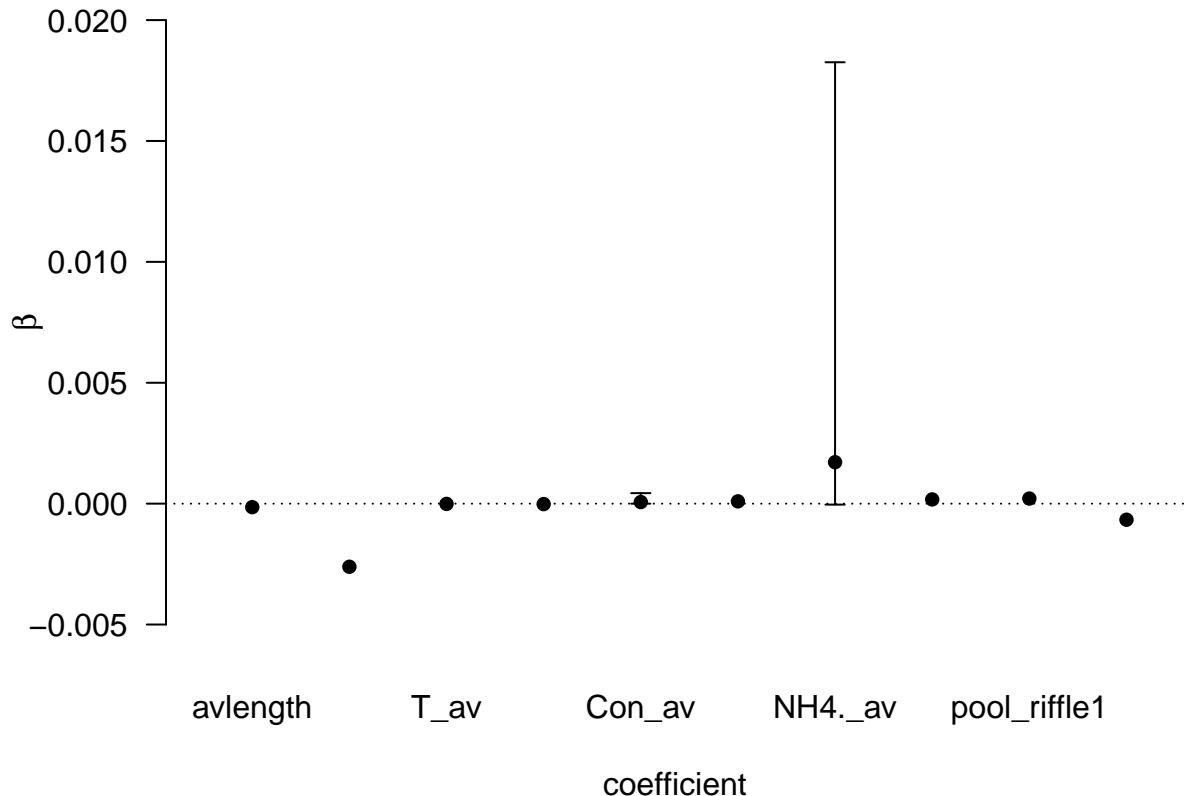
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept    7.828285e-01 8.850286e-01 8.340580e-01
## avlength     0.000000e+00 0.000000e+00 -1.453558e-04
## avcondition  0.000000e+00 0.000000e+00 -2.610917e-03
## T_av         0.000000e+00 0.000000e+00 -1.135202e-05
## O2_sat_av   0.000000e+00 0.000000e+00 -1.893466e-05
## Con_av       0.000000e+00 4.345984e-04  6.570599e-05
## COD_av       0.000000e+00 0.000000e+00  9.451636e-05
## NH4_av        0.000000e+00 1.957295e-02  1.713706e-03
## Nt_av         0.000000e+00 0.000000e+00  1.739937e-04
## pool_riffle1 0.000000e+00 0.000000e+00  2.104077e-04
## meander1     0.000000e+00 0.000000e+00 -6.652944e-04
## netcen        -3.605500e-06 1.340568e-07 -3.317353e-07
## updist        -1.290218e-06 1.989046e-08 -1.309710e-07
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

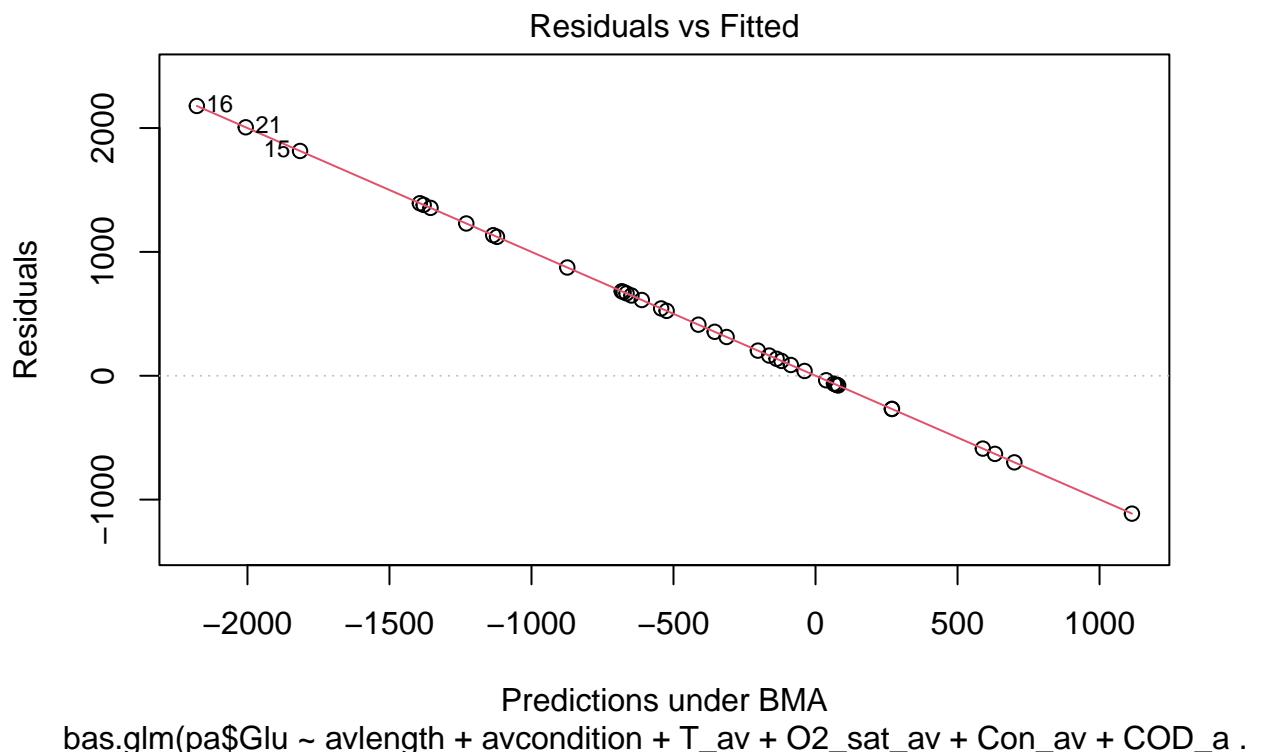
```
plot(confint(coef.model, parm = 2:11))
```

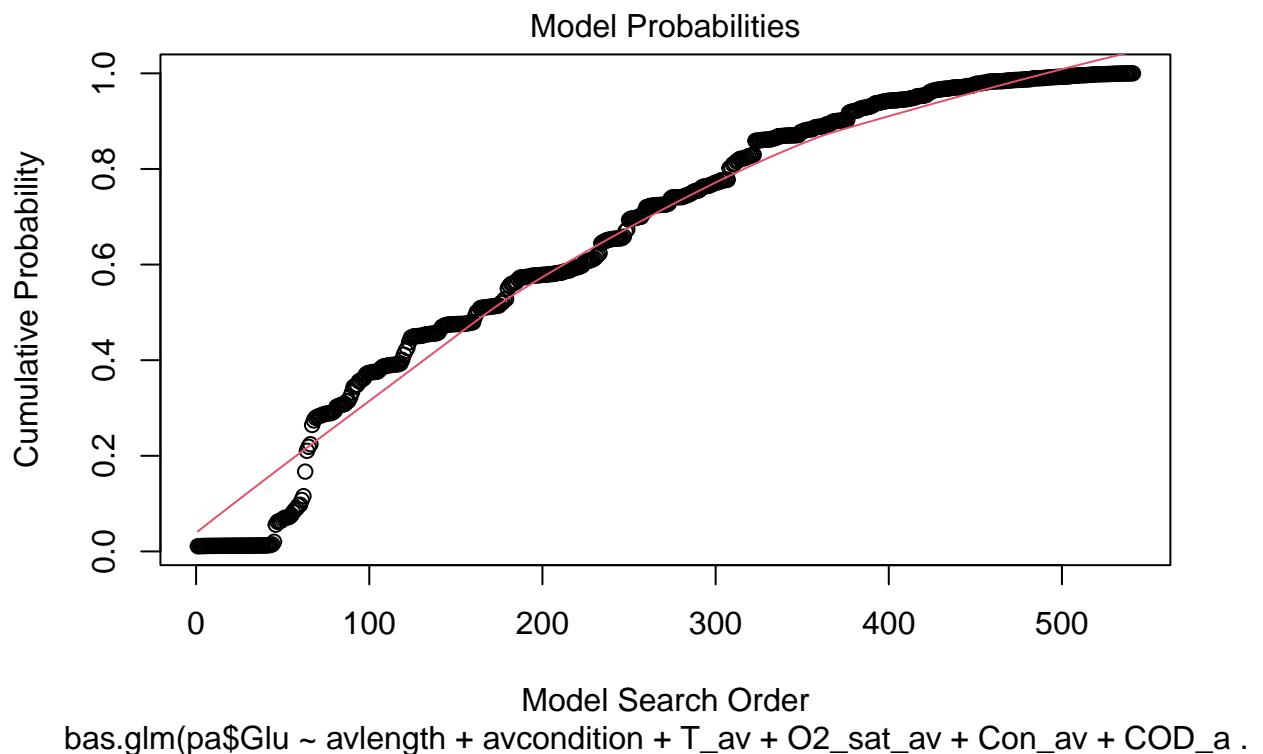


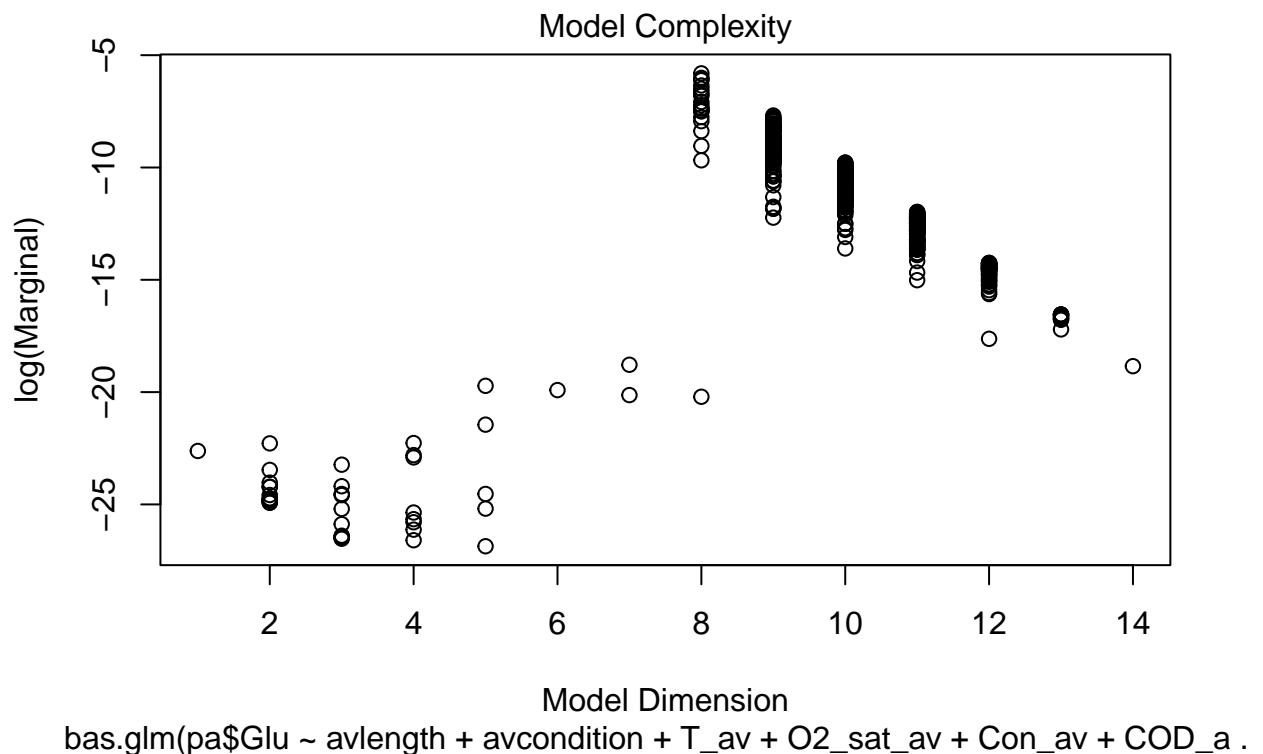
```
## NULL  
confint <- confint(coef.model, parm = 2:11)  
write.table(confint, "GyroAA.txt", sep = "\t")  
pip <- summary(bas.model)  
PIP[c(1:12), 3] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

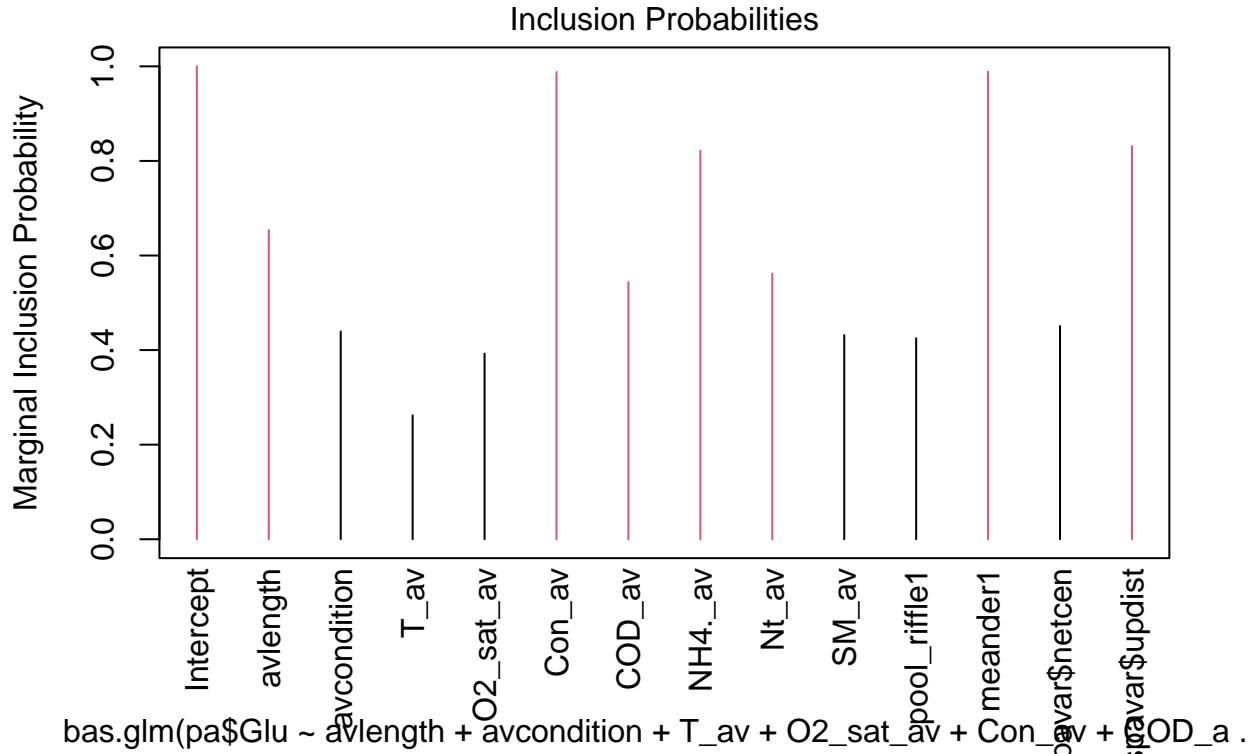
## 6.5 Variation in Glugea infection

```
bas.model <- bas.glm(pa$Glu ~ avlength + avcondition + T_av + O2_sat_av + Con_av +  
COD_av + NH4._av + Nt_av + SM_av + pool_riffle + meander + spavar$netcen + spavar$updist,  
data = environment2, betaprior = g.prior(100), family = binomial)  
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA  
r = bas.model$Y - yhat #these are the model residuals  
plot(bas.model)
```









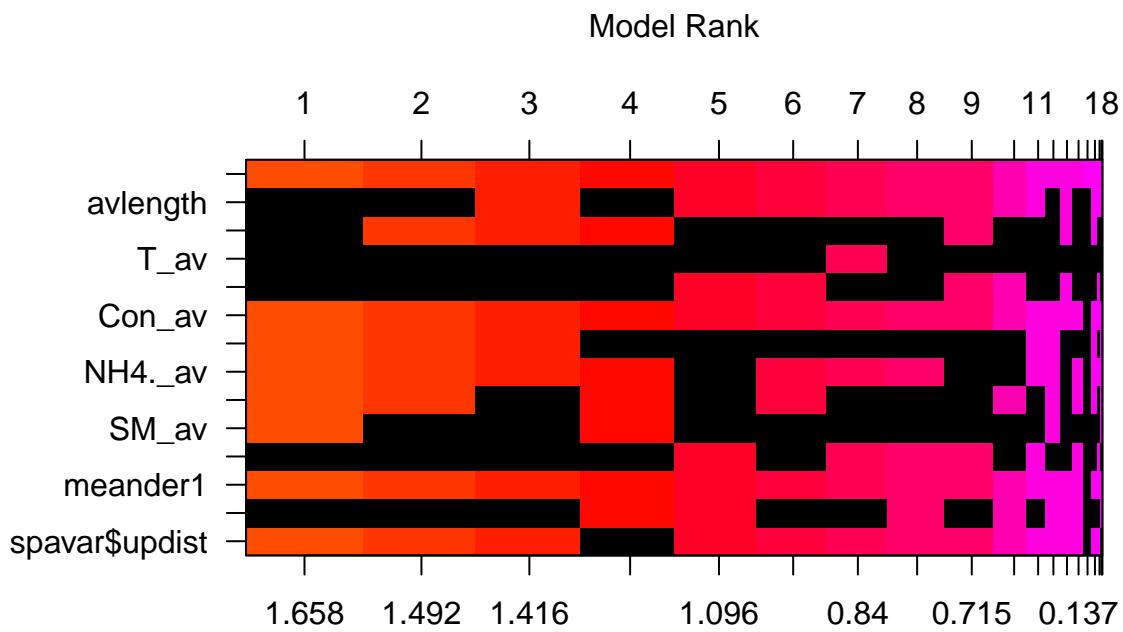
```
summary(bas.model)
```

	P(B != 0   Y)	model 1	model 2	model 3	model 4
## Intercept	1.0000000	1.000000	1.0000000	1.0000000	1.000000
## avlength	0.6541138	0.000000	0.0000000	1.0000000	0.000000
## avcondition	0.4392700	0.000000	1.0000000	1.0000000	1.000000
## T_av	0.2622314	0.000000	0.0000000	0.0000000	0.000000
## O2_sat_av	0.3927490	0.000000	0.0000000	0.0000000	0.000000
## Con_av	0.9879272	1.000000	1.0000000	1.0000000	1.000000
## COD_av	0.5438843	1.000000	1.0000000	1.0000000	0.000000
## NH4._av	0.8217651	1.000000	1.0000000	1.0000000	1.000000
## Nt_av	0.5617798	1.000000	1.0000000	0.0000000	1.000000
## SM_av	0.4316895	1.000000	0.0000000	0.0000000	1.000000
## pool_riffle1	0.4249878	0.000000	0.0000000	0.0000000	0.000000
## meander1	0.9885742	1.000000	1.0000000	1.0000000	1.000000
## spavar\$netcen	0.4508301	0.000000	0.0000000	0.0000000	1.000000
## spavar\$updist	0.8314209	1.000000	1.0000000	1.0000000	0.000000
## BF	NA	1.000000	0.7351301	0.8189871	0.769804
## PostProbs	NA	0.051300	0.0435000	0.0403000	0.035800
## R2	NA	1.000000	1.0000000	1.0000000	1.000000
## dim	NA	8.000000	8.0000000	8.0000000	8.000000
## logmarg	NA	-5.808403	-6.1161110	-6.0080902	-6.070023
##		model 5			
## Intercept	1.0000000				
## avlength	1.0000000				
## avcondition	0.0000000				

```

## T_av          0.0000000
## O2_sat_av   1.0000000
## Con_av       1.0000000
## COD_av       0.0000000
## NH4._av      0.0000000
## Nt_av        0.0000000
## SM_av        0.0000000
## pool_riffle1 1.0000000
## meander1     1.0000000
## spavar$netcen 1.0000000
## spavar$updist 1.0000000
## BF           0.5830791
## PostProbs    0.0292000
## R2           1.0000000
## dim          8.0000000
## logmarg      -6.3478357
image(bas.model, rotate = F)

```



### Log Posterior Odds

```

coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] FALSE FALSE
## [13] FALSE FALSE
confint(coef.model)

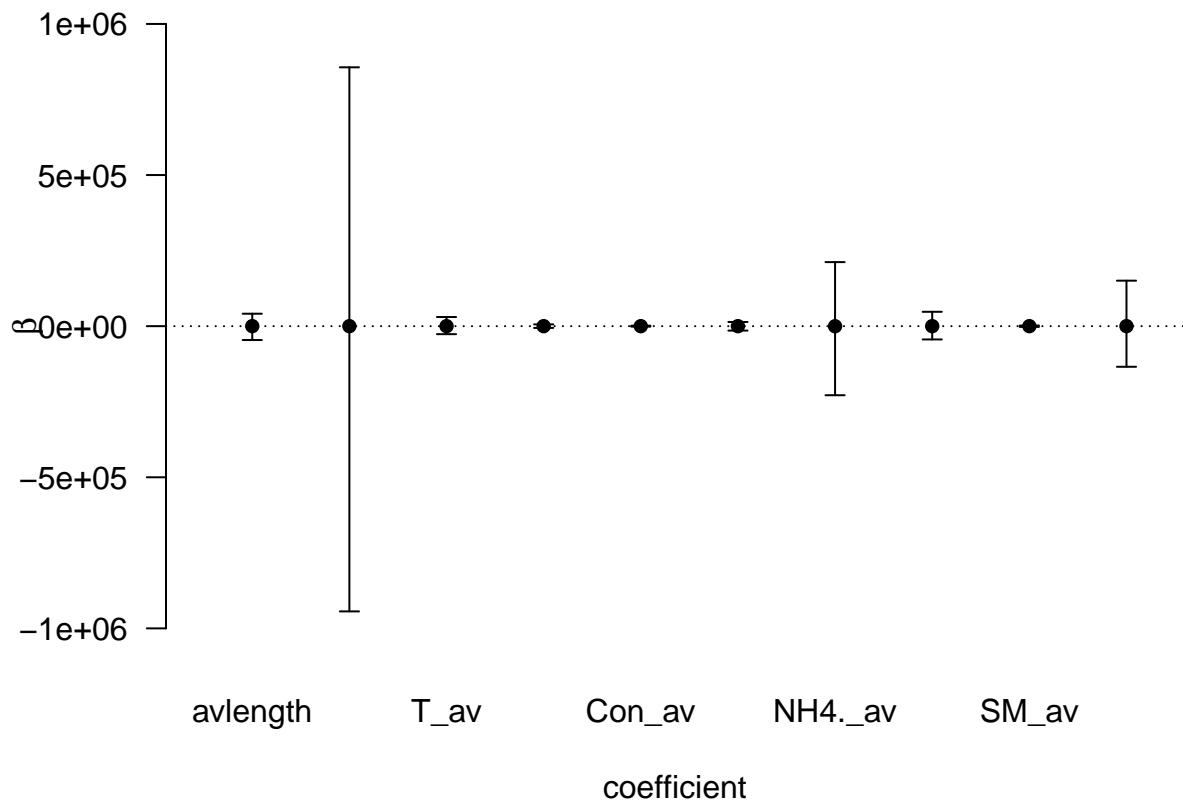
##                                2.5%      97.5%      beta

```

```

## Intercept      -2.648440e+06 2.950111e+06 -6.043606e+03
## avlength     -4.152797e+04 4.137963e+04  8.595722e+01
## avcondition   -9.060787e+05 8.374943e+05 -2.838759e+01
## T_av          -2.905210e+04 2.629954e+04  1.523151e+01
## O2_sat_av    -5.757385e+03 5.461031e+03  8.560397e+00
## Con_av        -1.469834e+03 1.620618e+03  4.207847e+00
## COD_av        -1.363519e+04 1.486630e+04  9.613528e+00
## NH4._av       -2.152276e+05 2.180475e+05 -2.655528e+02
## Nt_av          -4.446233e+04 4.405396e+04  3.648137e+01
## SM_av          -1.973583e+03 2.184182e+03 -1.126352e+00
## pool_riffle1  -1.379986e+05 1.533153e+05  8.612651e+01
## meander1      -5.013863e+05 4.745751e+05 -1.305012e+03
## spavar$netcen -9.228435e+00 9.249956e+00 -8.456631e-03
## spavar$updist -4.982334e+00 4.866071e+00 -7.944747e-03
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"
plot(confint(coef.model, parm = 2:11))

```



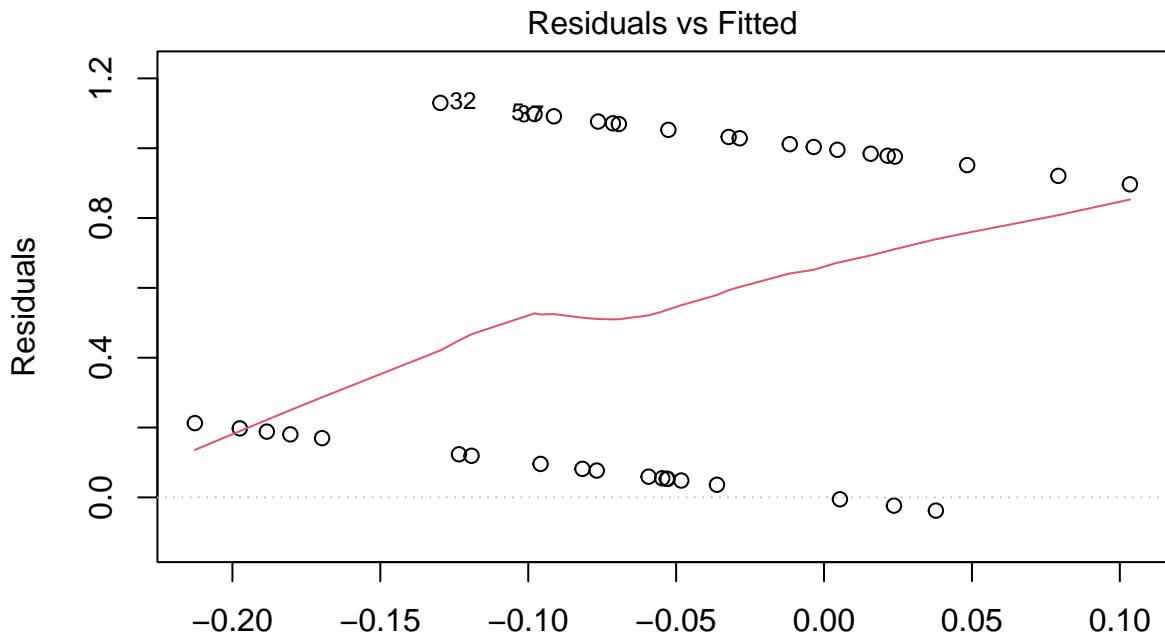
```

## NULL
confint <- confint(coef.model, parm = 2:11)
pip <- summary(bas.model)
PIP[c(1:12), 9] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

```

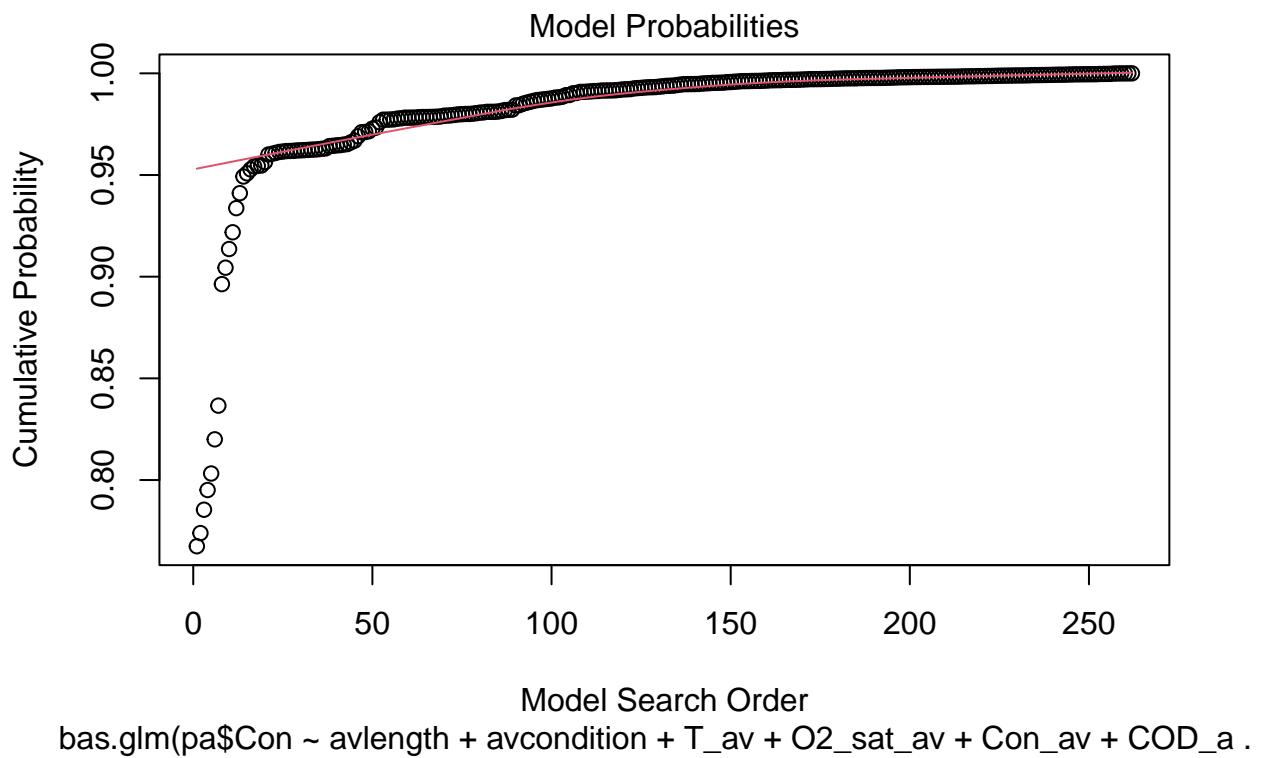
## 6.5 Variation in Contracaecum infection

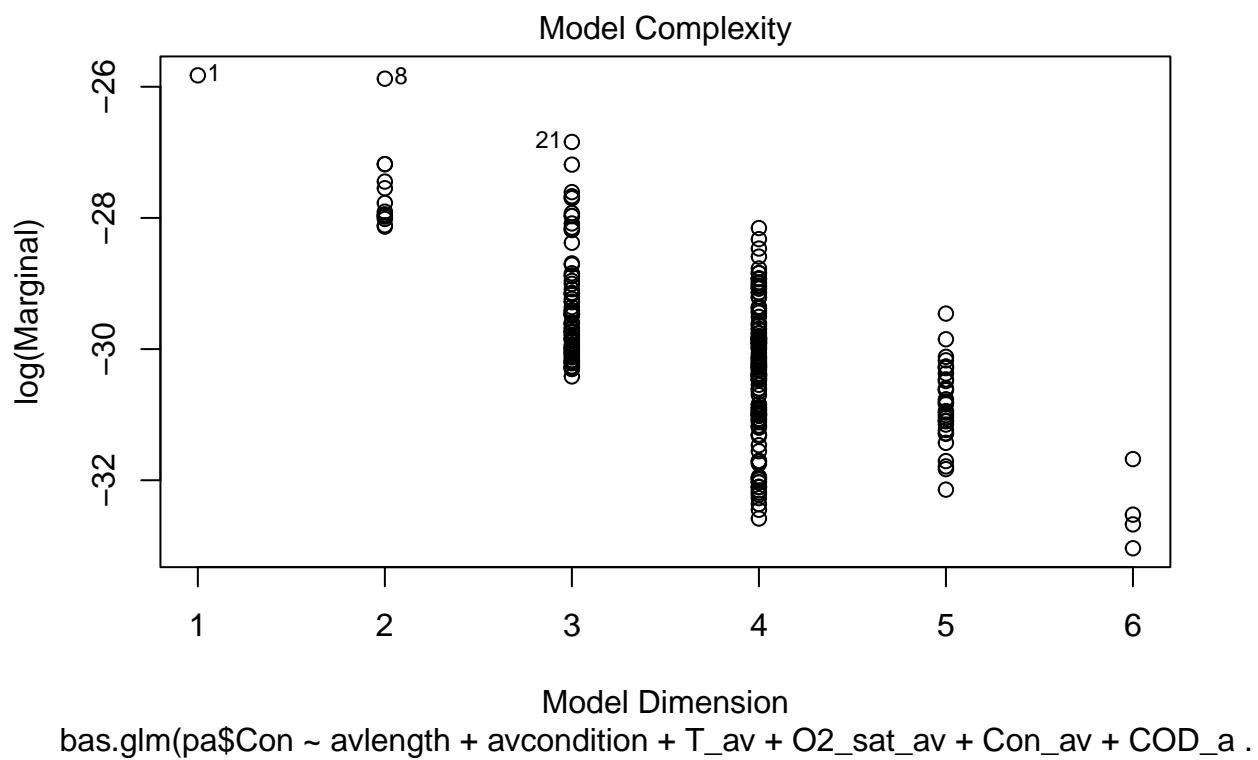
```
bas.model <- bas.glm(pa$Con ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + SM_av + pool_riffle + meander + spavar$netcen + spavar$updist,
  data = environment2, betaprior = g.prior(100), family = binomial)
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```

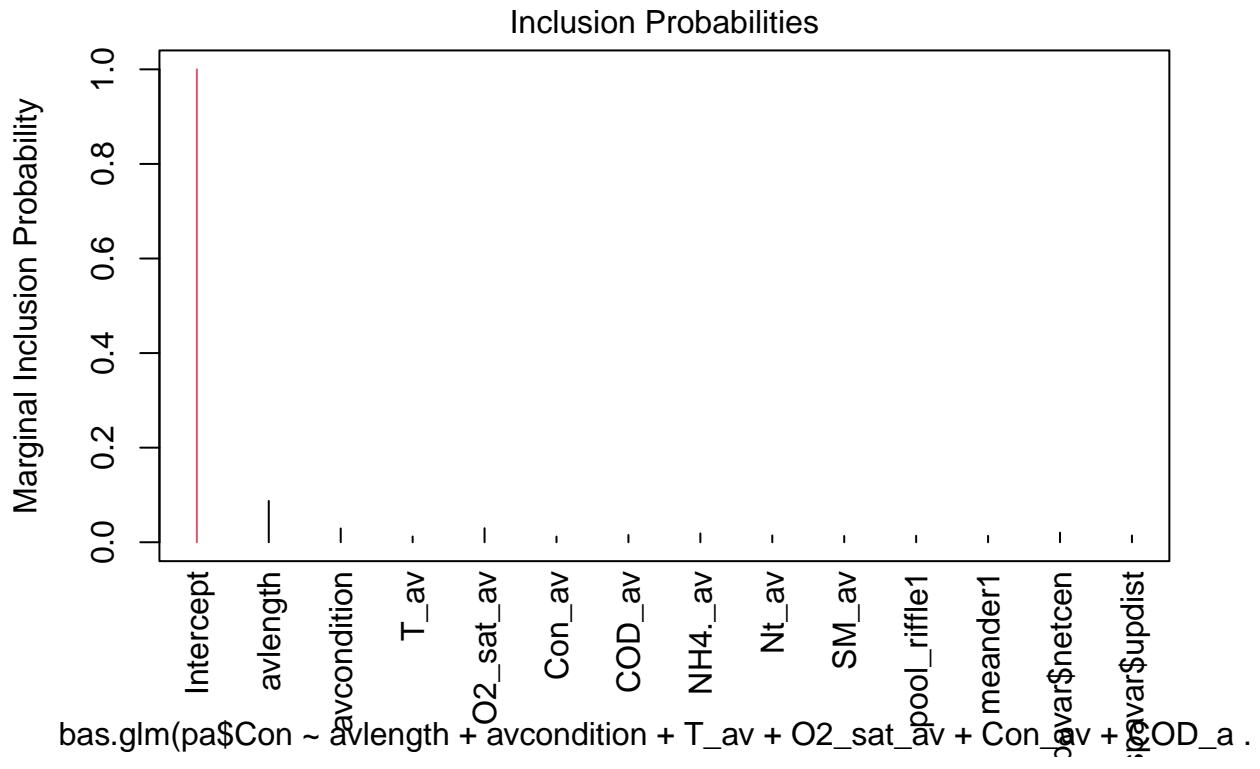


Predictions under BMA

```
bas.glm(pa$Con ~ avlength + avcondition + T_av + O2_sat_av + Con_av + COD_a .
```







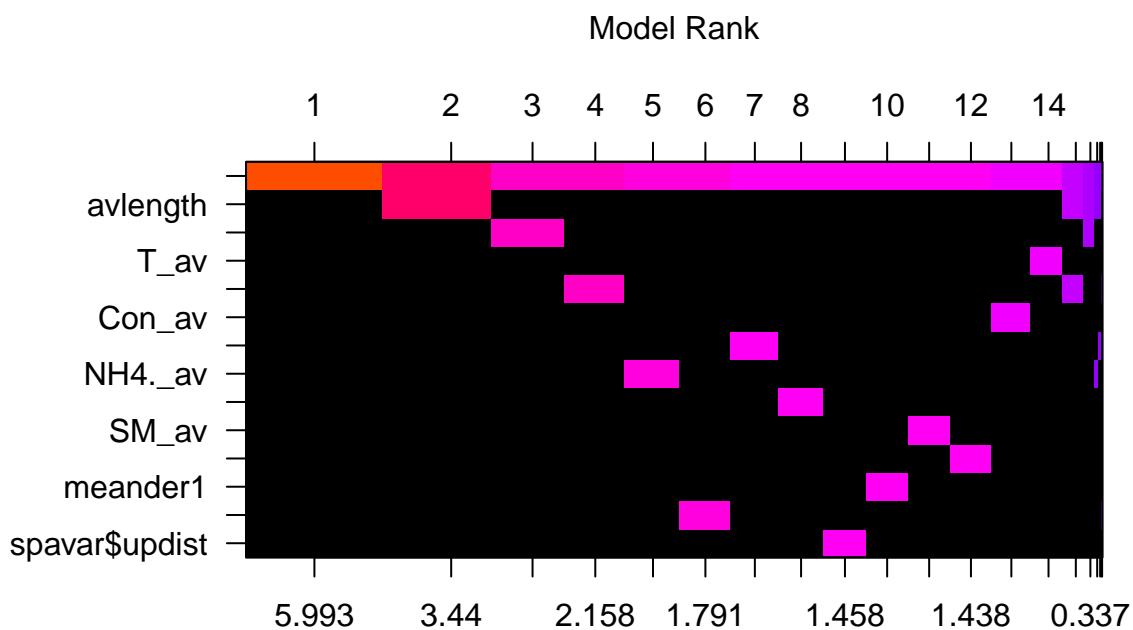
```
summary(bas.model)
```

	P(B != 0   Y)	model 1	model 2	model 3	model 4
## Intercept	1.00000000	1.000000	1.0000000	1.0000000	1.0000000
## avlength	0.08734131	0.000000	1.0000000	0.0000000	0.0000000
## avcondition	0.02913818	0.000000	0.0000000	1.0000000	0.0000000
## T_av	0.01218262	0.000000	0.0000000	0.0000000	0.0000000
## O2_sat_av	0.02960205	0.000000	0.0000000	0.0000000	1.0000000
## Con_av	0.01197510	0.000000	0.0000000	0.0000000	0.0000000
## COD_av	0.01529541	0.000000	0.0000000	0.0000000	0.0000000
## NH4._av	0.01870117	0.000000	0.0000000	0.0000000	0.0000000
## Nt_av	0.01435547	0.000000	0.0000000	0.0000000	0.0000000
## SM_av	0.01322021	0.000000	0.0000000	0.0000000	0.0000000
## pool_riffle1	0.01370850	0.000000	0.0000000	0.0000000	0.0000000
## meander1	0.01358643	0.000000	0.0000000	0.0000000	0.0000000
## spavar\$netcen	0.02033691	0.000000	0.0000000	0.0000000	0.0000000
## spavar\$updist	0.01425781	0.000000	0.0000000	0.0000000	0.0000000
## BF	NA	1.000000	0.9507459	0.2581488	0.2583995
## PostProbs	NA	0.76740	0.0598000	0.0168000	0.0166000
## R2	NA	0.00000	0.0864000	0.0365000	0.0366000
## dim	NA	1.00000	2.0000000	2.0000000	2.0000000
## logmarg	NA	-25.82594	-25.8764468	-27.1801577	-27.1791867
##		model 5			
## Intercept	1.00000				
## avlength	0.00000				
## avcondition	0.00000				

```

## T_av           0.00000
## O2_sat_av    0.00000
## Con_av        0.00000
## COD_av        0.00000
## NH4._av       1.00000
## Nt_av          0.00000
## SM_av          0.00000
## pool_riffle1  0.00000
## meander1       0.00000
## spavar$netcen 0.00000
## spavar$updist 0.00000
## BF             0.19791
## PostProbs     0.01190
## R2             0.02640
## dim            2.00000
## logmarg        -27.44588
image(bas.model, rotate = F)

```



```

coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] FALSE FALSE
## [13] FALSE FALSE
confint(coef.model)

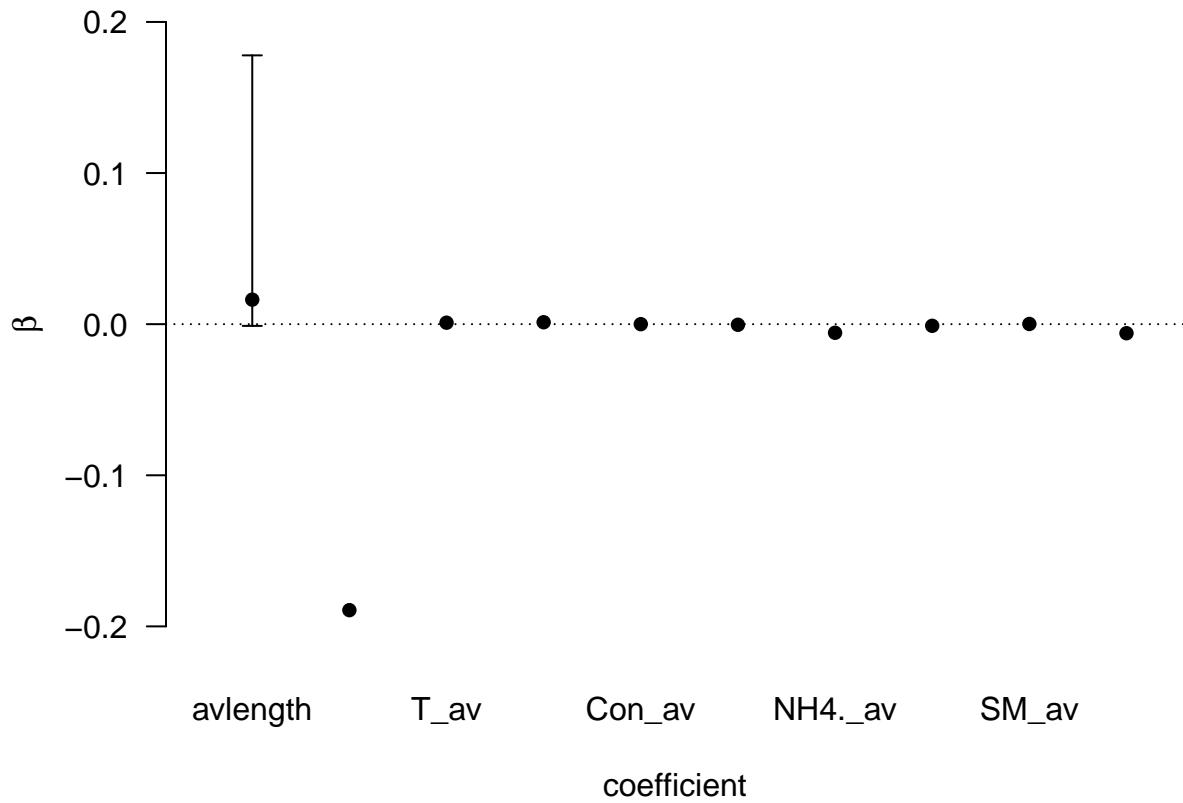
##                2.5%      97.5%      beta

```

```

## Intercept      -9.414584 2.6527821 -5.774662e-01
## avlength       0.000000 0.1743856  1.622454e-02
## avcondition    0.000000 0.0000000 -1.892708e-01
## T_av           0.000000 0.0000000  9.629987e-04
## O2_sat_av     0.000000 0.0000000  1.275636e-03
## Con_av          0.000000 0.0000000  4.794210e-06
## COD_av          0.000000 0.0000000 -4.079285e-04
## NH4._av         0.000000 0.0000000 -5.707037e-03
## Nt_av           0.000000 0.0000000 -1.094052e-03
## SM_av            0.000000 0.0000000  1.275784e-04
## pool_riffle1   0.000000 0.0000000 -5.996030e-03
## meander1        0.000000 0.0000000  3.724159e-03
## spavar$netcen   0.000000 0.0000000 -8.932387e-07
## spavar$updist   0.000000 0.0000000 -1.422514e-07
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"
plot(confint(coef.model, parm = 2:11))

```



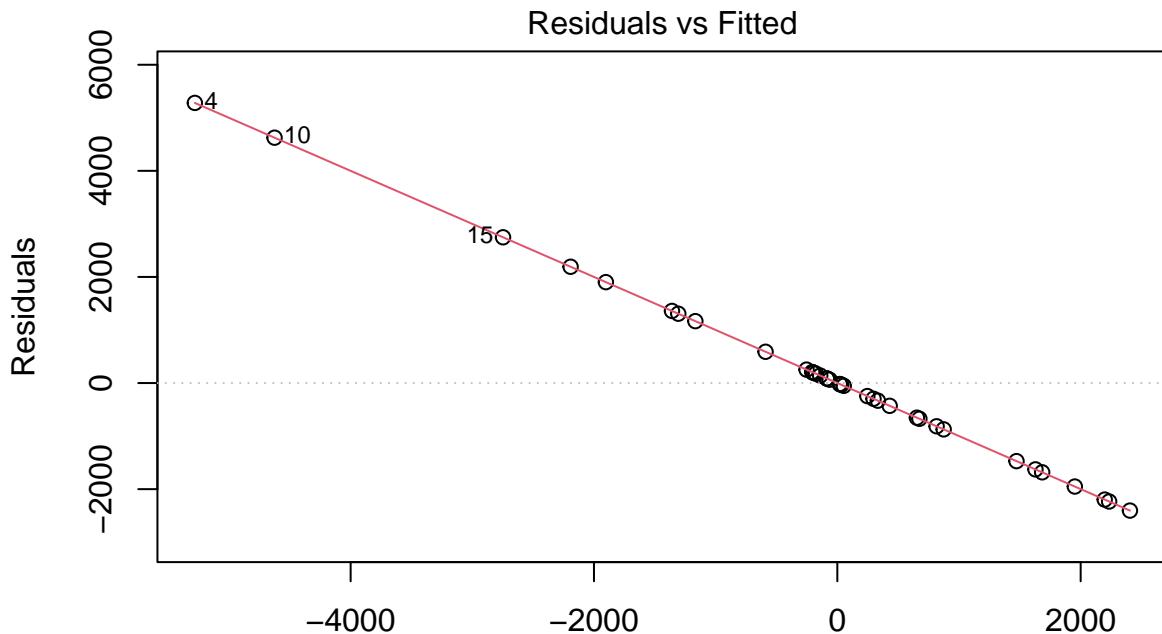
```

## NULL
confint <- confint(coef.model, parm = 2:11)
pip <- summary(bas.model)
PIP[c(1:12), 10] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

```

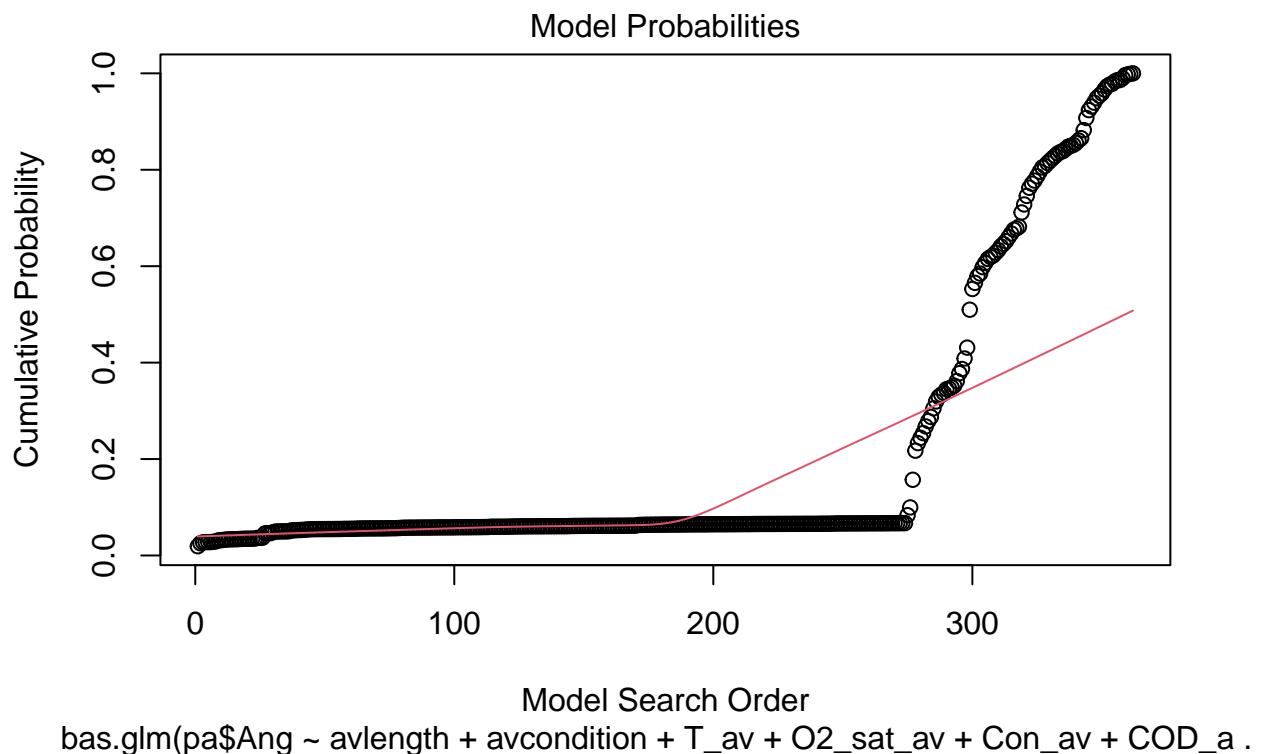
## Variation in Anguillicoloides infection

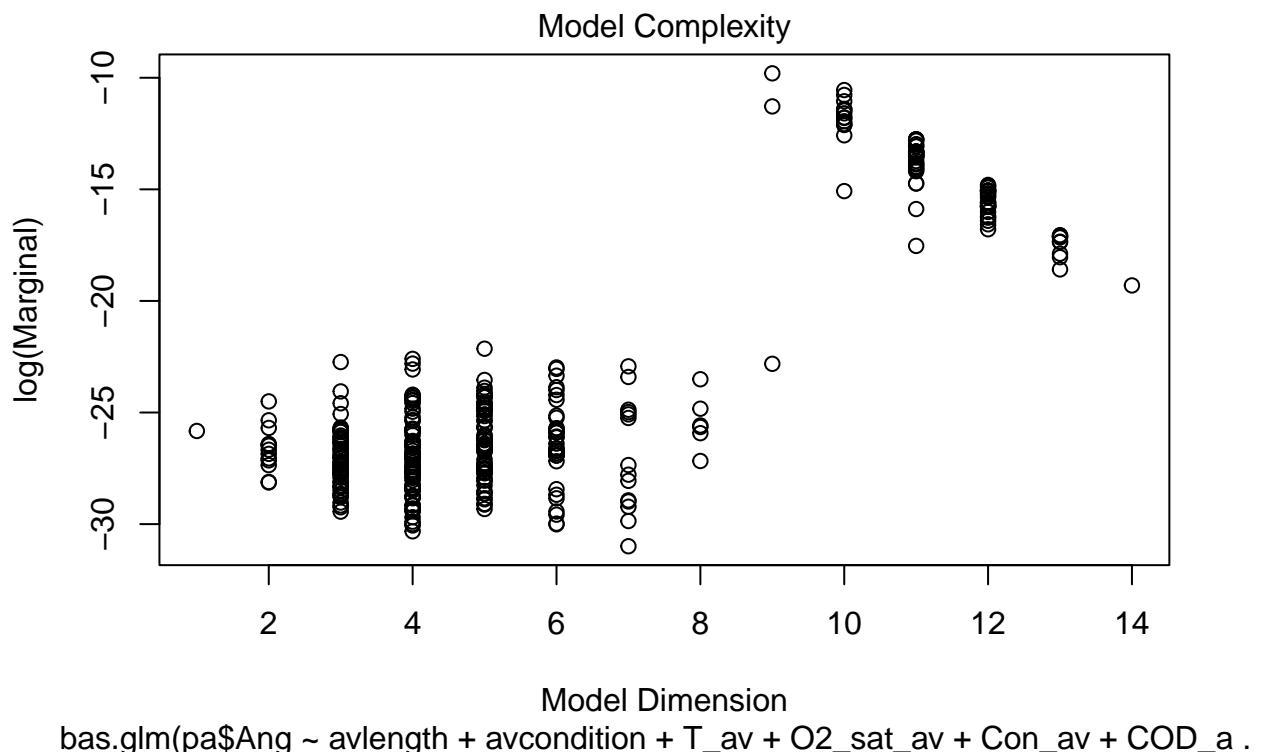
```
bas.model <- bas.glm(pa$Ang ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4._av + Nt_av + SM_av + pool_riffle + meander + spavar$netcen + spavar$updist,
  data = environment2, betaprior = g.prior(100), family = binomial)
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```

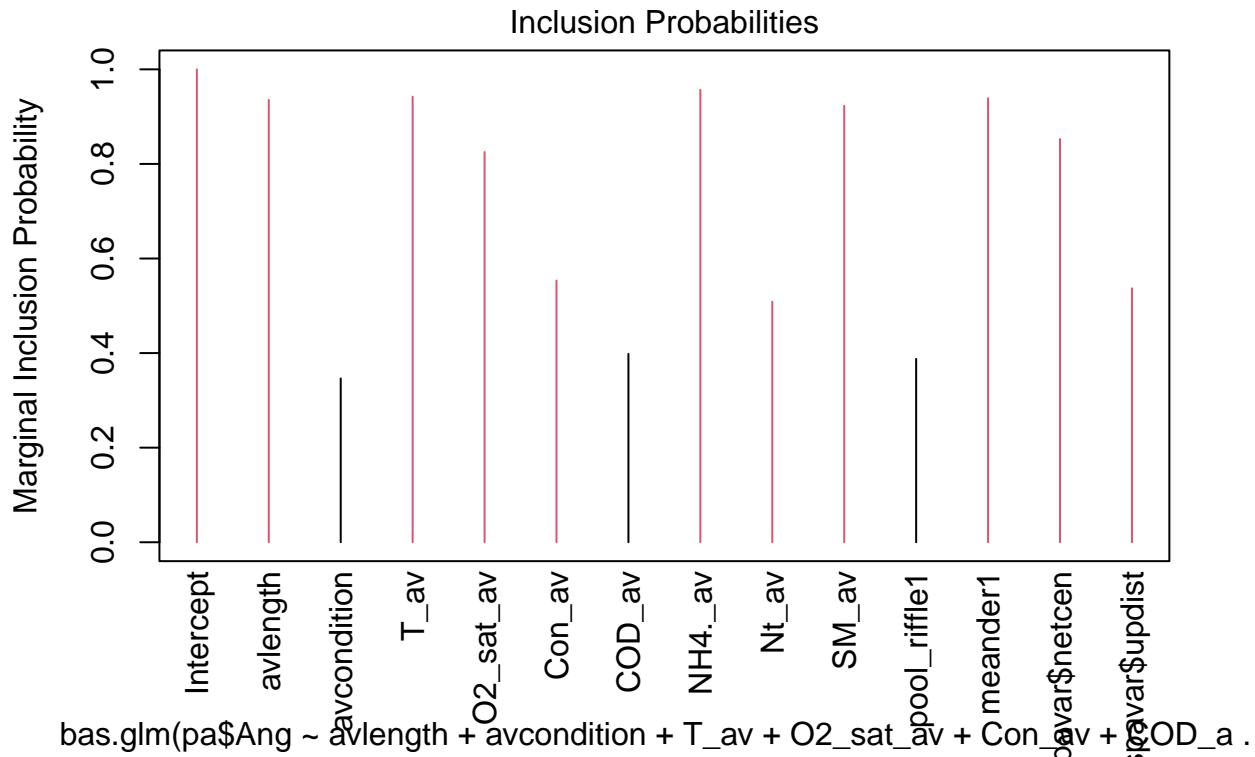


Predictions under BMA

```
bas.glm(pa$Ang ~ avlength + avcondition + T_av + O2_sat_av + Con_av + COD_a .
```







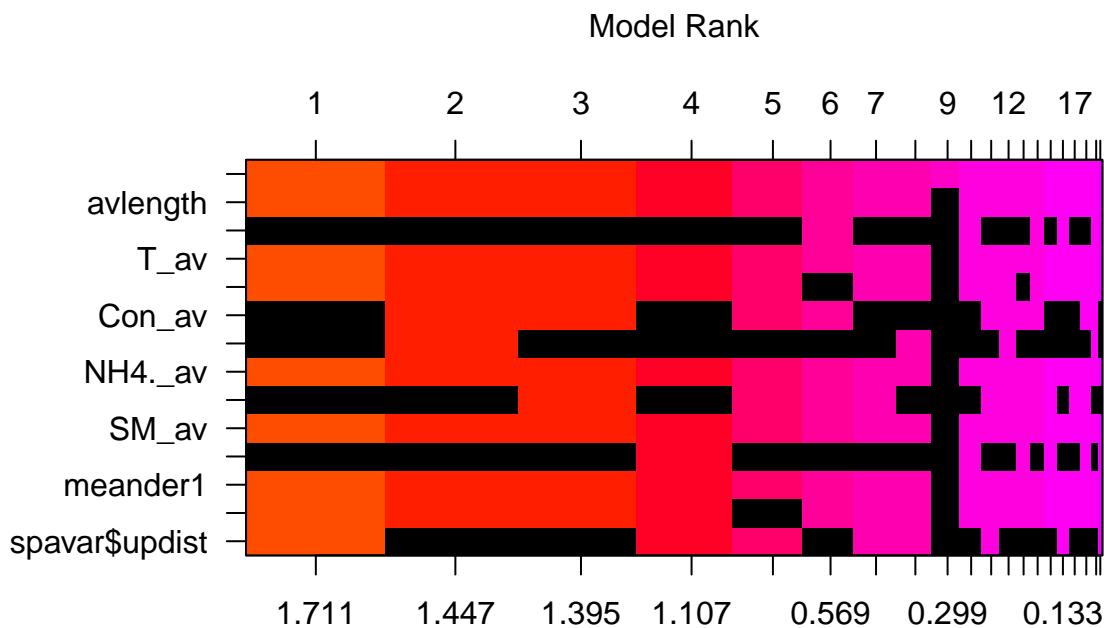
```
summary(bas.model)
```

	P(B != 0   Y)	model 1	model 2	model 3	model 4
## Intercept	1.0000000	1.000000	1.0000000	1.0000000	1.0000000
## avlength	0.9355103	1.000000	1.0000000	1.0000000	1.0000000
## avcondition	0.3463745	0.000000	0.0000000	0.0000000	0.0000000
## T_av	0.9421875	1.000000	1.0000000	1.0000000	1.0000000
## O2_sat_av	0.8250732	1.000000	1.0000000	1.0000000	1.0000000
## Con_av	0.5533691	0.000000	1.0000000	1.0000000	0.0000000
## COD_av	0.3983643	0.000000	1.0000000	0.0000000	0.0000000
## NH4._av	0.9568481	1.000000	1.0000000	1.0000000	1.0000000
## Nt_av	0.5085571	0.000000	0.0000000	1.0000000	0.0000000
## SM_av	0.9229126	1.000000	1.0000000	1.0000000	1.0000000
## pool_riffle1	0.3878662	0.000000	0.0000000	0.0000000	1.0000000
## meander1	0.9389893	1.000000	1.0000000	1.0000000	1.0000000
## spavar\$netcen	0.8527466	1.000000	1.0000000	1.0000000	1.0000000
## spavar\$updist	0.5369995	1.000000	0.0000000	0.0000000	1.0000000
## BF	NA	1.000000	0.3794633	0.4728289	0.2858158
## PostProbs	NA	0.078600	0.0604000	0.0573000	0.0430000
## R2	NA	1.000000	1.0000000	1.0000000	1.0000000
## dim	NA	9.000000	10.0000000	10.0000000	10.0000000
## logmarg	NA	-9.802499	-10.7714968	-10.5515211	-11.0549070
##		model 5			
## Intercept	1.0000000				
## avlength	1.0000000				
## avcondition	0.0000000				

```

## T_av           1.0000000
## O2_sat_av    1.0000000
## Con_av        1.0000000
## COD_av        0.0000000
## NH4._av       1.0000000
## Nt_av         1.0000000
## SM_av         1.0000000
## pool_riffle1 0.0000000
## meander1      1.0000000
## spavar$netcen 0.0000000
## spavar$updist 1.0000000
## BF            0.1967739
## PostProbs     0.0296000
## R2            1.0000000
## dim           10.0000000
## logmarg       -11.4281992
image(bas.model, rotate = F)

```



```

coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

```

```

## [1] FALSE FALSE
## [13] FALSE FALSE
confint(coef.model)

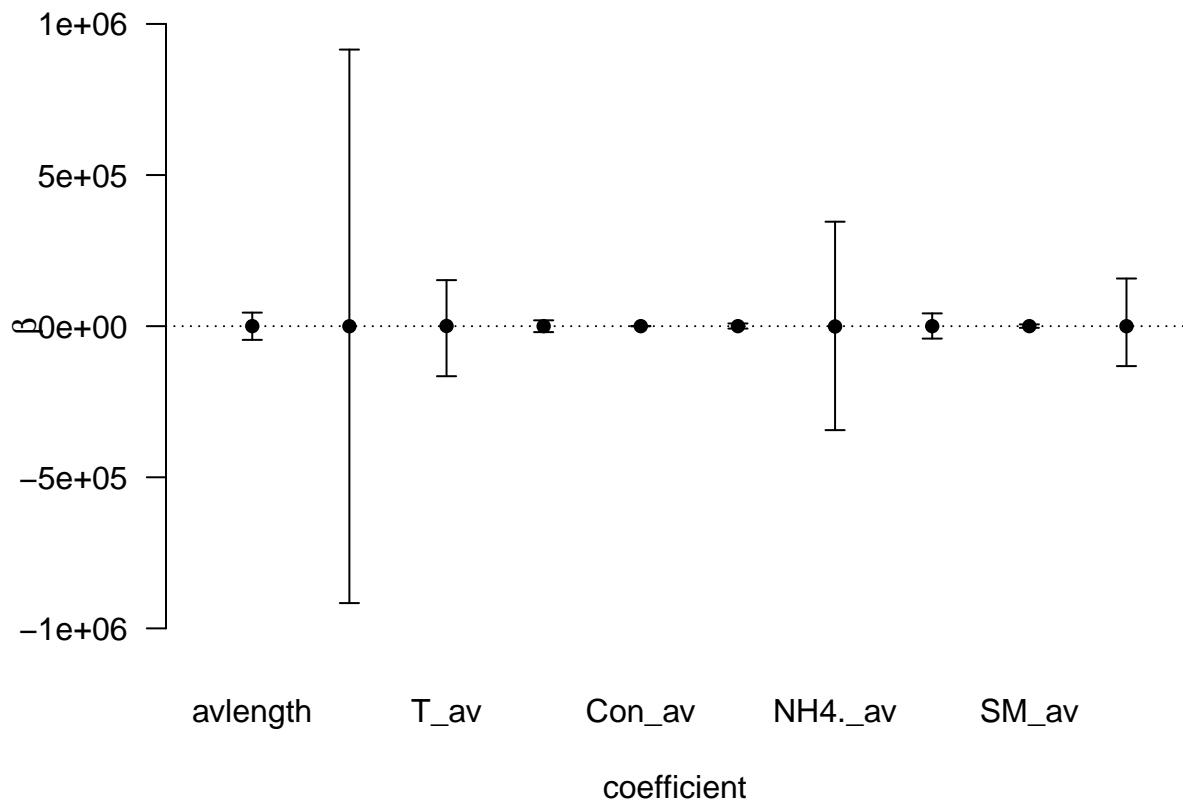
```

##	2.5%	97.5%	beta
----	------	-------	------

```

## Intercept      -2.342397e+06 2.292359e+06 -6.144254e+03
## avlength     -4.167656e+04 4.302794e+04  1.269685e+02
## avcondition   -8.433288e+05 9.198846e+05 -3.946828e+02
## T_av          -1.555082e+05 1.503038e+05  5.703285e+02
## O2_sat_av    -1.820844e+04 1.753037e+04 -3.692908e+01
## Con_av        -6.154323e+02 6.277199e+02  3.734592e-01
## COD_av        -7.639271e+03 8.760342e+03  1.836151e+00
## NH4._av       -3.489163e+05 3.232031e+05 -1.028673e+03
## Nt_av          -4.111878e+04 4.070122e+04 -4.220598e+01
## SM_av          -5.325797e+03 5.862174e+03  1.257268e+01
## pool_riffle1  -1.414276e+05 1.278708e+05 -1.339754e+02
## meander1       -4.025889e+05 3.904431e+05 -1.332238e+03
## spavar$netcen -1.466684e+01 1.243781e+01 -2.616840e-02
## spavar$updist -2.899881e+00 3.168840e+00  1.211959e-03
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"
plot(confint(coef.model, parm = 2:11))

```



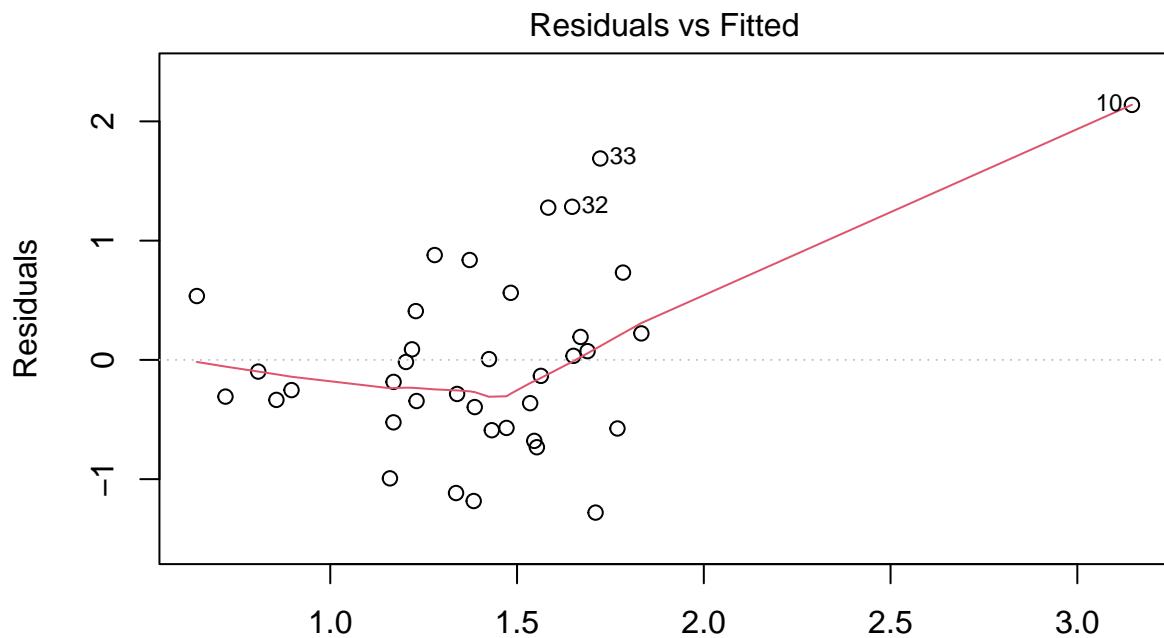
```

## NULL
confint <- confint(coef.model, parm = 2:11)
pip <- summary(bas.model)
PIP[c(1:12), 11] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

```

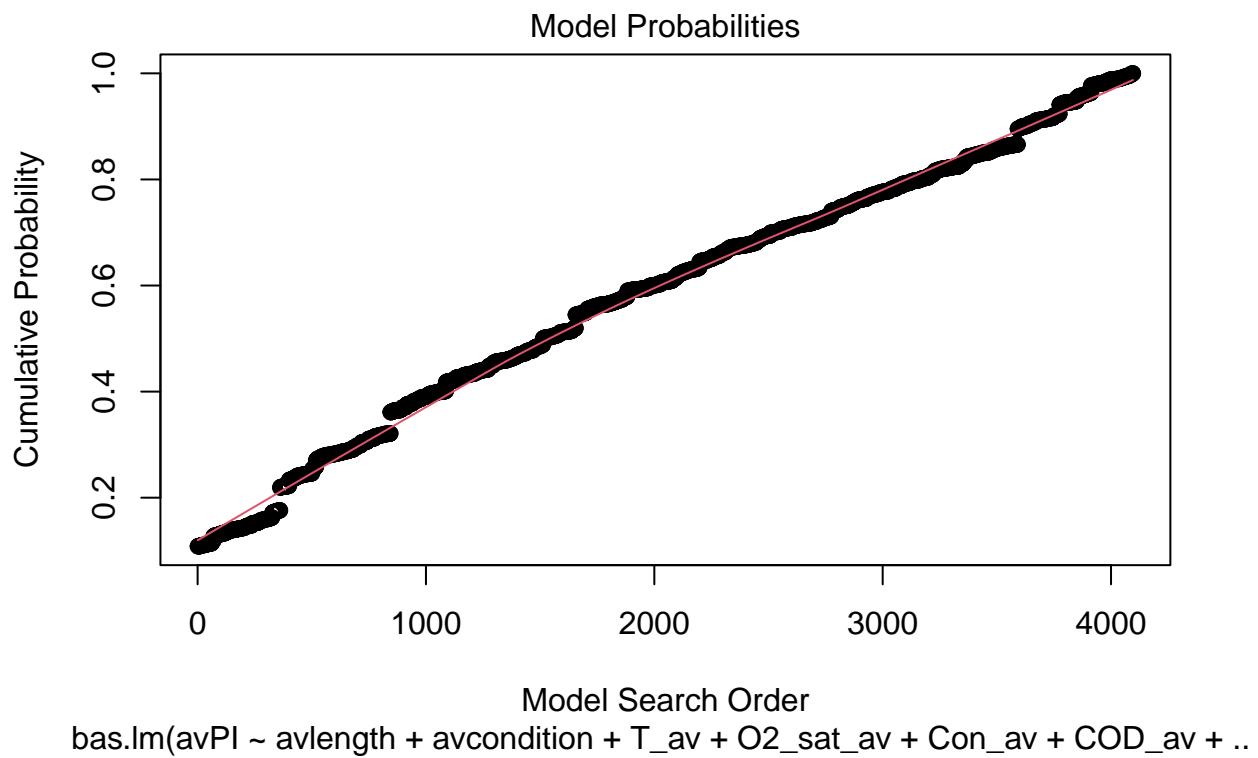
## Variation in Individual Parasitization Index (all parasites)

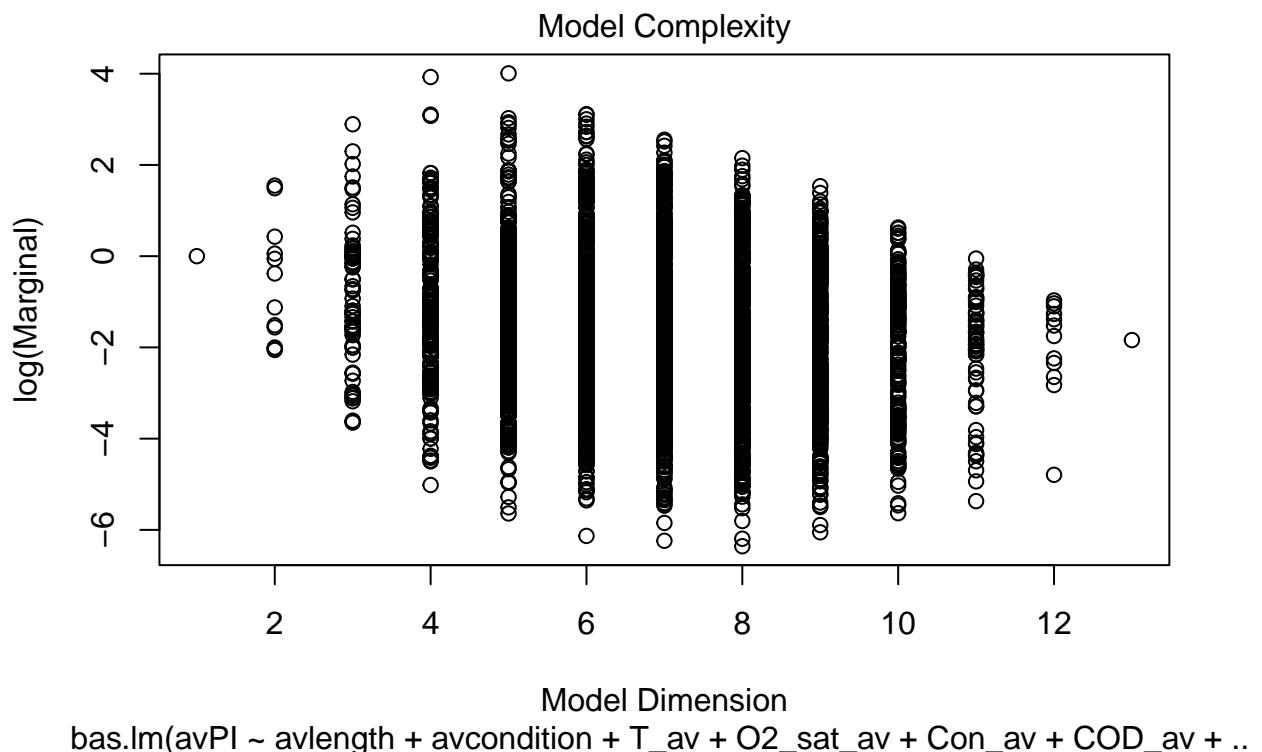
```
bas.model <- bas.lm(avPI ~ avlength + avcondition + T_av + O2_sat_av + Con_av + COD_av +
NH4._av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```

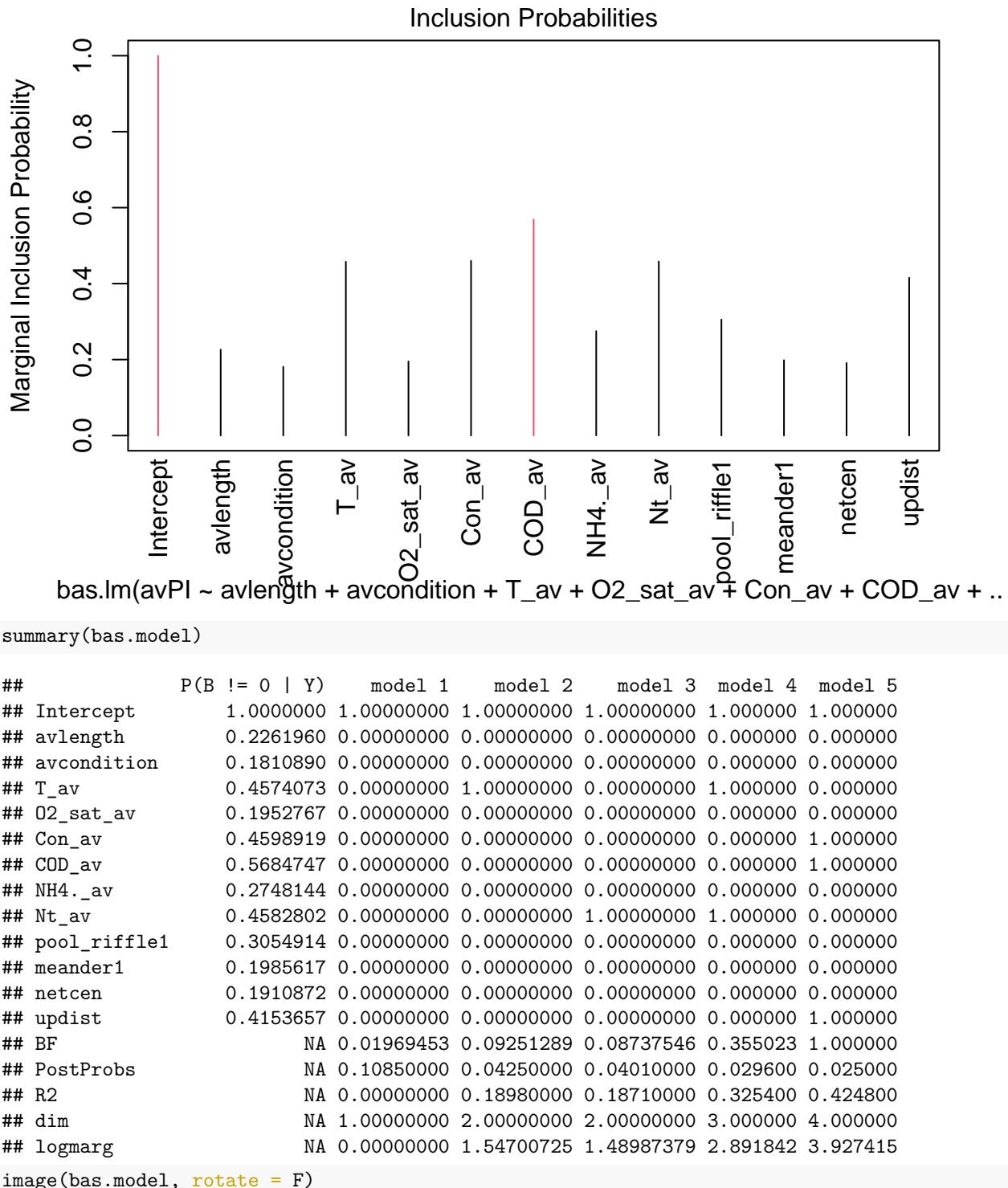


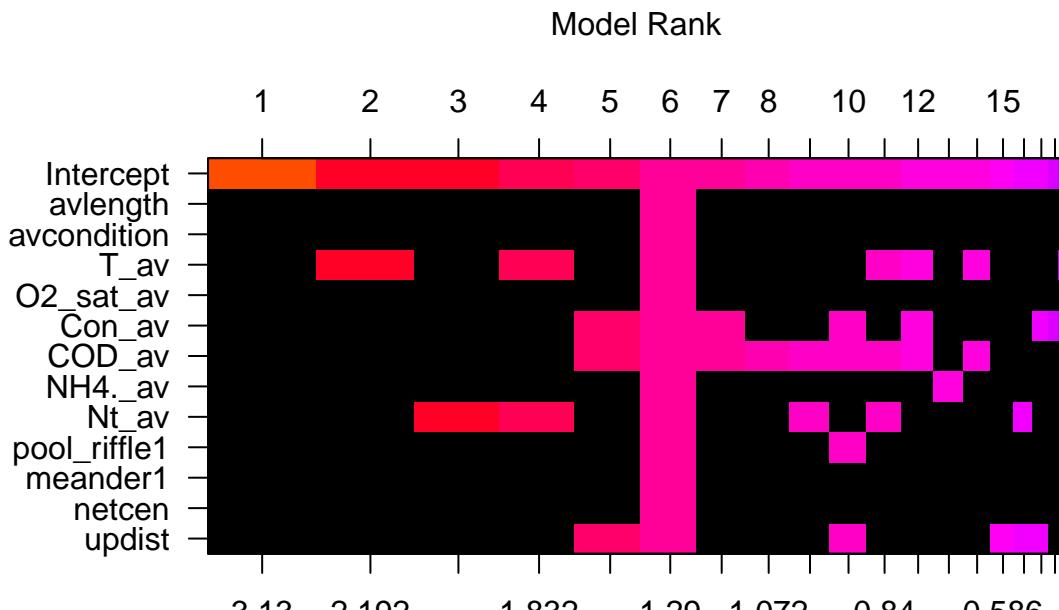
Predictions under BMA

```
bas.lm(avPI ~ avlength + avcondition + T_av + O2_sat_av + Con_av + COD_av + ..
```









### Log Posterior Odds

```

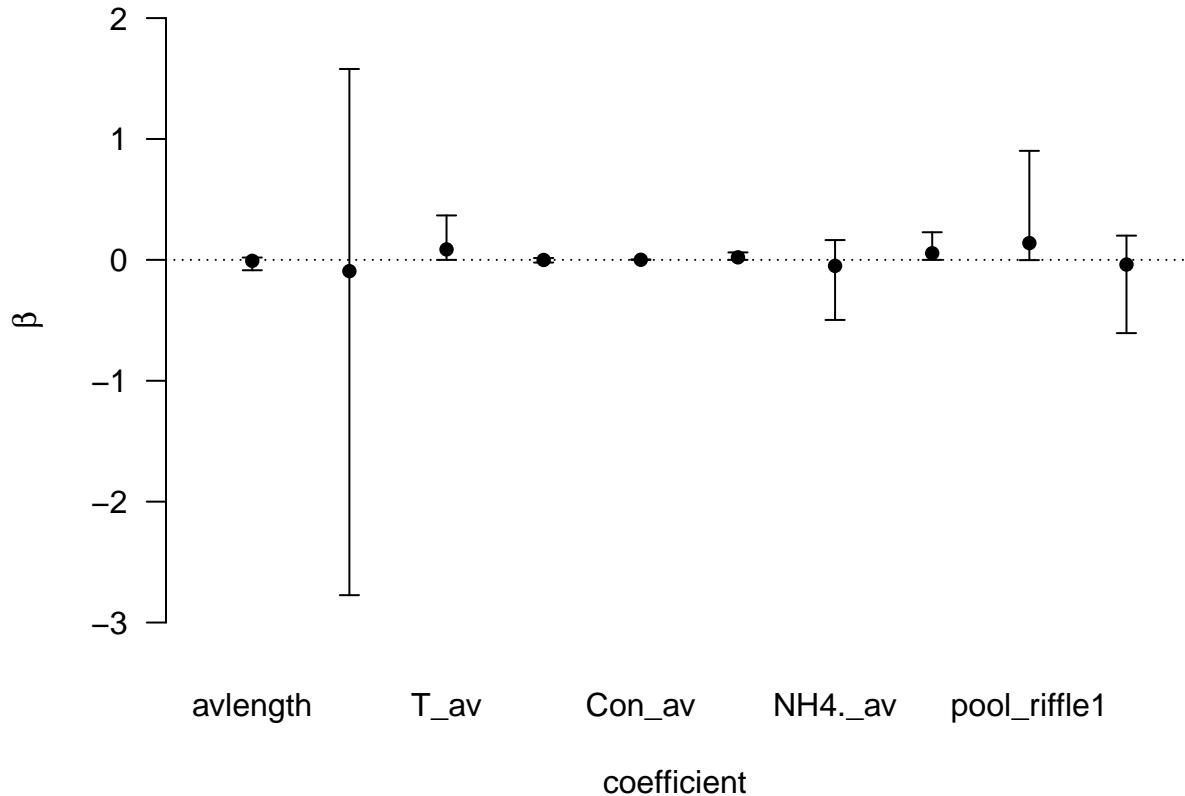
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept     1.120291e+00 1.707073e+00 1.422090e+00
## avlength      -8.824325e-02 1.649504e-02 -7.992245e-03
## avcondition   -2.672622e+00 1.898960e+00 -9.325811e-02
## T_av          -9.669331e-04 3.654261e-01 8.644408e-02
## O2_sat_av     -2.183472e-02 1.857077e-02 -7.476618e-04
## Con_av         -6.528939e-07 3.247900e-03 8.405554e-04
## COD_av         0.000000e+00 6.200938e-02 2.109188e-02
## NH4_av         -5.215179e-01 1.749201e-01 -4.965441e-02
## Nt_av          0.000000e+00 2.291633e-01 5.507750e-02
## pool_riffle1  -2.031055e-03 9.306171e-01 1.389140e-01
## meander1       -6.309762e-01 1.961508e-01 -3.866036e-02
## netcen          -2.867936e-05 1.995745e-05 -1.270941e-06
## updist          -2.698295e-05 0.000000e+00 -5.952452e-06
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"

```

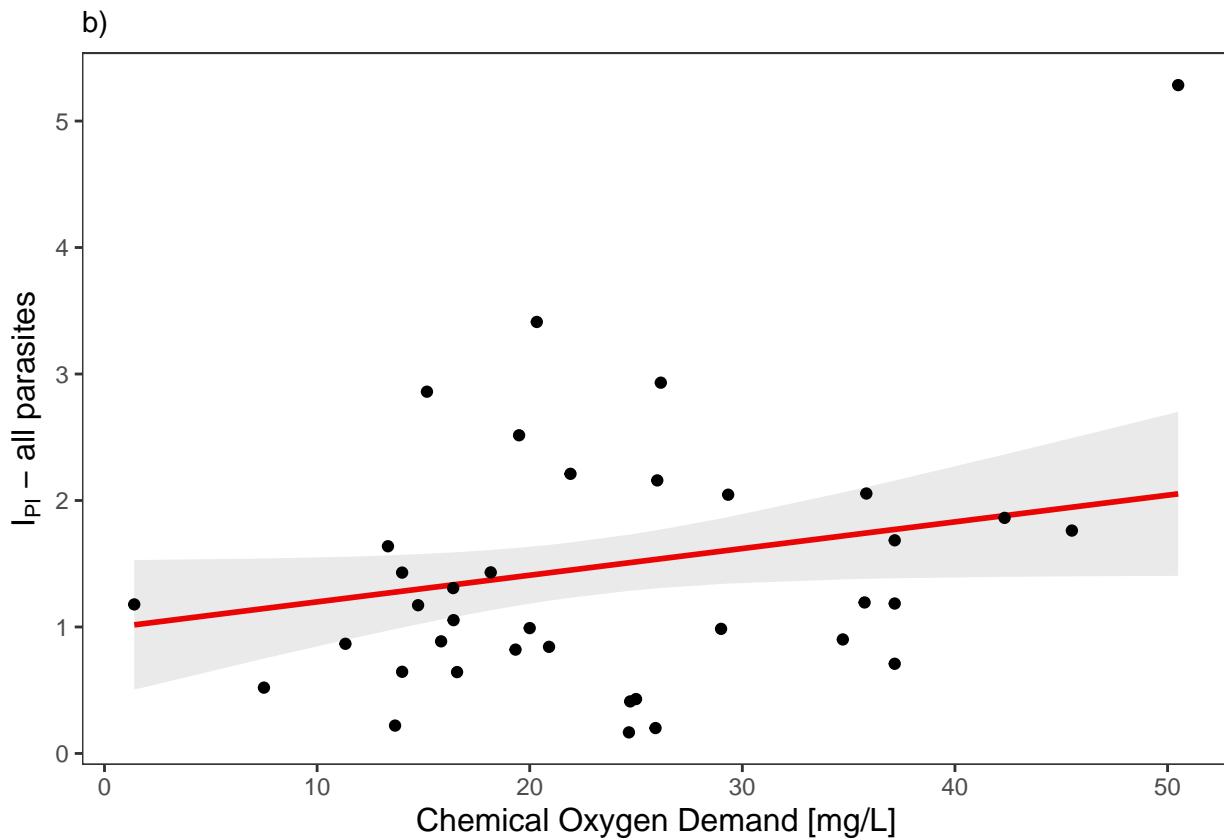
```
plot(confint(coef.model, parm = 2:11))
```



```
## NULL
confint <- confint(coef.model, parm = 2:11)
pip <- summary(bas.model)
PIP[c(1:12), 12] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

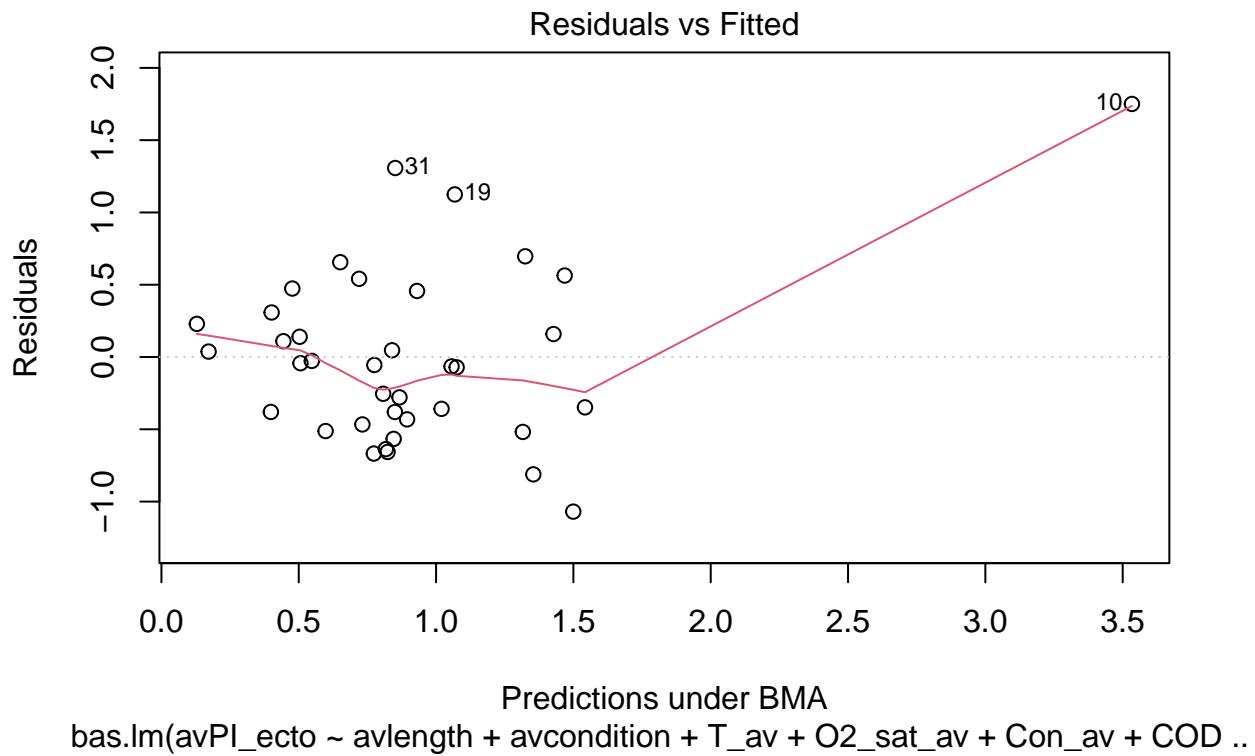
## Prediction plot for marginal effect of COD on Individual Parasitization Index

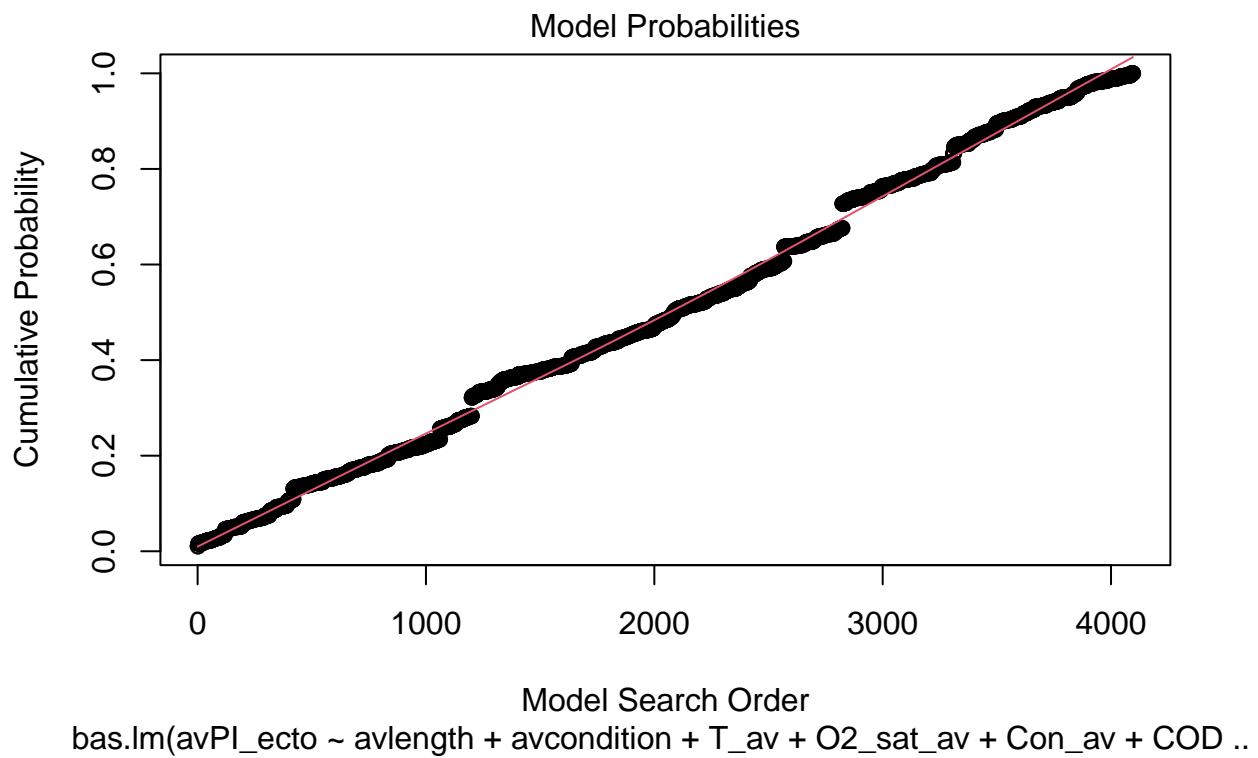
```
newdata1 <- newdata
newdata1[, "COD_av"] <- environment2$COD_av
BMA_IPI_COD_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3b = ggplot(environment2, aes(COD_av, BMA_IPI_COD_av$fit)) + theme_bw() + geom_line(color = "red",
size = 1) + geom_ribbon(aes(ymin = (BMA_IPI_COD_av$fit - BMA_IPI_COD_av$se.bma.fit),
ymax = (BMA_IPI_COD_av$fit + BMA_IPI_COD_av$se.bma.fit)), alpha = 0.1) + geom_point(data = environment2,
aes(x = COD_av, y = avPI)) + labs(x = expression("Chemical Oxygen Demand [mg/L]"),
y = expression("I"["I"] * " - all parasites")) + theme(axis.title.x = element_text(size = 12),
axis.title.y = element_text(size = 12)) + theme(panel.grid.major = element_blank(),
panel.grid.minor = element_blank()) + labs(subtitle = "b")
figure3b
```

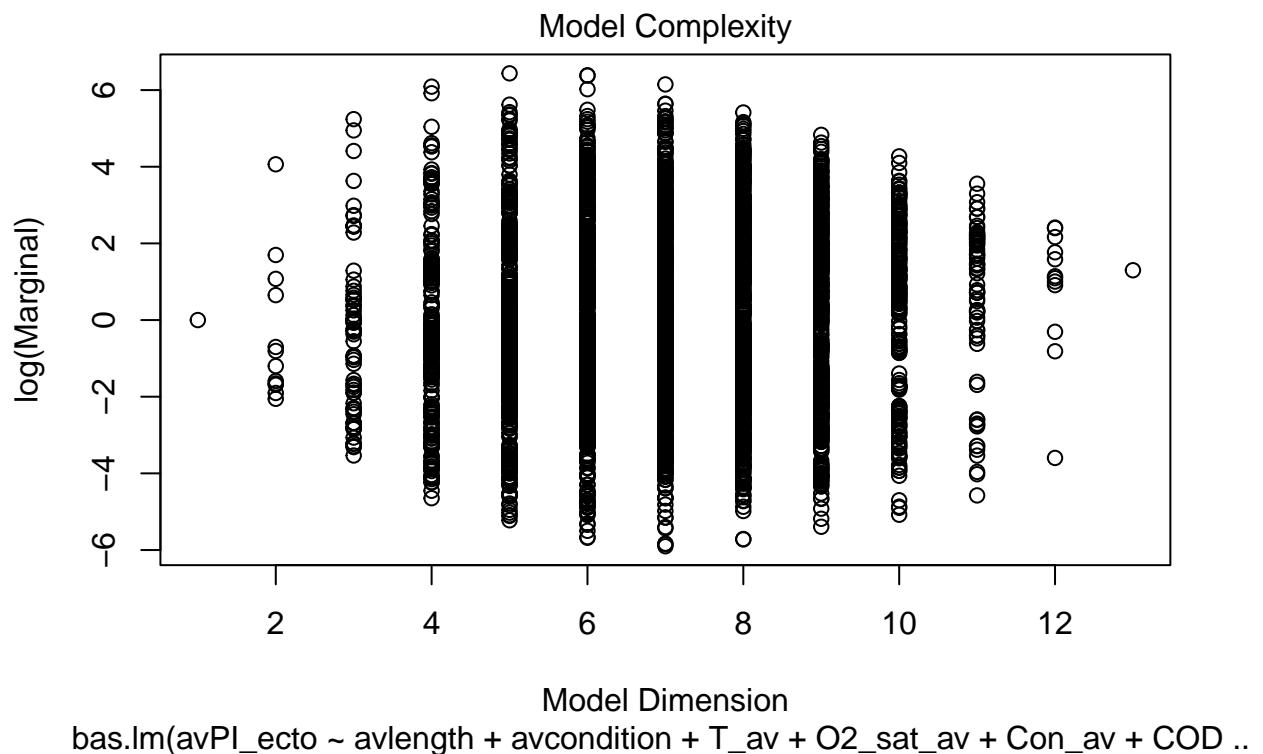


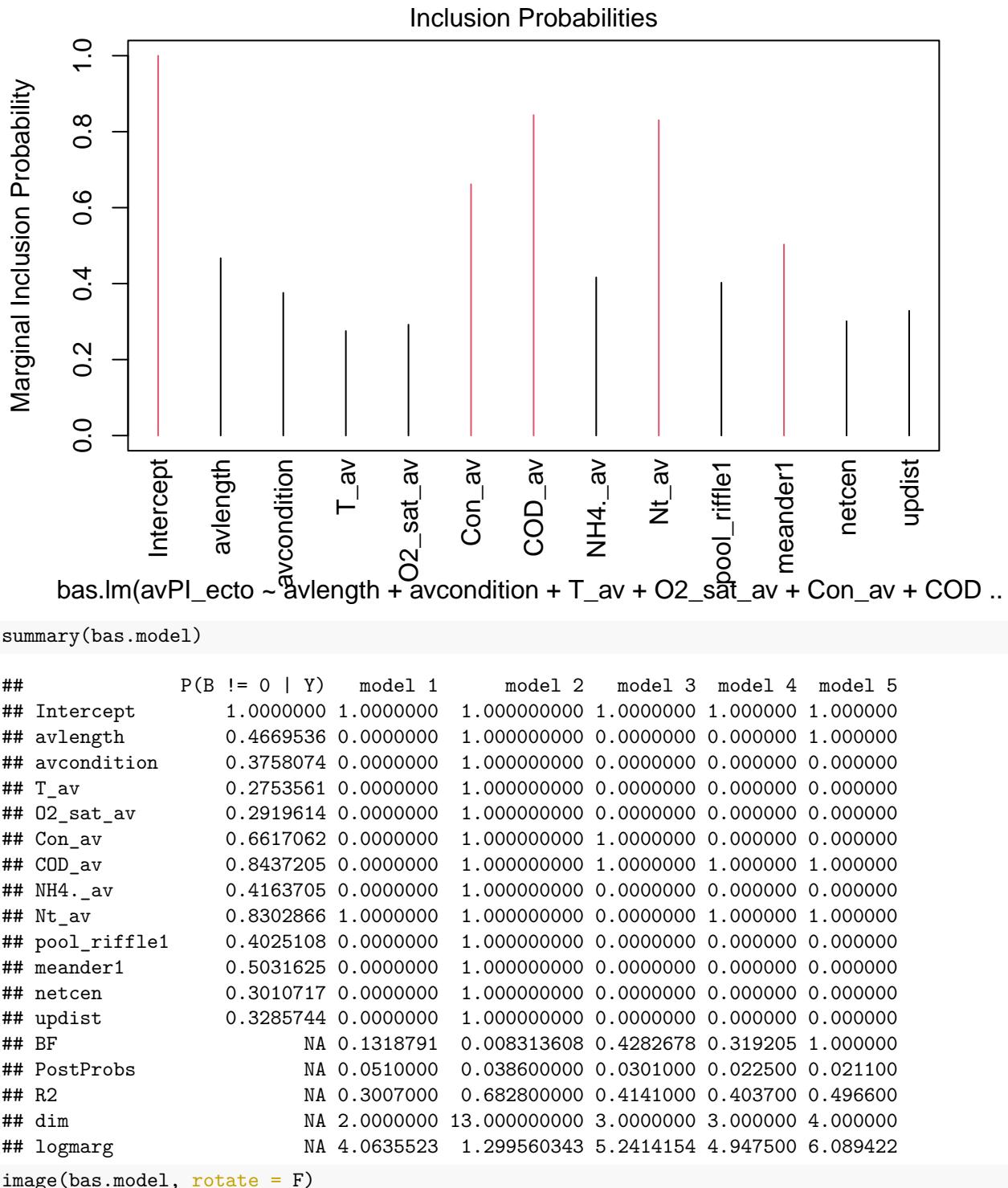
#### Variation in Individual Parasitization Index (only ectoparasites)

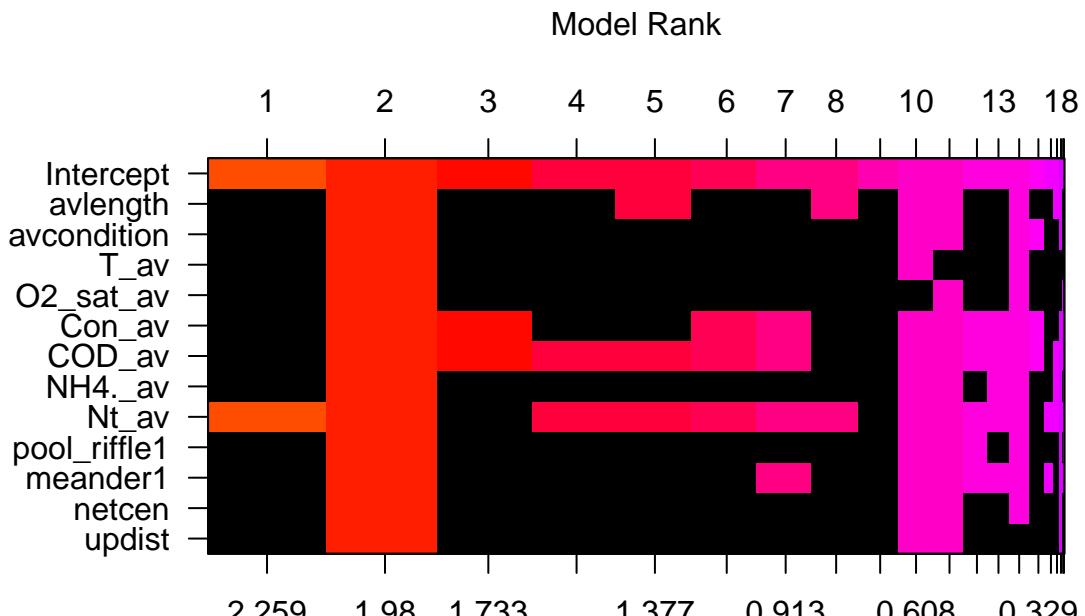
```
bas.model <- bas.lm(avPI_ecto ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4_av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```











### Log Posterior Odds

```

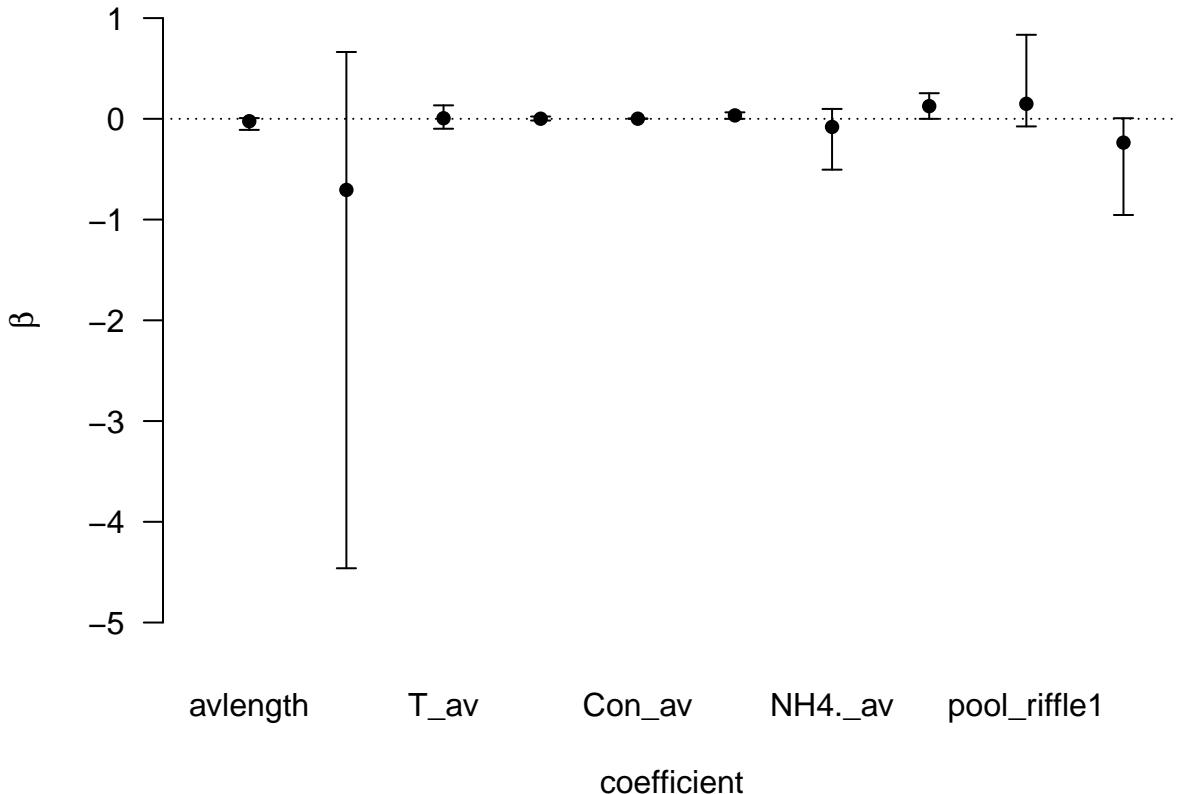
coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE
## [13] FALSE
confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept     6.844050e-01 1.163925e+00 9.199577e-01
## avlength     -1.089617e-01 1.058491e-02 -2.357999e-02
## avcondition  -4.485616e+00 5.683599e-01 -7.055549e-01
## T_av         -9.980029e-02 1.386524e-01 5.818763e-03
## O2_sat_av   -1.574072e-02 2.350604e-02 1.122707e-03
## Con_av        0.000000e+00 3.271708e-03 1.203349e-03
## COD_av        0.000000e+00 6.416318e-02 3.405818e-02
## NH4._av      -4.907699e-01 1.115362e-01 -8.034086e-02
## Nt_av         0.000000e+00 2.543941e-01 1.257830e-01
## pool_riffle1 -7.083547e-02 8.292214e-01 1.490513e-01
## meander1     -9.479577e-01 3.471591e-02 -2.358742e-01
## netcen        -3.463624e-05 2.019553e-05 -2.447510e-06
## updist        -1.521838e-05 4.554671e-06 -1.836166e-06
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

```

```
plot(confint(coef.model, parm = 2:11))
```

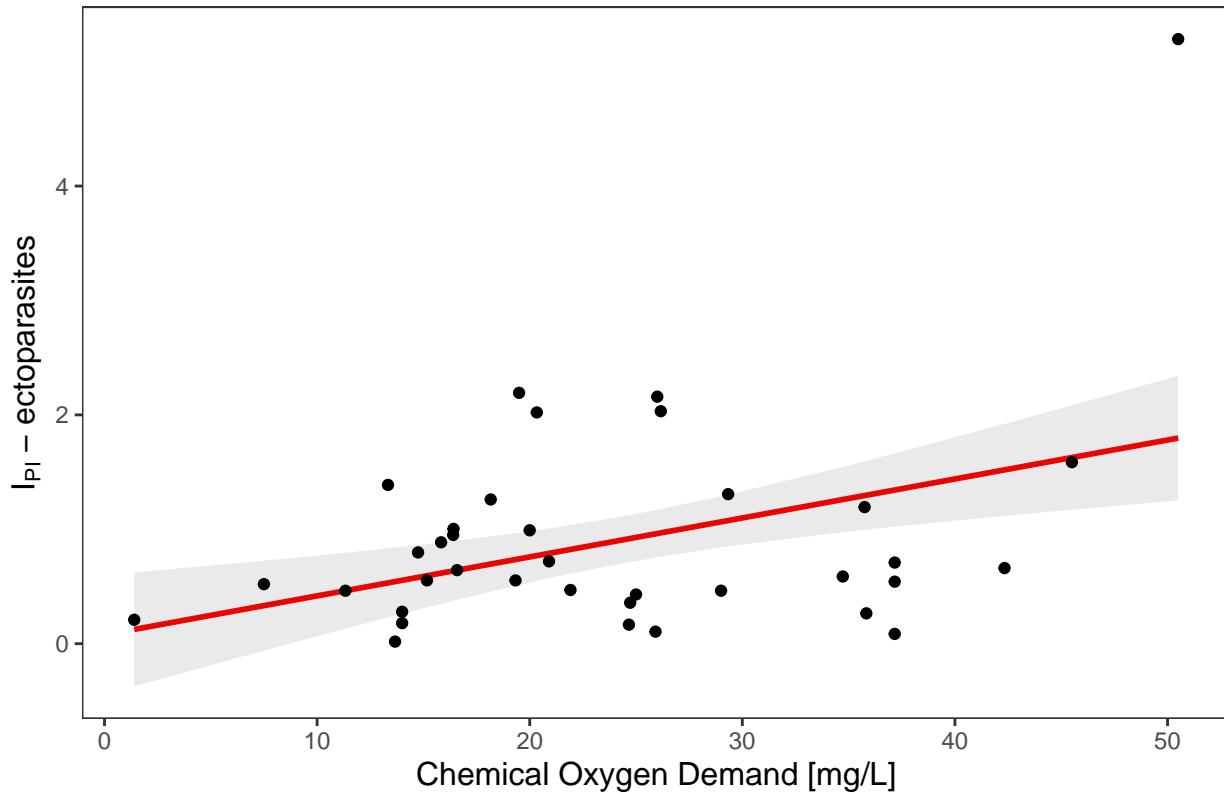


```
## NULL  
confint <- confint(coef.model, parm = 2:11)  
pip <- summary(bas.model)  
PIP[c(1:12), 13] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])
```

## Prediction plot for marginal effect of COD on Individual Parasitation Index (only ectoparasites)

```
newdata1 <- newdata  
newdata1[, "COD_av"] <- environment2$COD_av  
BMA_IPIecto_COD_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)  
figure3c = ggplot(environment2, aes(COD_av, BMA_IPIecto_COD_av$fit)) + theme_bw() +  
  geom_line(color = "red", size = 1) + geom_ribbon(aes(ymin = (BMA_IPIecto_COD_av$fit -  
    BMA_IPIecto_COD_av$se.bma.fit), ymax = (BMA_IPIecto_COD_av$fit + BMA_IPIecto_COD_av$se.bma.fit)),  
  alpha = 0.1) + geom_point(data = environment2, aes(x = COD_av, y = avPI_ecto)) +  
  labs(x = expression("Chemical Oxygen Demand [mg/L]"), y = expression("I"["PI"] *  
    " - ectoparasites")) + theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +  
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +  
  labs(subtitle = "c")  
figure3c
```

c)



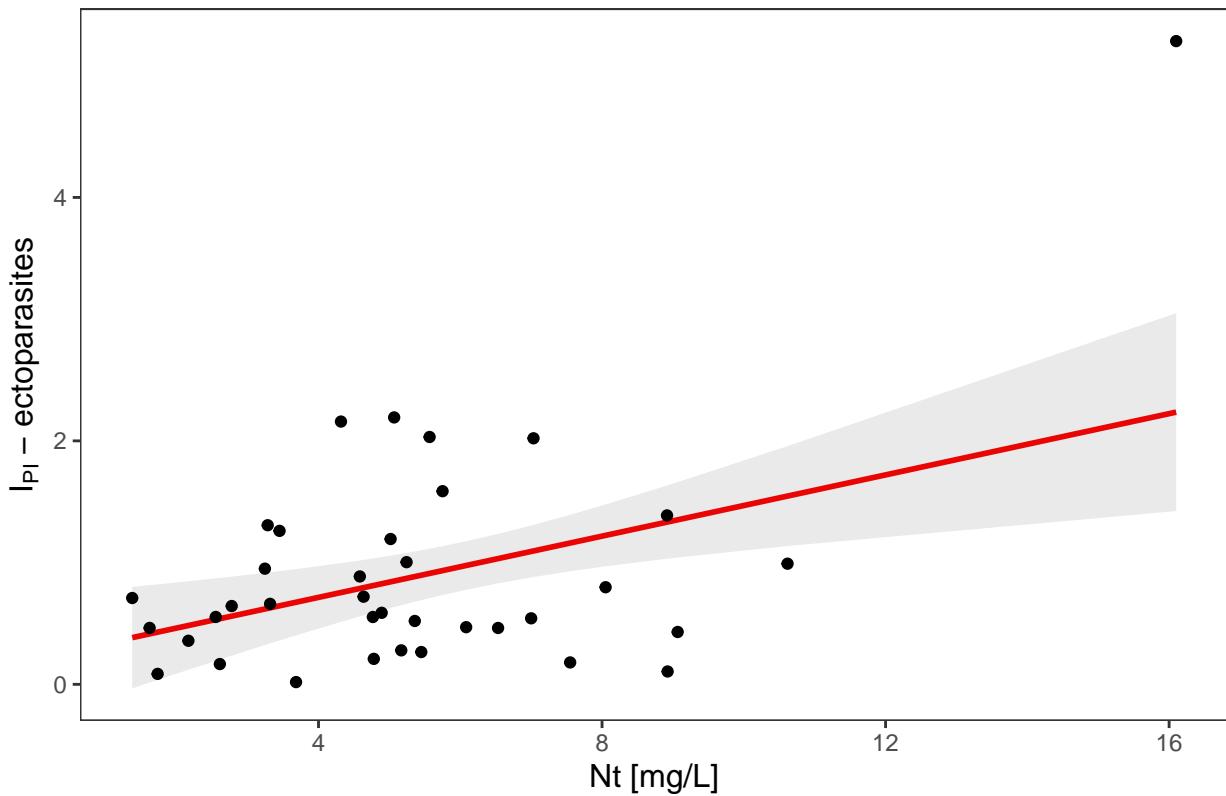
### Prediction plot for marginal effect of total nitrogen on Individual Parasitation Index (only ectoparasites)

```

newdata1 <- newdata
newdata1[, "Nt_av"] <- environment2$Nt_av
BMA_IPIecto_Nt_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3f = ggplot(environment2, aes(Nt_av, BMA_IPIecto_Nt_av$fit)) + theme_bw() +
  geom_line(color = "red", size = 1) + geom_ribbon(aes(ymin = (BMA_IPIecto_Nt_av$fit -
  BMA_IPIecto_Nt_av$se.bma.fit), ymax = (BMA_IPIecto_Nt_av$fit + BMA_IPIecto_Nt_av$se.bma.fit)),
  alpha = 0.1) + geom_point(data = environment2, aes(x = Nt_av, y = avPI_ecto)) +
  labs(x = expression("Nt [mg/L]"), y = expression("I"[PI] * "-" ectoparasites")) +
  theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "f")
figure3f

```

f)

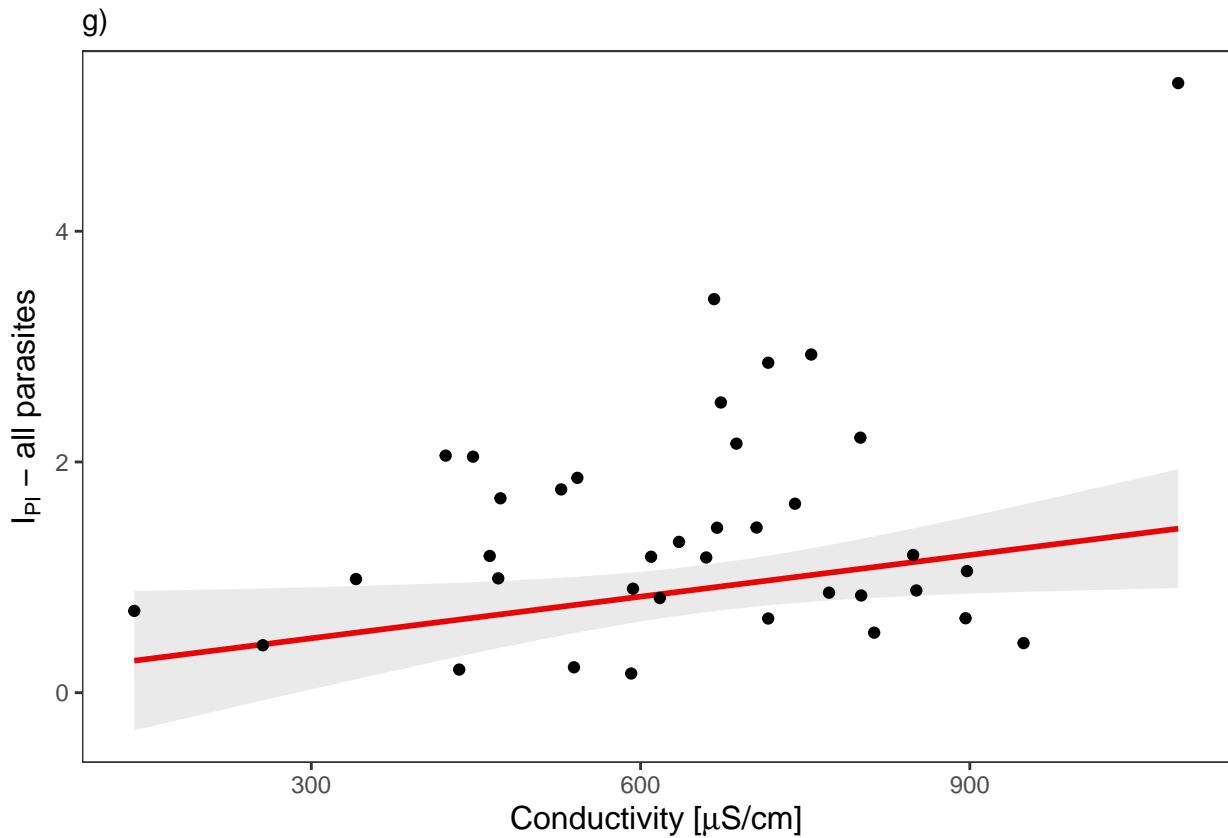


### Prediction plot for marginal effect of conductivity on Individual Parasitation Index (only ectoparasites)

```

newdata1 <- newdata
newdata1[, "Con_av"] <- environment2$Con_av
BMA_IPIecto_Con_av <- predict(bas.model, newdata = newdata1, estimator = "BMA", se.fit = TRUE)
figure3g = ggplot(environment2, aes(Con_av, BMA_IPIecto_Con_av$fit)) + theme_bw() +
  geom_line(color = "red", size = 1) + geom_ribbon(aes(ymin = (BMA_IPIecto_Con_av$fit -
  BMA_IPIecto_Con_av$se.bma.fit), ymax = (BMA_IPIecto_Con_av$fit + BMA_IPIecto_Con_av$se.bma.fit)),
  alpha = 0.1) + geom_point(data = environment2, aes(x = Con_av, y = avPI)) + labs(x = expression(pas-
  mu, "S/cm")), y = expression("I"[PI] * "-" all parasites")) + theme(axis.title.x = element_text(si-
  ze = 12), axis.title.y = element_text(size = 12)) + theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) + labs(subtitle = "g")
figure3g

```

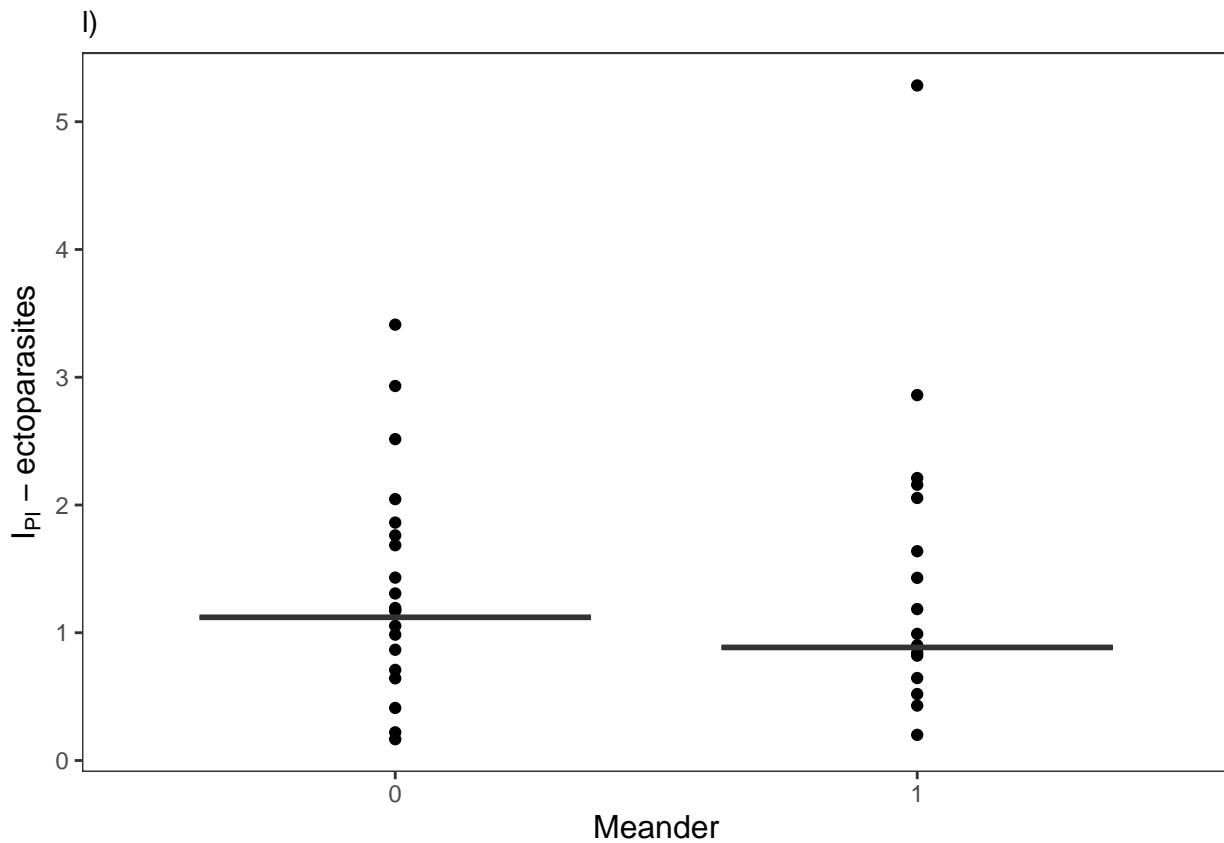


### Prediction plot for marginal effect of meanders on Individual Parasitization Index (only ectoparasites)

```

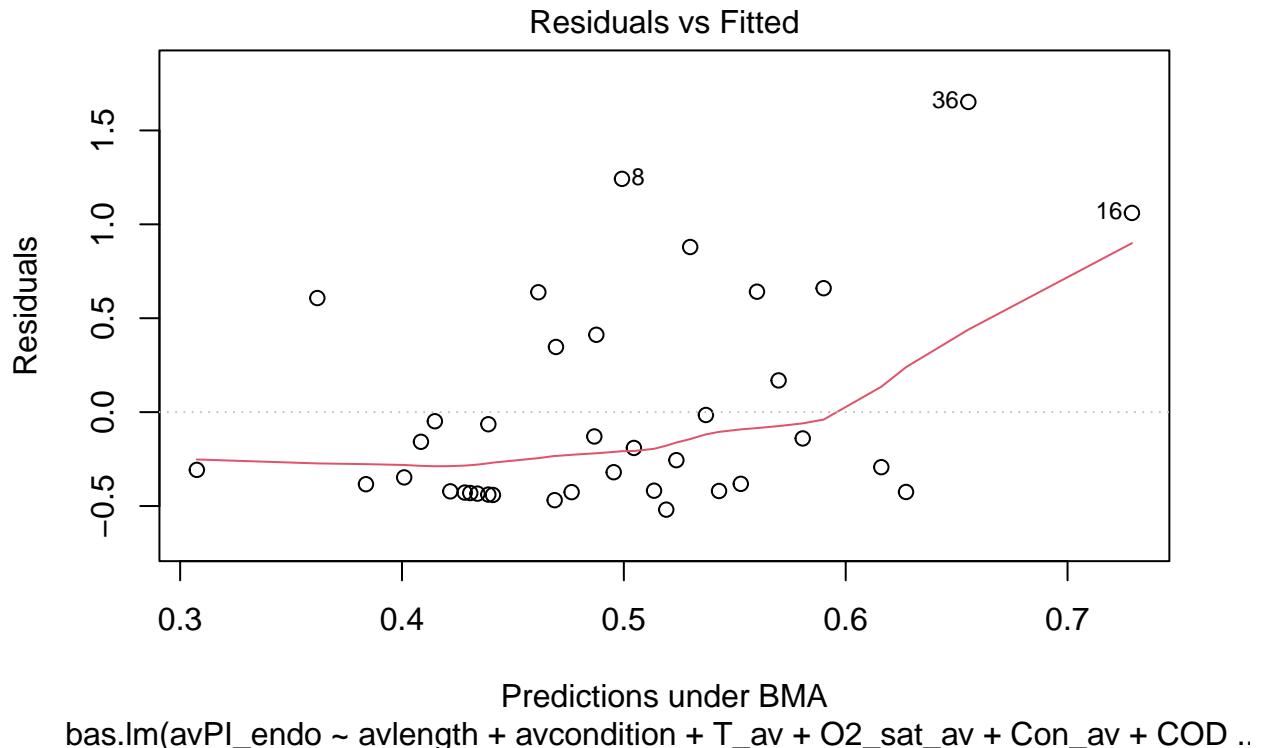
newdata1 <- newdata
newdata1[, "meander"] <- environment2$meander
BMA_IPIecto_meander <- predict(bas.model, newdata = newdata1, estimator = "BMA",
  se.fit = TRUE)
figure31 = ggplot(environment2, aes(meander, BMA_IPIecto_meander$fit)) + theme_bw() +
  # geom_line(color='red', size=1) + geom_ribbon(aes(ymin =
  # (BMA$fit-BMA$se.bma.fit), ymax = (BMA$fit+BMA$se.bma.fit)), alpha = .1) +
  geom_point(data = environment2, aes(x = meander, y = avPI)) + geom_boxplot(aes(lower = (BMA_IPIecto_meander$se.bma.fit),
  middle = BMA_IPIecto_meander$fit, upper = (BMA_IPIecto_meander$fit + BMA_IPIecto_meander$se.bma.fit))) + labs(x = expression("Meander"), y = expression("I"[PI] *
  " - ectoparasites")) + theme(axis.title.x = element_text(size = 12), axis.title.y = element_text(size = 12))
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(subtitle = "1")
figure31

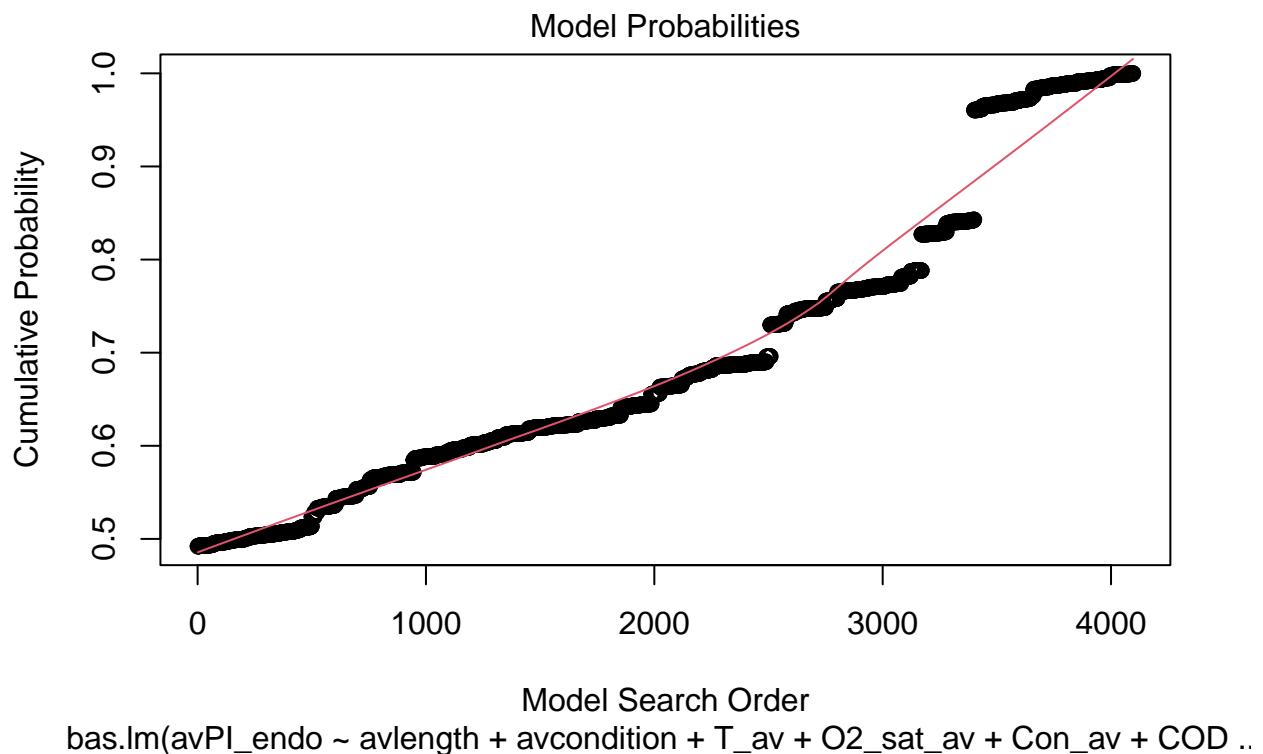
```

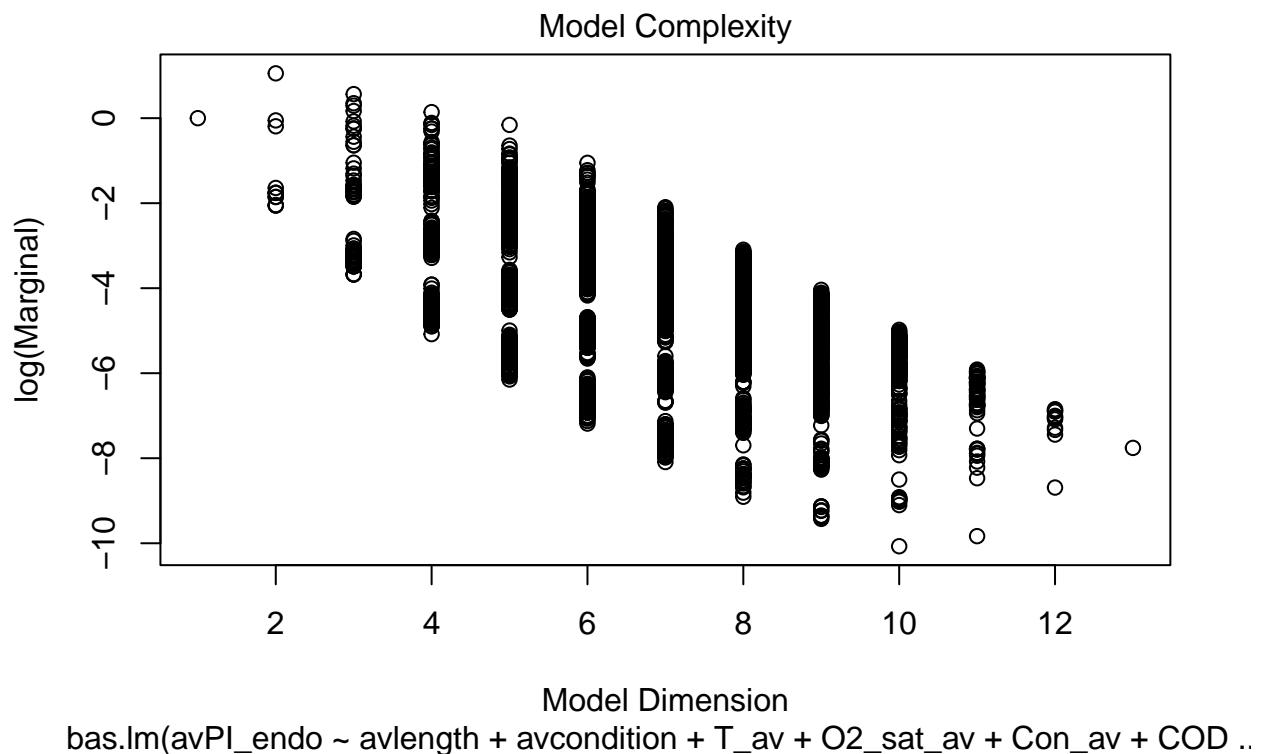


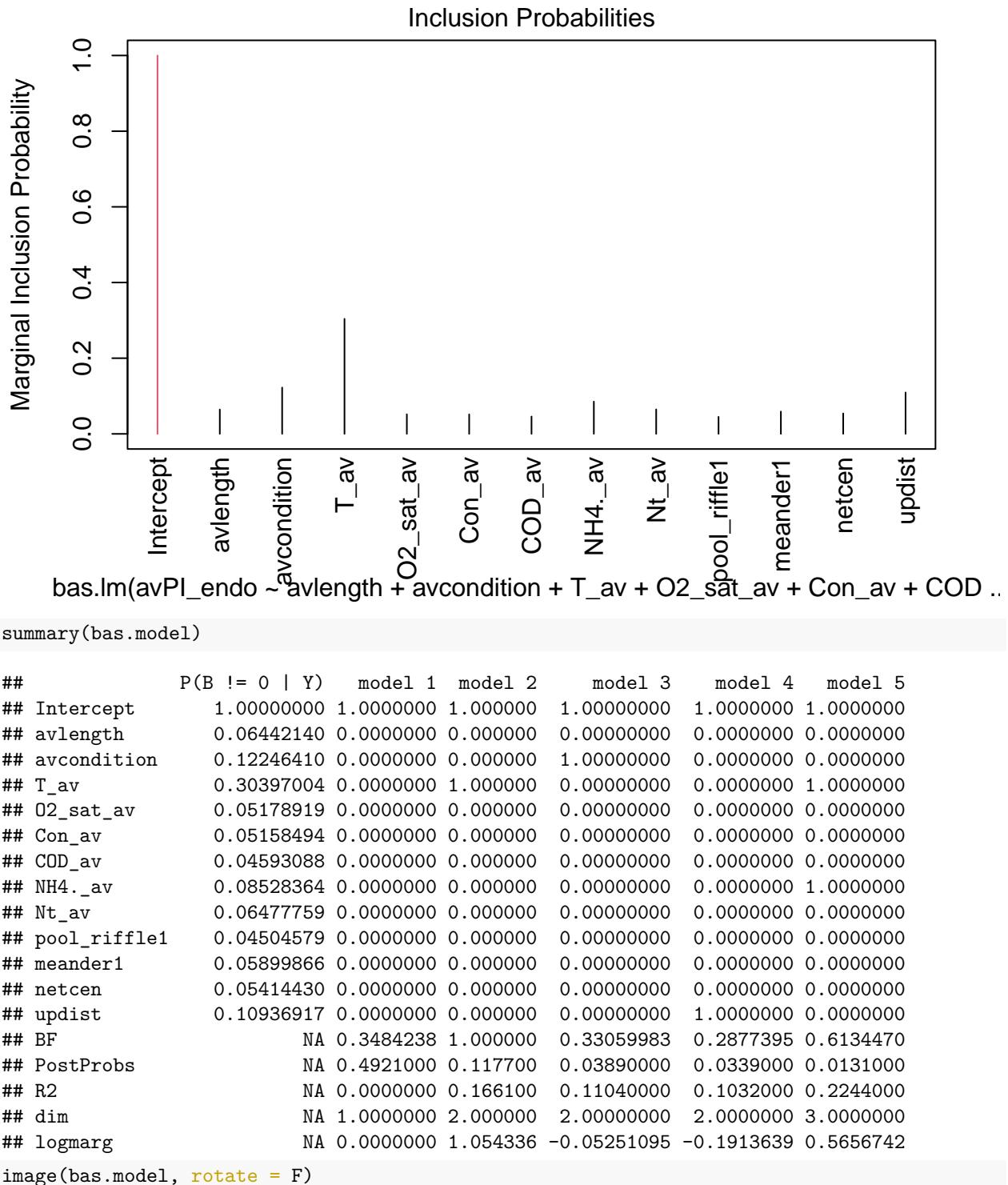
### Variation in Individual Parasitization Index (only endoparasites)

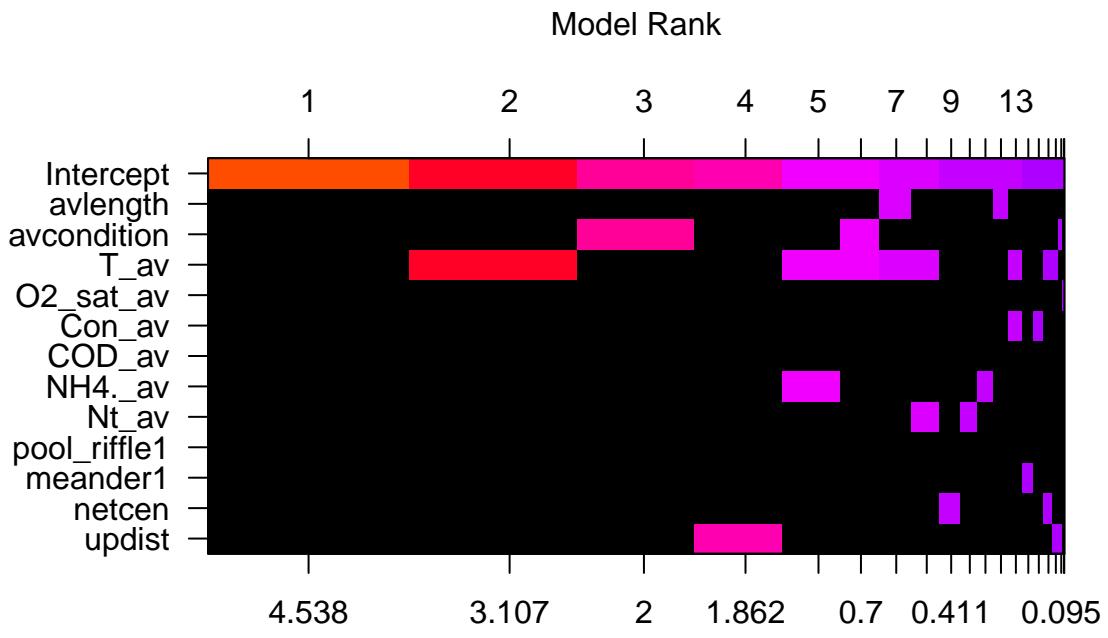
```
bas.model <- bas.lm(avPI_endo ~ avlength + avcondition + T_av + O2_sat_av + Con_av +
  COD_av + NH4_av + Nt_av + pool_riffle + meander + netcen + updist, data = environment2,
  prior = "JZS")
yhat = fitted(bas.model, estimator = "BMA") #these are the fitted values under BMA
r = bas.model$Y - yhat #these are the model residuals
plot(bas.model)
```











```

coef.model <- coef(bas.model)
abs(coef.model$postmean) - 2 * coef.model$postsd > 0

## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE

confint(coef.model)

##                   2.5%      97.5%       beta
## Intercept    2.974427e-01 6.907803e-01 4.947994e-01
## avlength     -1.405216e-05 1.675596e-02 1.398963e-03
## avcondition  0.000000e+00 2.486055e+00 2.446911e-01
## T_av         0.000000e+00 2.138495e-01 4.520999e-02
## O2_sat_av   -5.972427e-04 4.323272e-04 -2.446396e-04
## Con_av       -8.134177e-07 3.917697e-05 -1.385311e-05
## COD_av       0.000000e+00 0.000000e+00 -1.072193e-04
## NH4._av     -1.124588e-01 0.000000e+00 -9.322620e-03
## Nt_av        -2.675627e-02 7.626380e-04 -2.149910e-03
## pool_riffle1 0.000000e+00 0.000000e+00 2.267863e-03
## meander1    -7.249265e-03 7.318603e-02 1.005422e-02
## netcen       -3.093780e-06 3.588756e-08 -3.607730e-07
## updist      -9.482173e-06 0.000000e+00 -8.091451e-07
## attr(),"Probability")
## [1] 0.95
## attr(),"class")
## [1] "confint.bas"

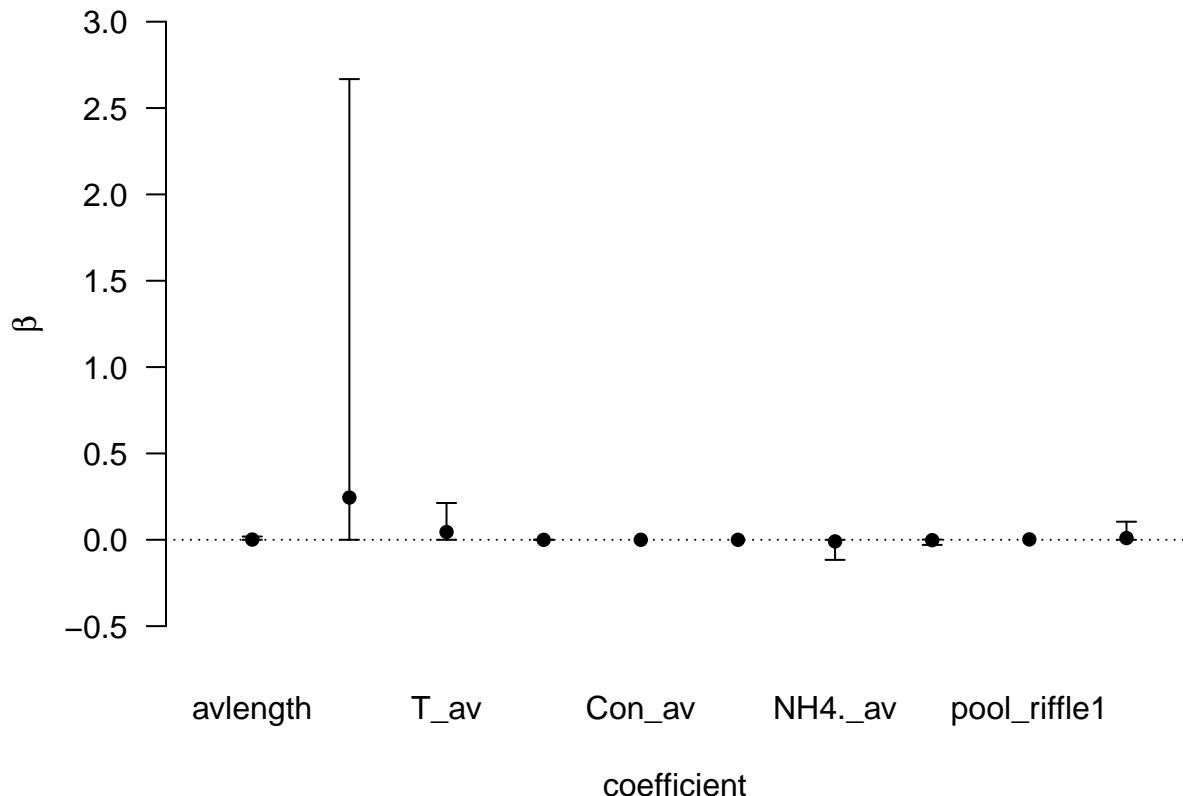
```

```

plot(confint(coef.model, parm = 2:11))

## Warning in arrows(x[not.deg], ci[not.deg, 1], x[not.deg], ci[not.deg, 2], :
## zero-length arrow is of indeterminate angle and so skipped

```



```

## NULL
confint <- confint(coef.model, parm = 2:11)
pip <- summary(bas.model)
PIP[c(1:12), 14] <- pip[2:13, 1] * sign(coef.model$postmean[2:13])

```

**Figure 3**

```

library(gridExtra)
pdf("Figure3.pdf", width = 10.8, height = 14.4)
grid.arrange(figure3a, figure3b, figure3c, figure3d, figure3e, figure3f, figure3g,
            figure3h, figure3i, figure3j, figure3k, figure3l)
dev.off()

## pdf
## 2

```

## 7. BORAL analysis

Model-based analysis of multivariate abundance data using Bayesian Markov chain Monte Carlo methods for parameter estimation

## 7.1 BORAL analysis for average abundances of parasites

```

data$Site <- as.factor(data$site)
levels(data$site) <- levels(as.factor(environment2$Site))
data_m <- merge(data, environment2, by = "Site")
data_all <- na.omit(data_m)
names(data_all)

## [1] "Site"                 "site"                  "fish"
## [4] "parspeciesrichness"   "div_shannon"          "div_simpson"
## [7] "temp"                 "pH"                   "conductivity"
## [10] "nitrogen"             "phosphorus"           "oxygen"
## [13] "netcen.x"              "updist.x"              "updist2"
## [16] "updist3"               "fishspeciesrichness" "weight"
## [19] "weigh..g."              "length"                "SMI"
## [22] "Sex"                  "Gyr"                  "Tri"
## [25] "Glu"                  "ecto_screener"        "Con"
## [28] "CysL"                 "Pro"                  "Aca"
## [31] "Cam"                  "Ang"                  "CysI"
## [34] "endo_screener"         "PI"                   "PI_ecto"
## [37] "PI_endo"              "T_av"                 "O2_sat_av"
## [40] "Con_av"                "COD_av"               "NH4_.av"
## [43] "Nt_av"                 "SM_av"                "pool_riffle"
## [46] "meander"               "updist.y"              "netcen.y"

avcondition <- aggregate(data$SMI, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2]
avlength <- aggregate(data$length, by = list(data[, 1]), function(x) {
  mean(x, na.rm = T)
})[, 2]

y <- round(cbind(avab$Gyr, avab$Tri, avab$Glu, avab$Con, avab$Ang))
X <- cbind(avcondition, avlength, environment2$T_av, environment2$O2_sat_av, environment2$Con_av,
           environment2$COD_av, environment2$NH4_.av, environment2$Nt_av, environment2$netcen,
           environment2$updist, as.numeric(environment2$pool_riffle), as.numeric(environment2$meander))
colnames(X) <- c("avcondition", "avlength", "T", "O2", "Con", "COD", "NH4", "Nt",
                "netcen", "updist", "pool_riffle", "meander")

example_mcmc_control <- list(n.burnin = 1000, n.iteration = 10000, n.thin = 1)
testpath <- file.path(tempdir(), "jagsboralmodel.txt")
paramod <- boral(y, X = X, family = "negative.binomial", mcmc.control = example_mcmc_control,
                 model.name = testpath, lv.control = list(num.lv = 2, type = "independent"), save.model = TRUE)

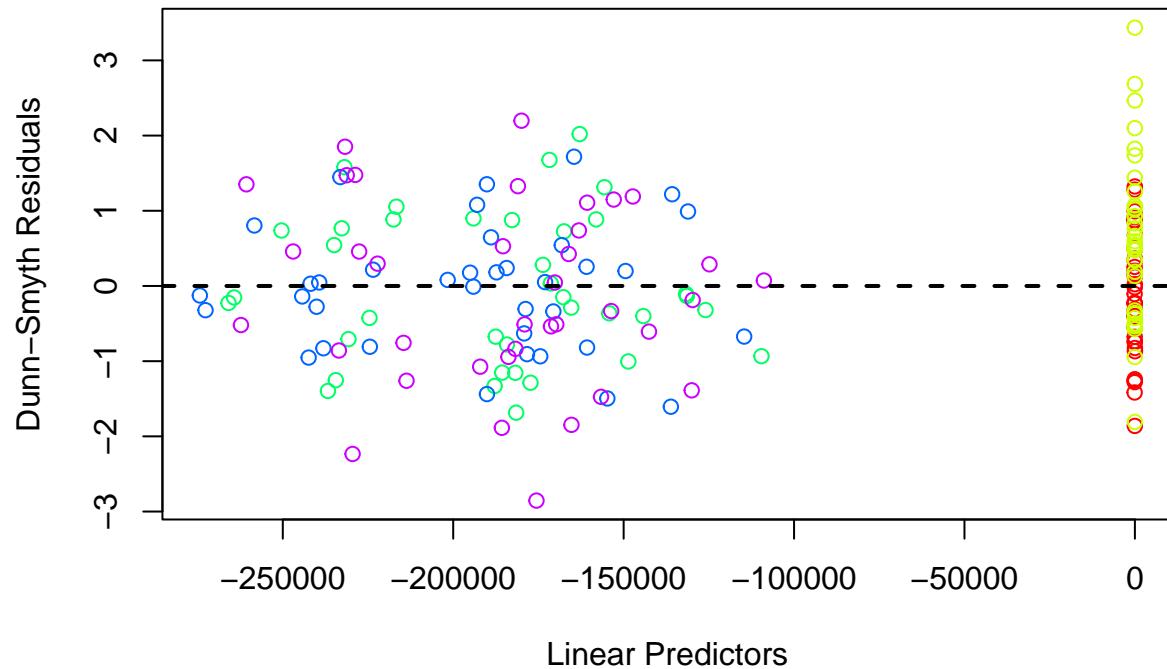
## module glm loaded

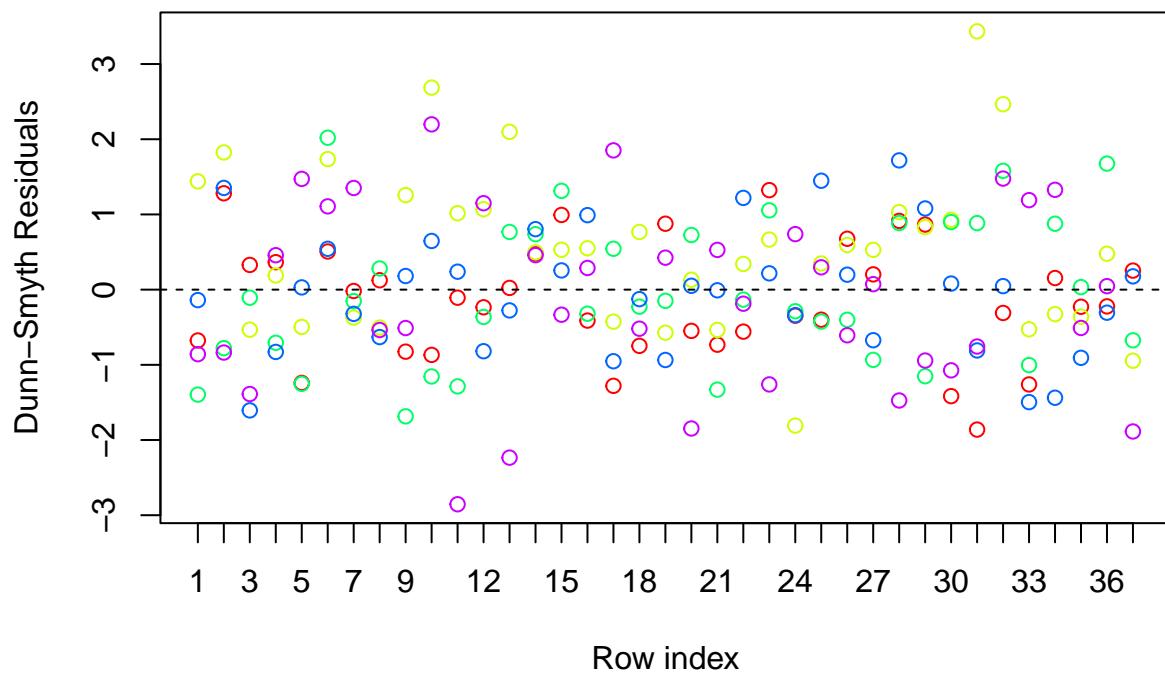
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 185
##   Unobserved stochastic nodes: 338
##   Total graph size: 2173
##
## Initializing model

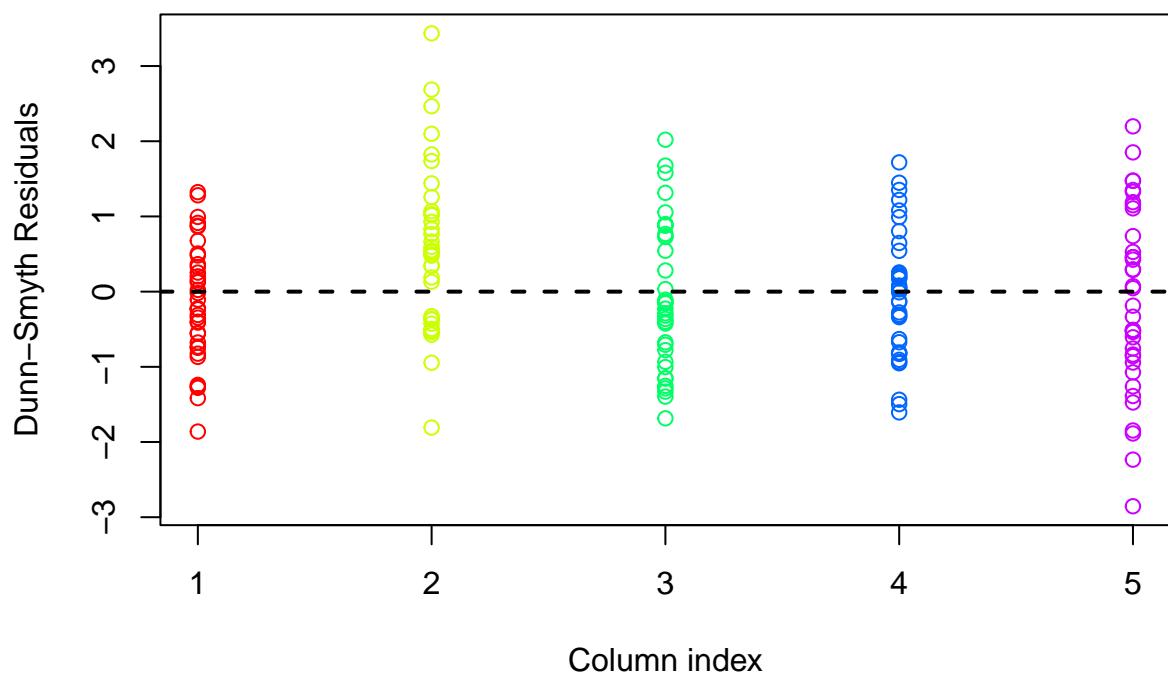
```

```
plot(paramod)
```

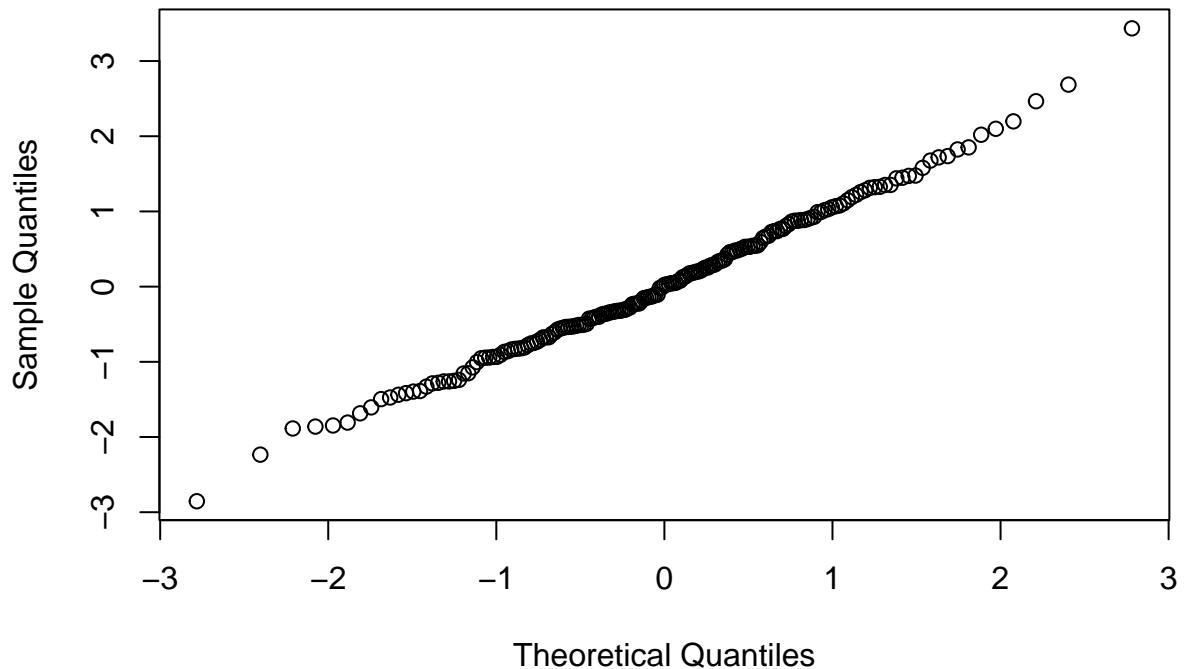
```
## NULL
```





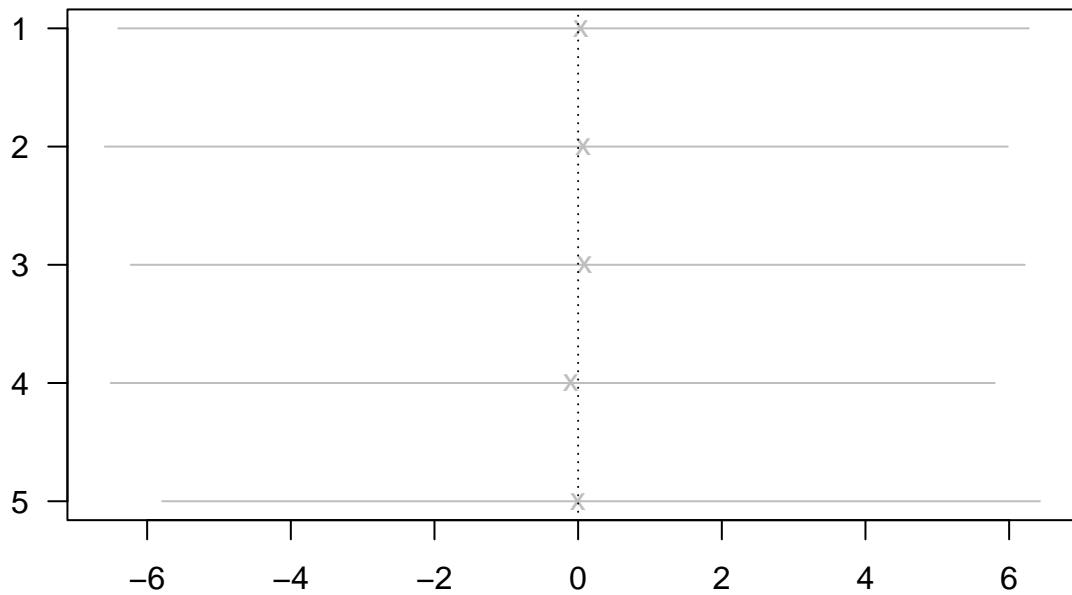


## Quantile–Quantile Plot



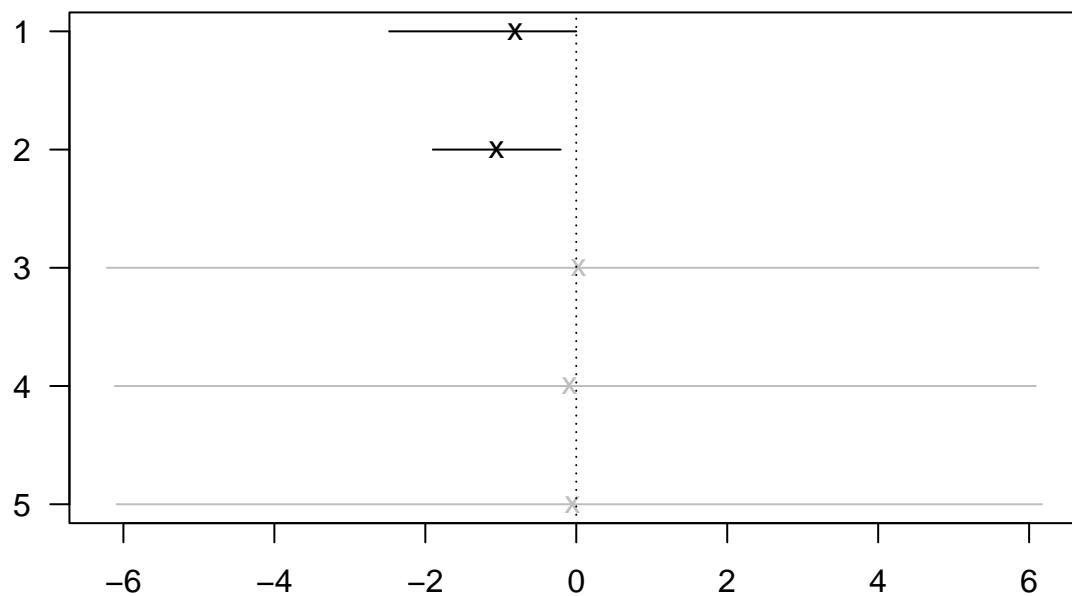
```
coefsplot(covname = "avcondition", object = paramod) #Condition
```

## avcondition



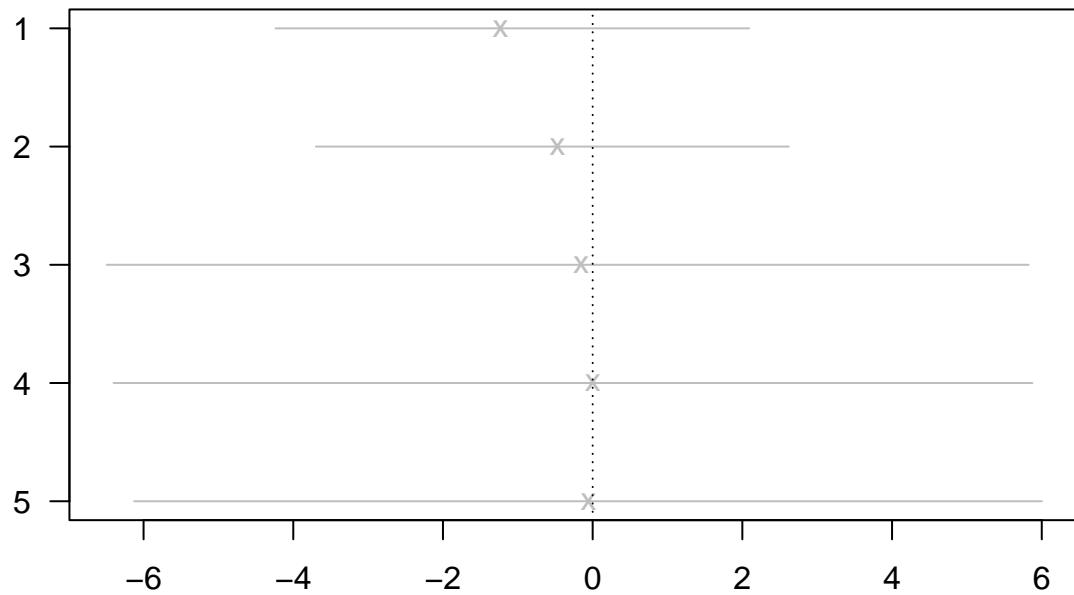
```
coefsplot(covname = "avlength", object = paramod) #Length
```

## avlength



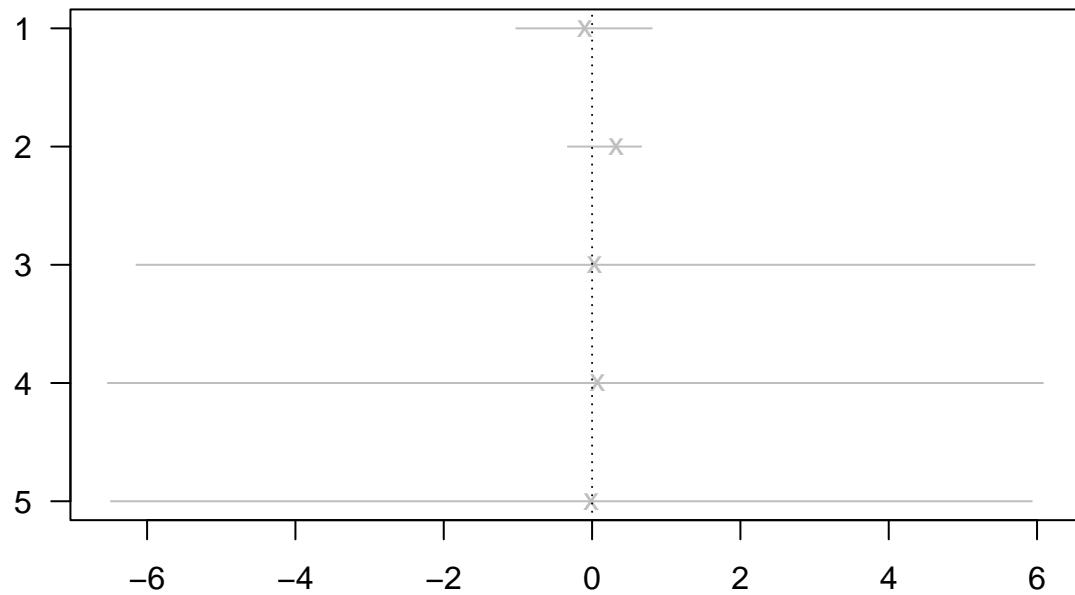
```
coefsplot(covname = "T", object = paramod) #Temperature
```

T



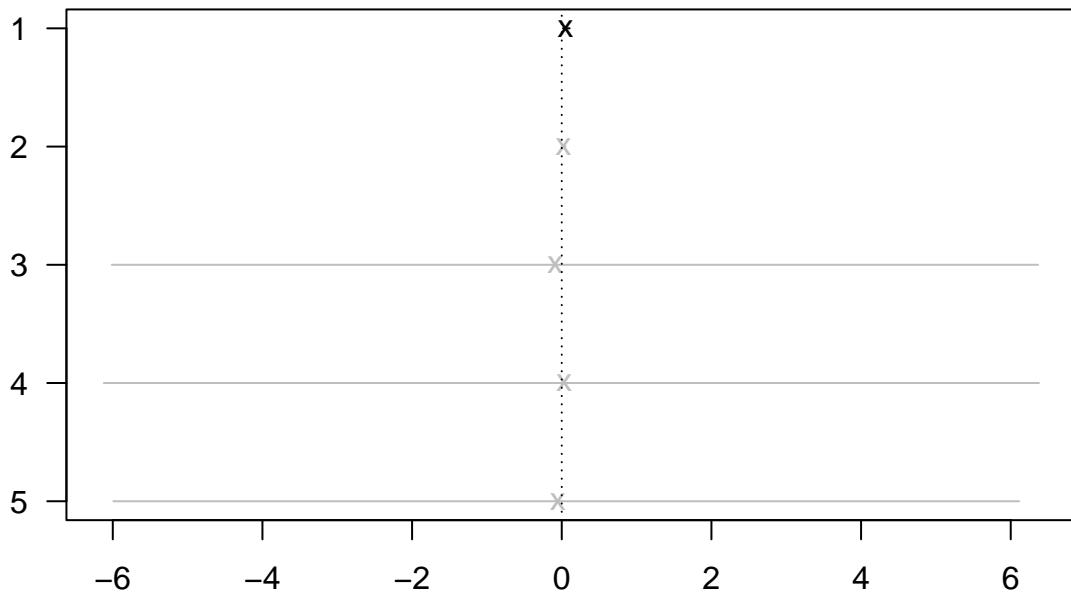
```
coefsplot(covname = "02", object = paramod) #Oxygen
```

## O2



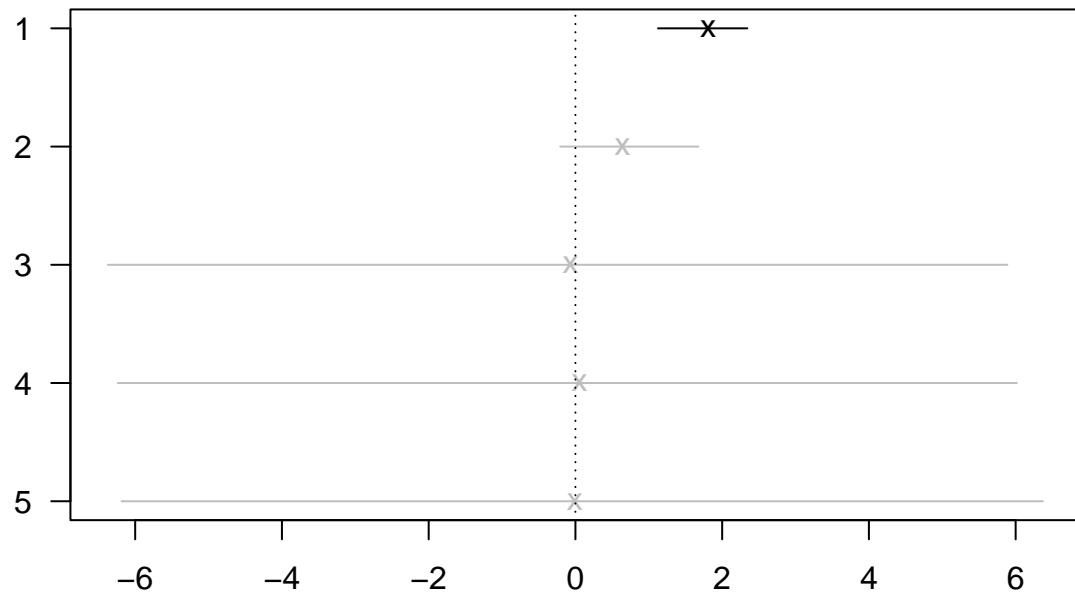
```
coefsplot(covname = "Con", object = paramod) #Conductivity
```

## Con



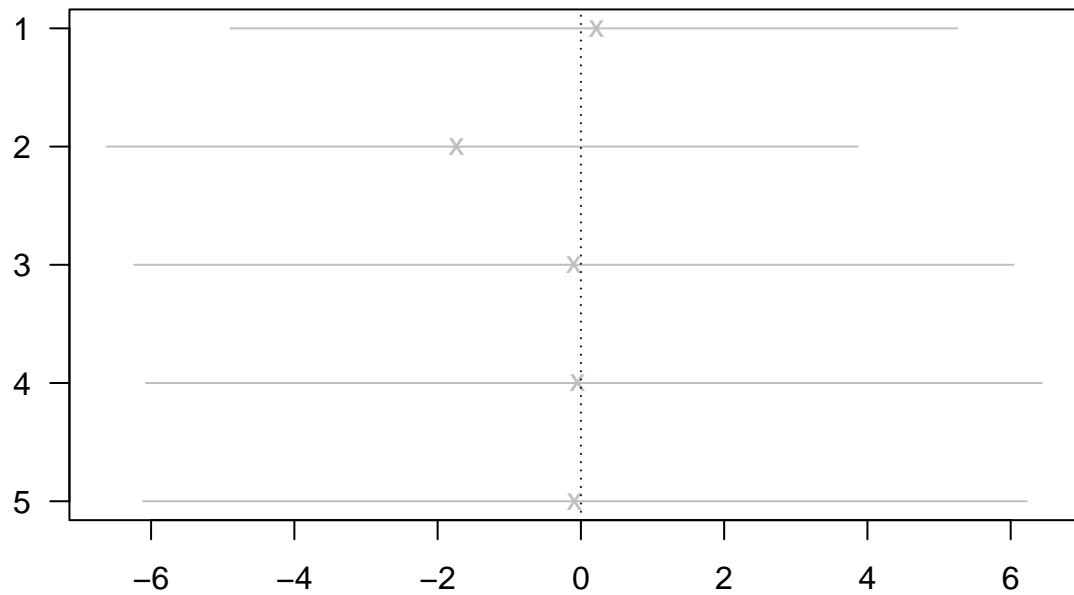
```
coefsplot(covname = "COD", object = paramod) #COD
```

## COD



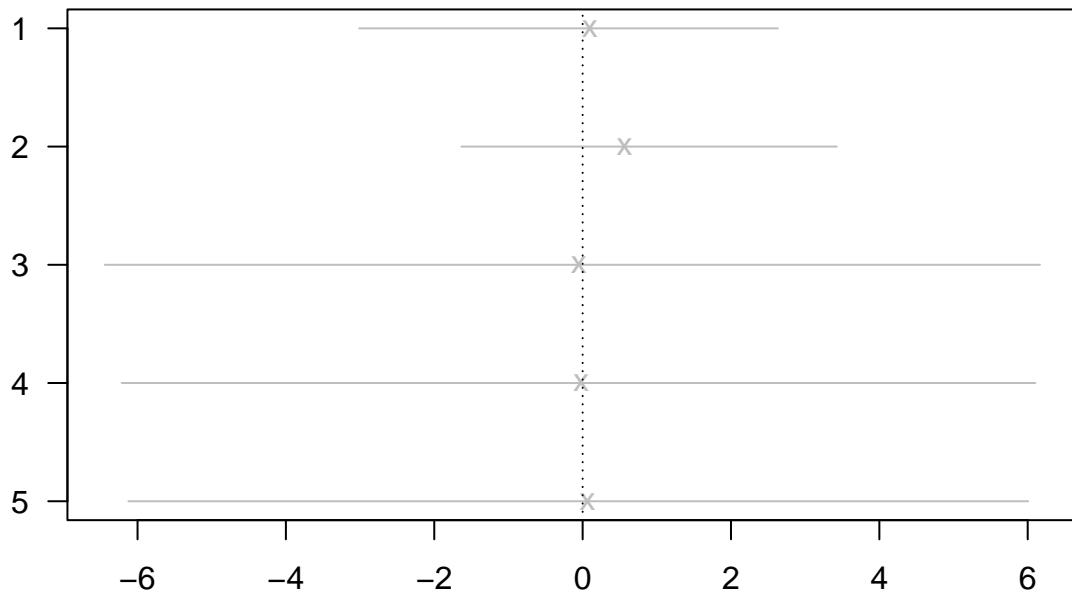
```
coefsplot(covname = "NH4", object = paramod) #NH4
```

## NH4



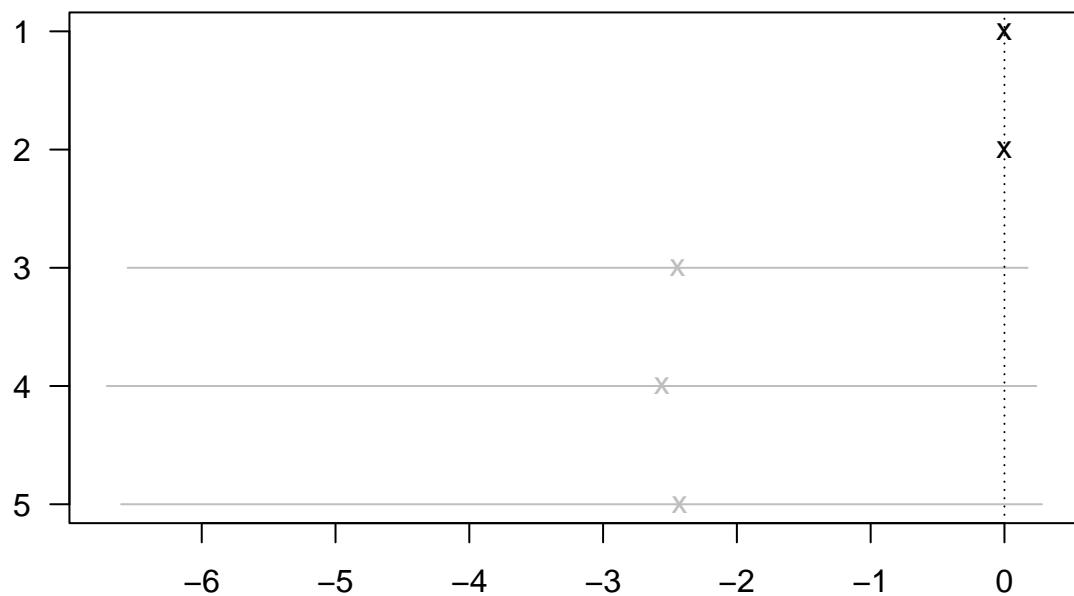
```
coefsplot(covname = "Nt", object = paramod) #Nt
```

**Nt**



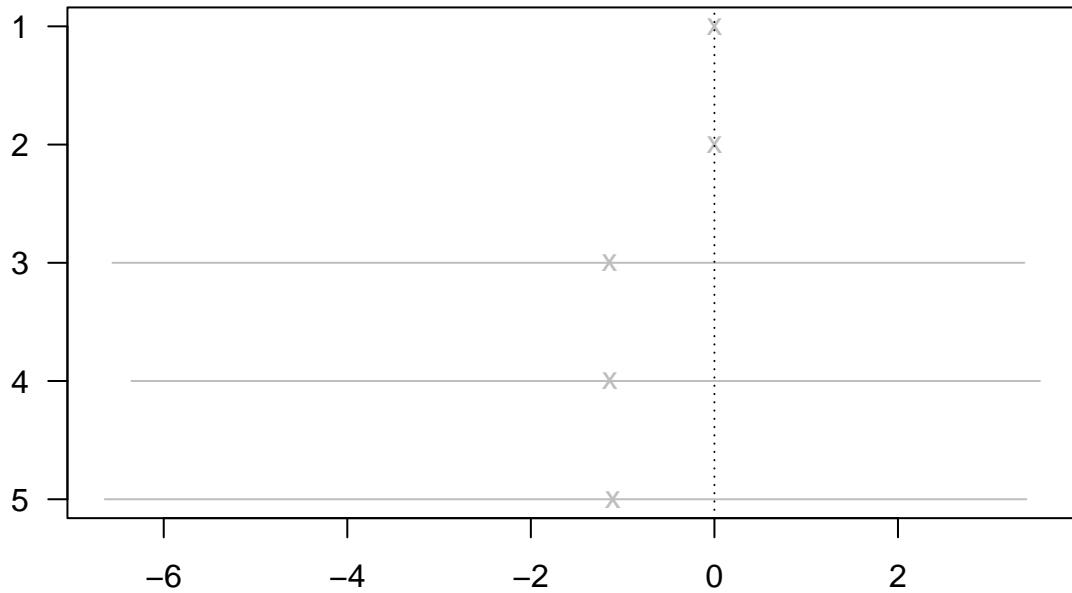
```
coefsplot(covname = "netcen", object = paramod) #netcen
```

## netcen



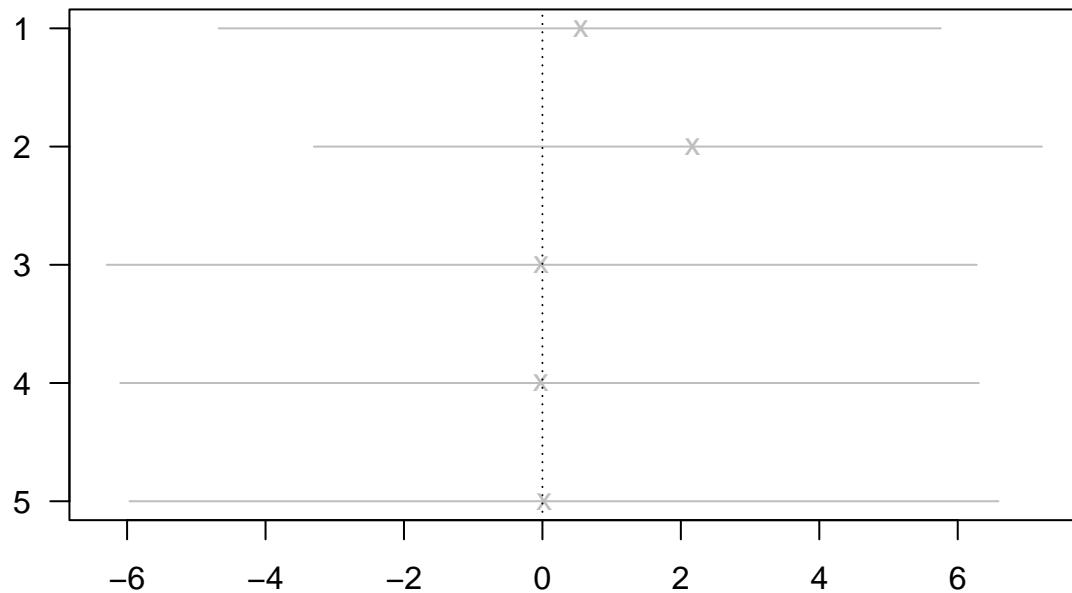
```
coefsplot(covname = "updist", object = paramod) #updist
```

## updist



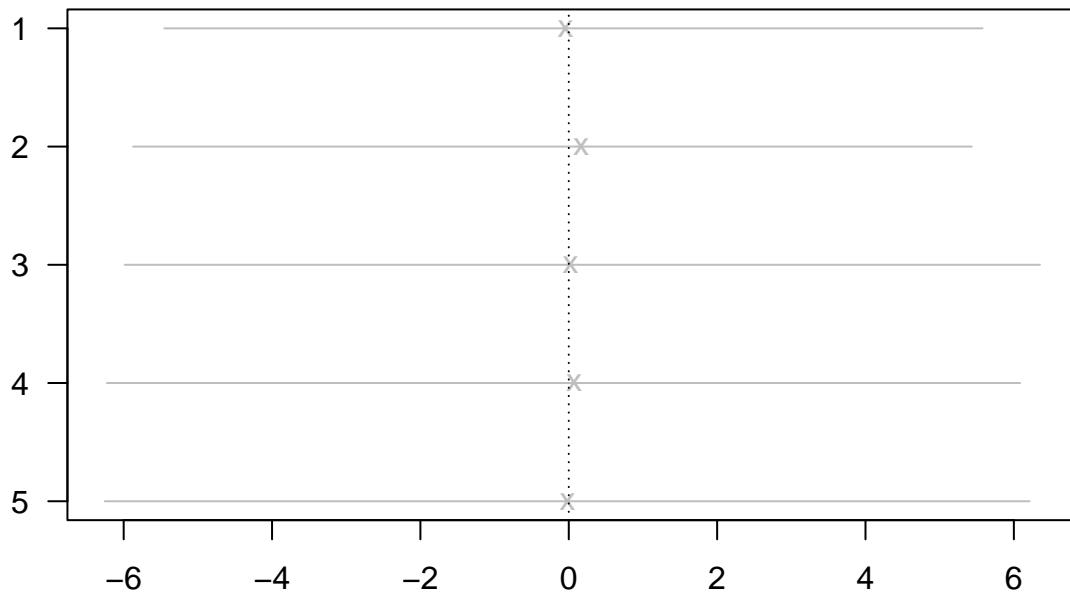
```
coefsplot(covname = "pool_riffle", object = paramod) #poolriffle
```

## pool\_riffle



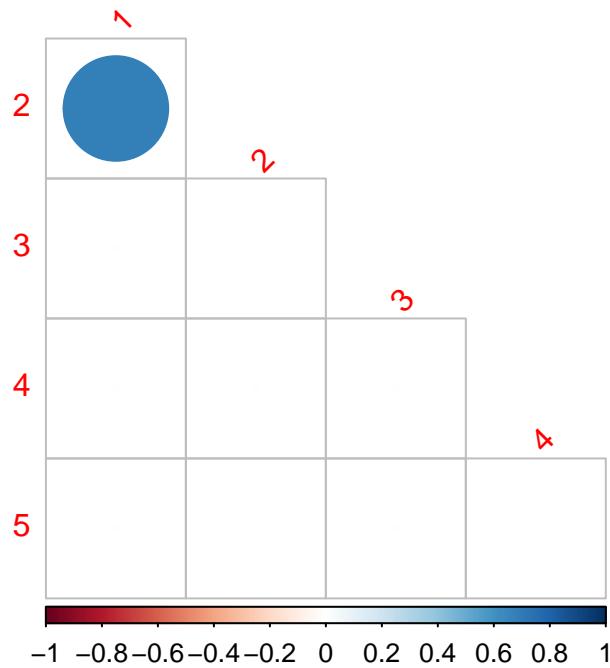
```
coefsplot(covname = "meander", object = paramod) #meander
```

## meander



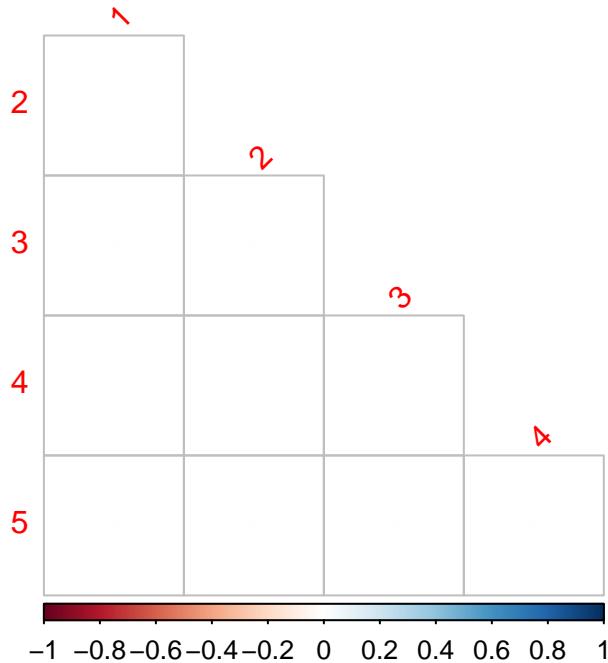
```
envcors <- get.enviro.cor(paramod)
rescors <- get.residual.cor(paramod)
corrplot(envcors$sig.cor, type = "lower", diag = FALSE, title = "Correlations due to covariates",
         mar = c(3, 0.5, 2, 1), tl.srt = 45)
```

## Correlations due to covariates



```
corrplot(rescors$sig.cor, type = "lower", diag = FALSE, title = "Residual correlations",
         mar = c(3, 0.5, 2, 1), tl.srt = 45)
```

## Residual correlations

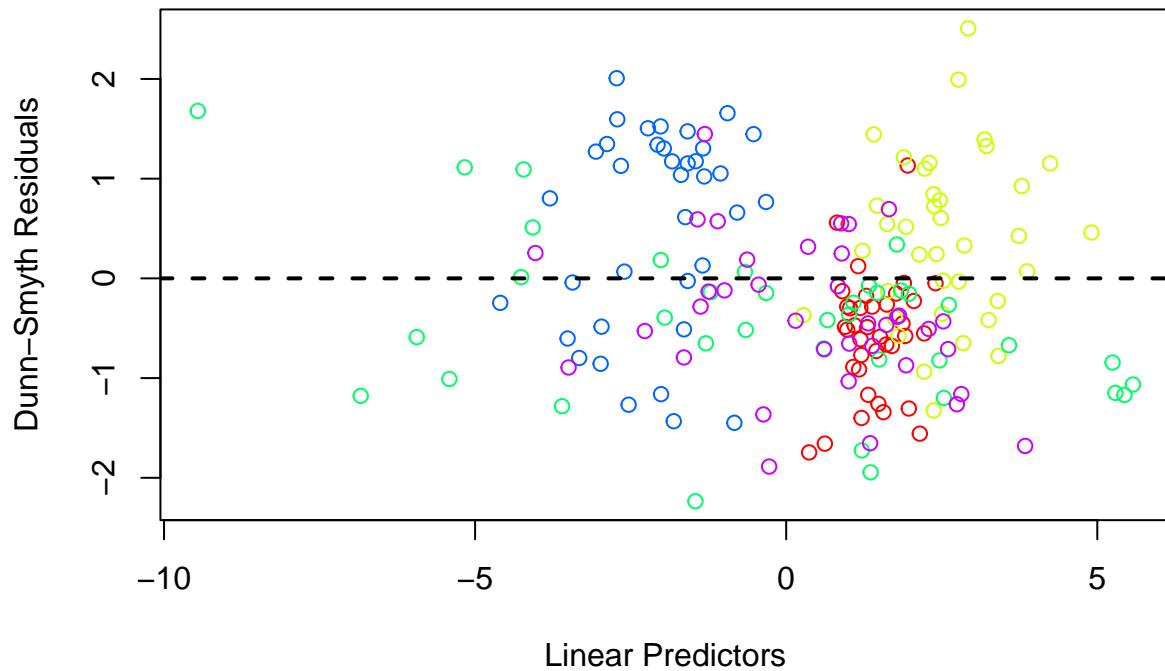


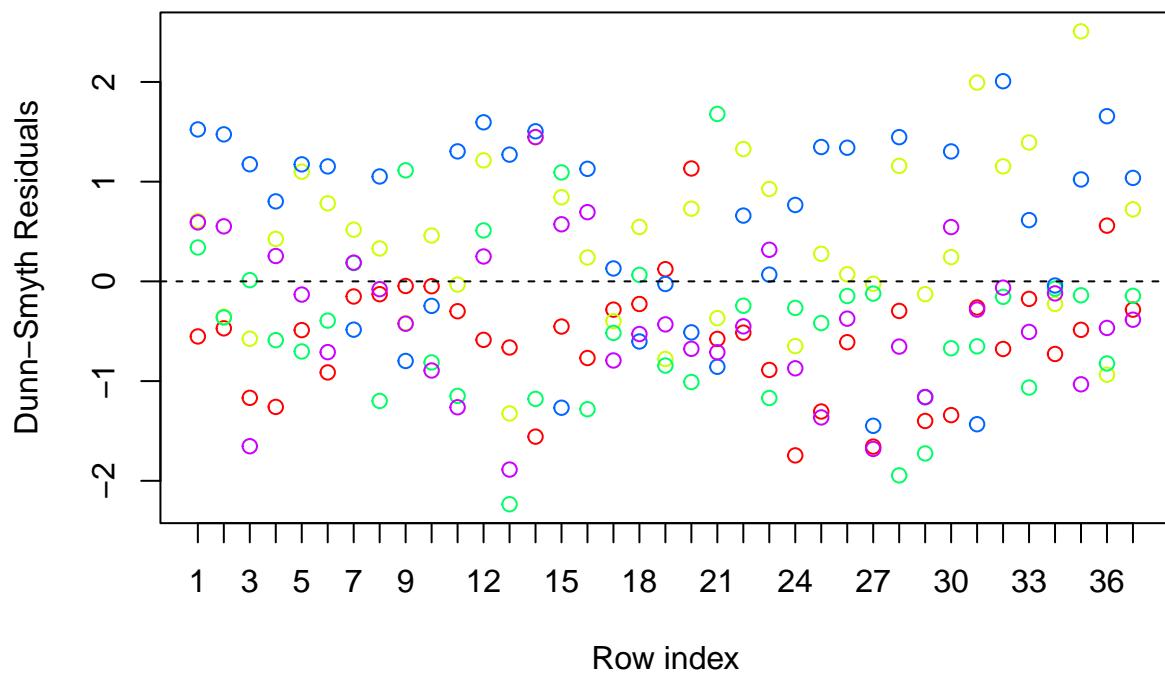
### 7.1 BORAL analysis for median infection intensities of parasites

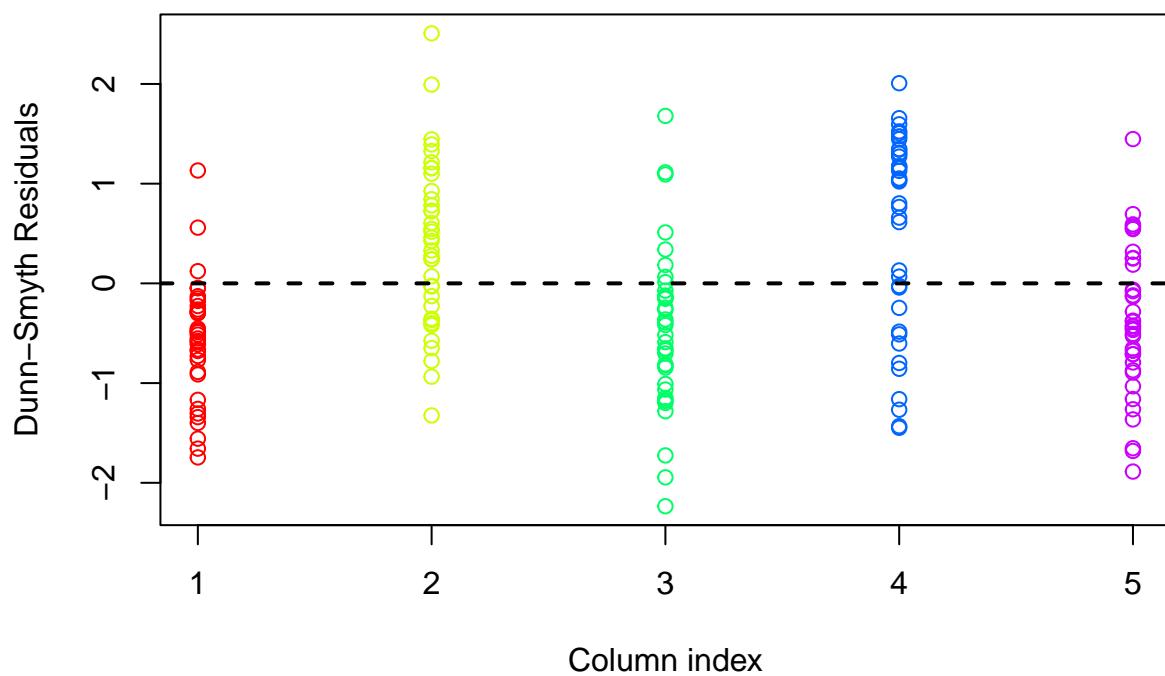
```
y <- round(cbind(medin$Gyr, medin$Tri, medin$Glu, medin$Con, medin$Ang))
paramod <- boral(y, X = X, family = "negative.binomial", mcmc.control = example_mcmc_control,
  model.name = testpath, lv.control = list(num.lv = 2, type = "independent"), save.model = TRUE)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 185
##   Unobserved stochastic nodes: 338
##   Total graph size: 2173
##
## Initializing model
plot(paramod)

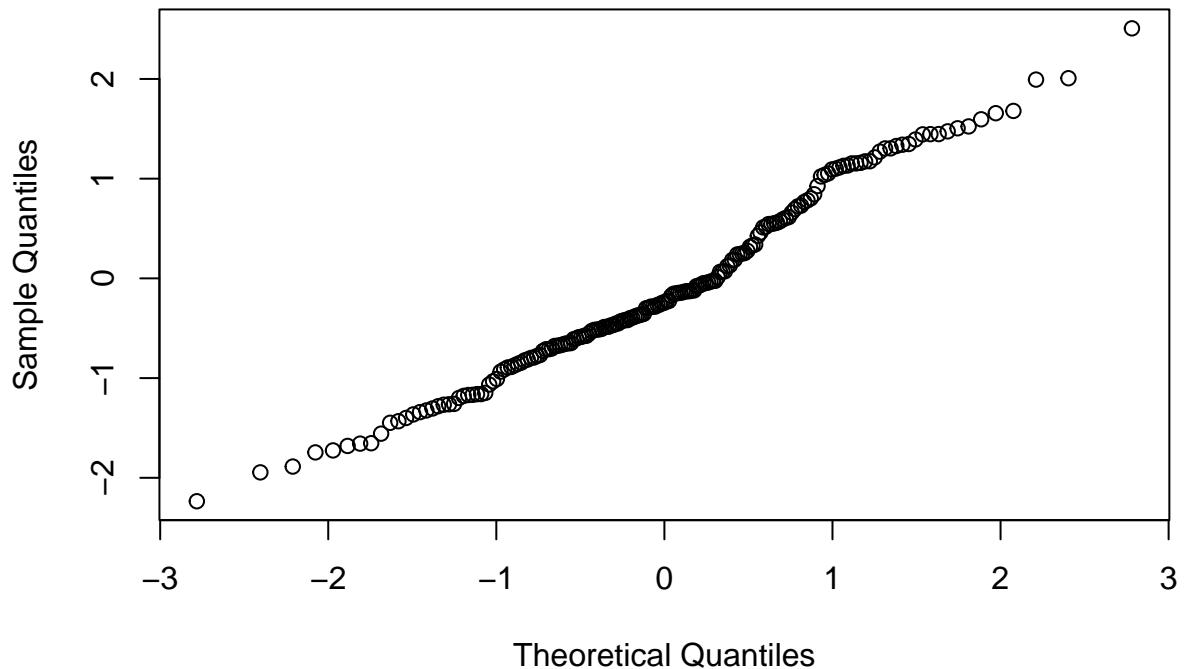
## NULL
```





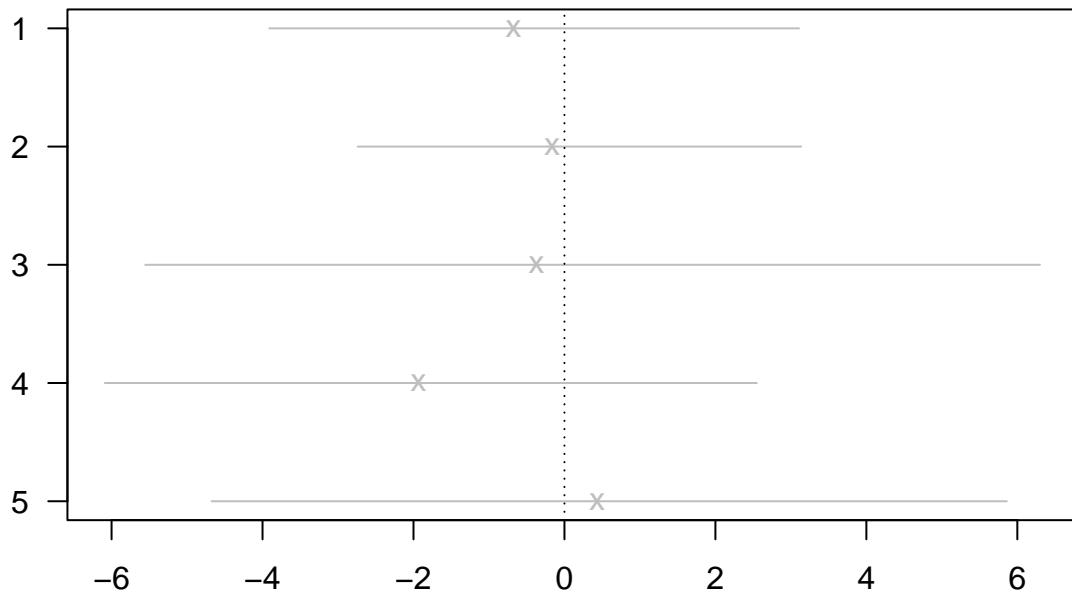


## Quantile–Quantile Plot



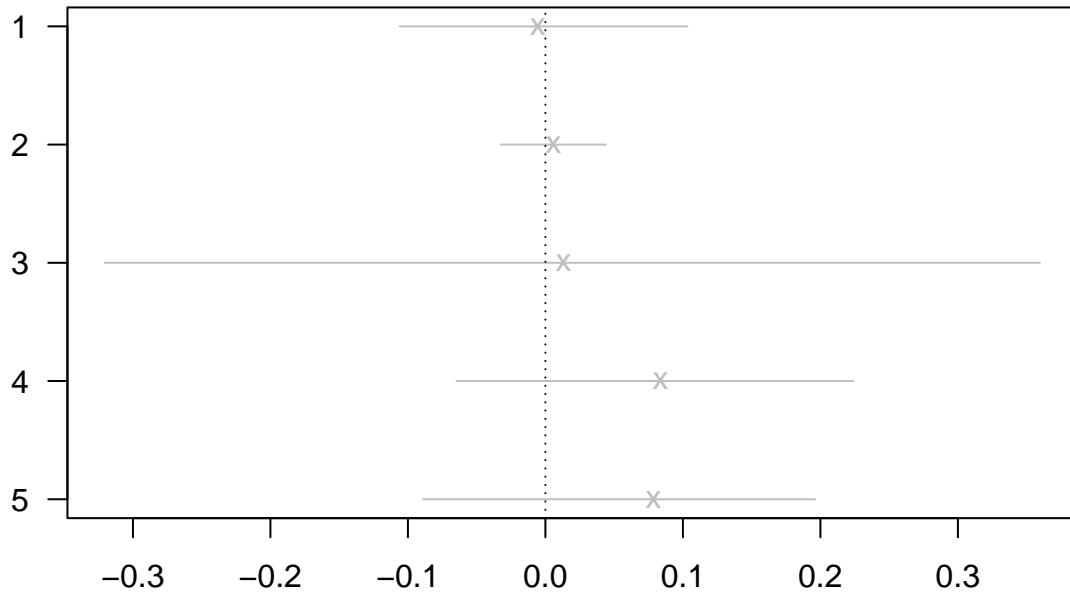
```
coefsplot(covname = "avcondition", object = paramod) #Condition
```

## avcondition



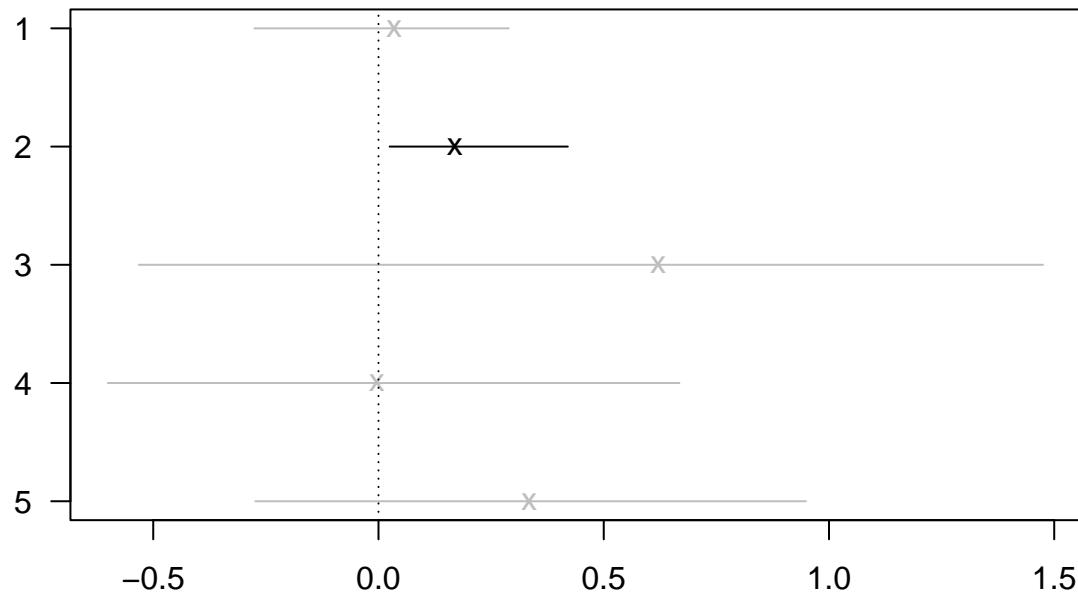
```
coefsplot(covname = "avlength", object = paramod) #Length
```

## avlength



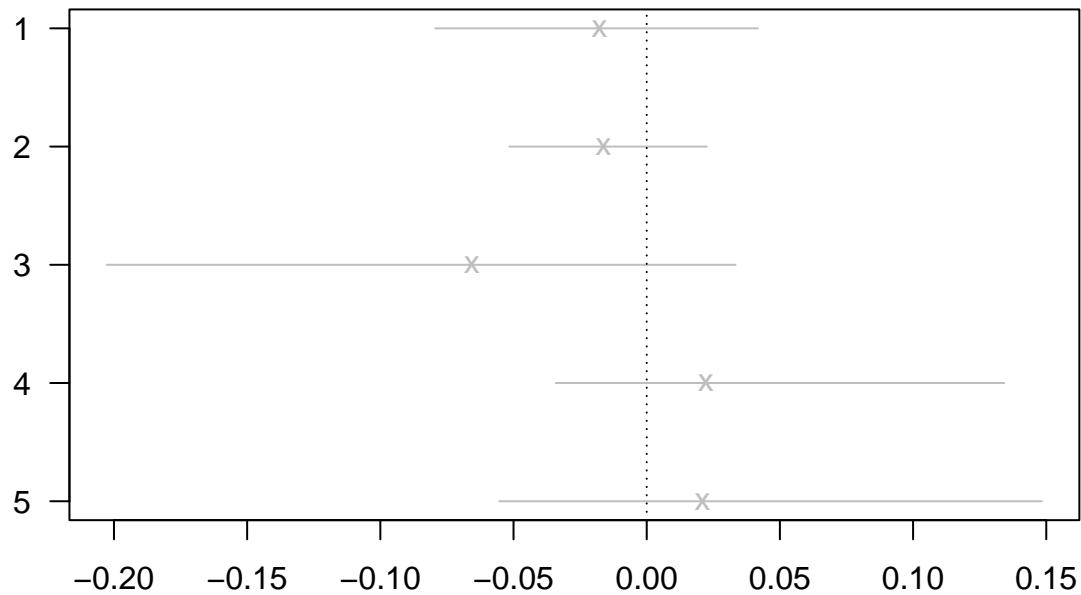
```
coefsplot(covname = "T", object = paramod) #Temperature
```

T



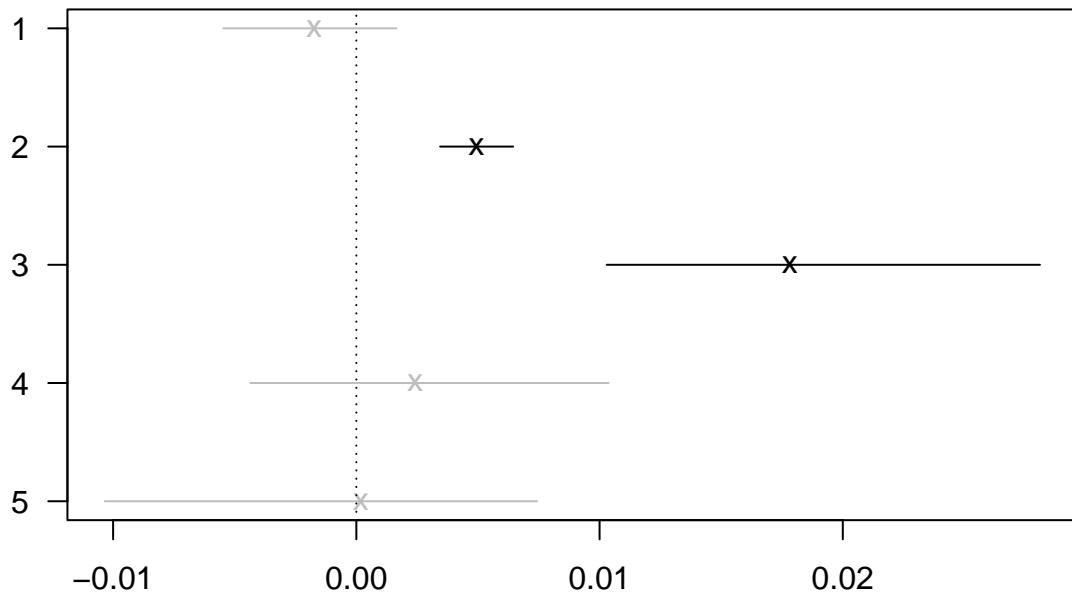
```
coefsplot(covname = "02", object = paramod) #Oxygen
```

## O2



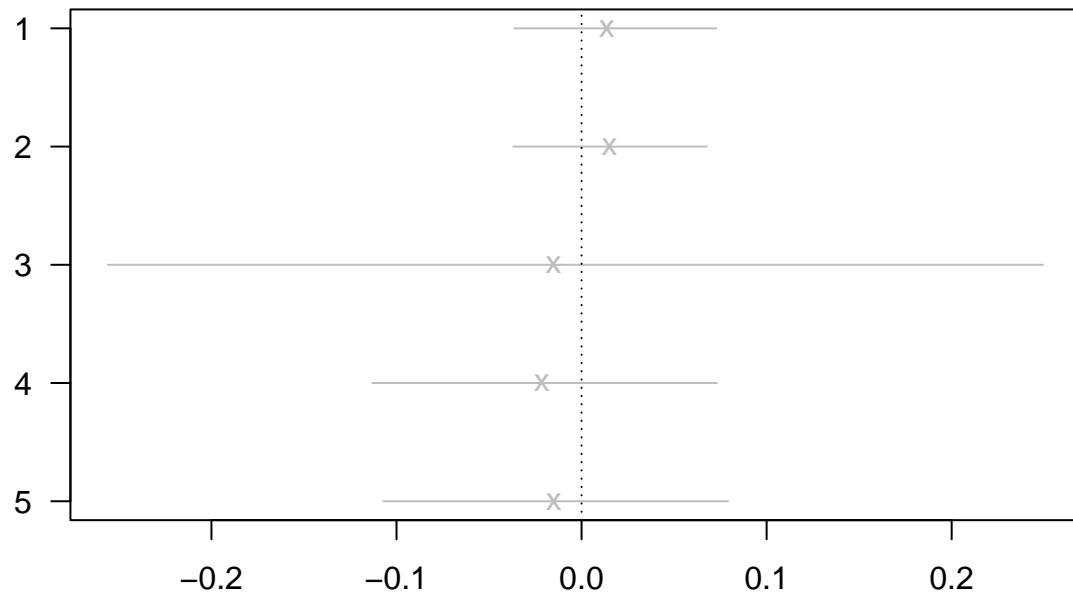
```
coefsplot(covname = "Con", object = paramod) #Conductivity
```

## Con



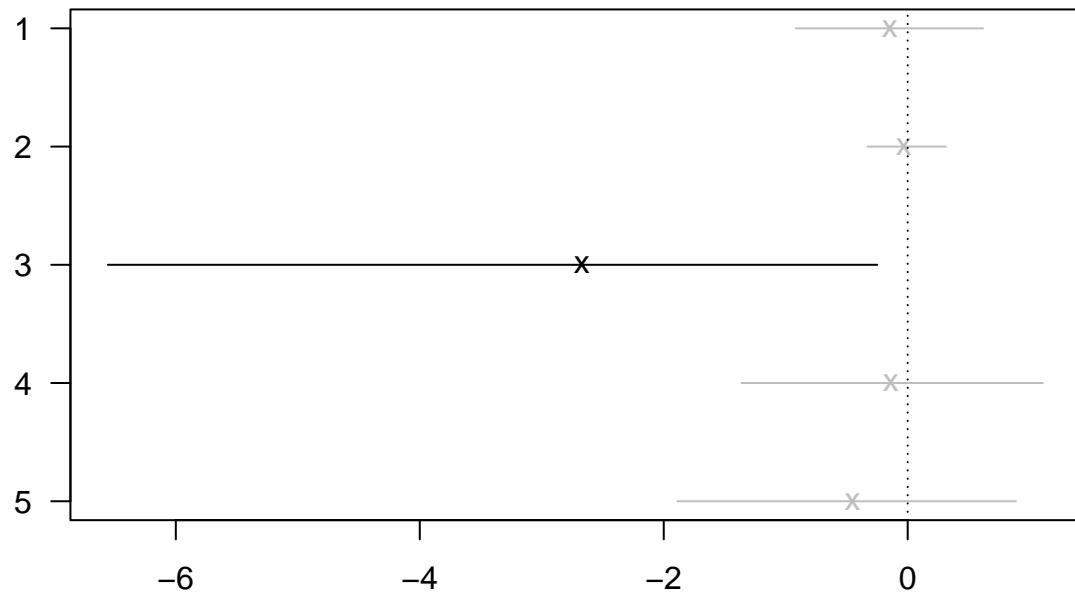
```
coefsplot(covname = "COD", object = paramod) #COD
```

## COD



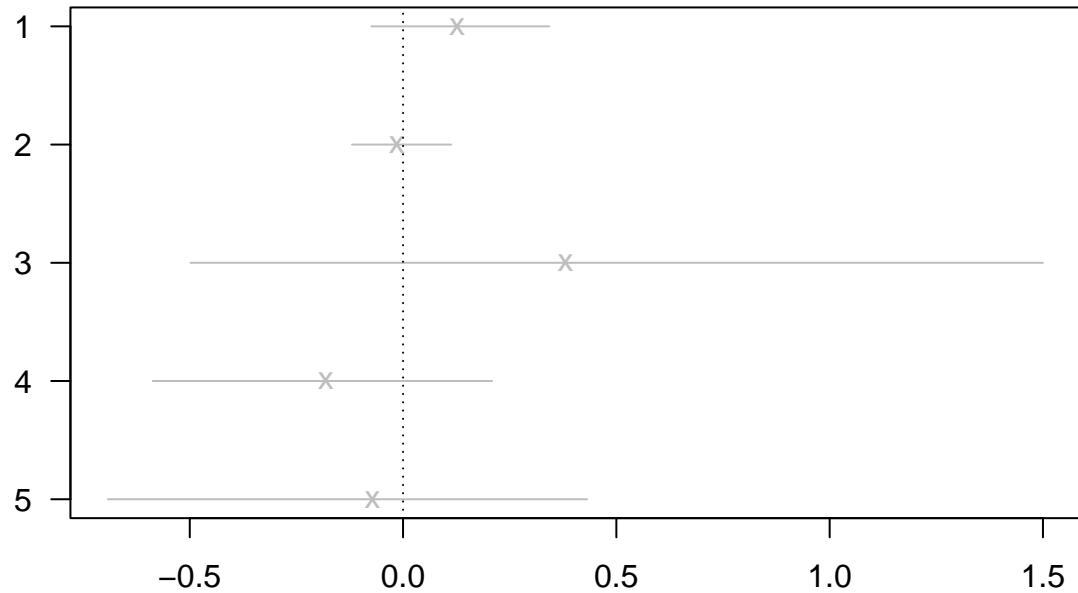
```
coefsplot(covname = "NH4", object = paramod) #NH4
```

## NH4



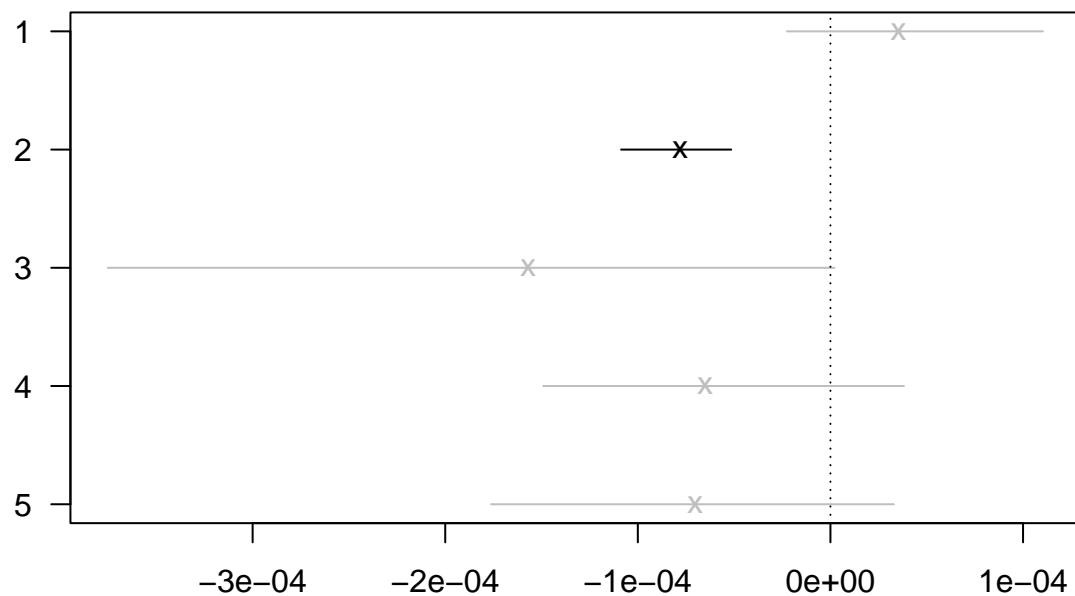
```
coefsplot(covname = "Nt", object = paramod) #Nt
```

**Nt**



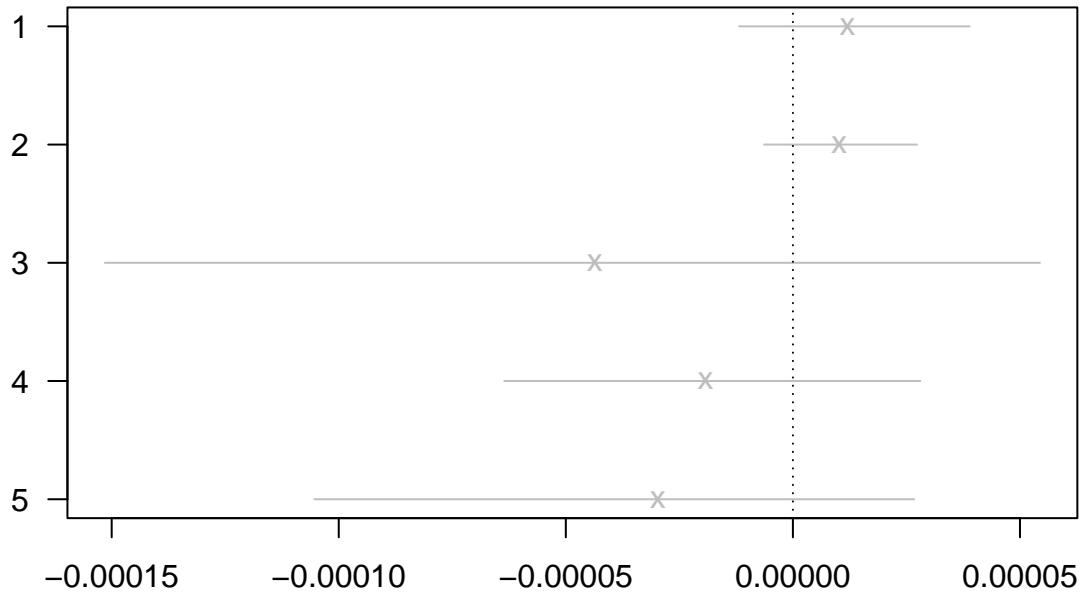
```
coefsplot(covname = "netcen", object = paramod) #netcen
```

## netcen



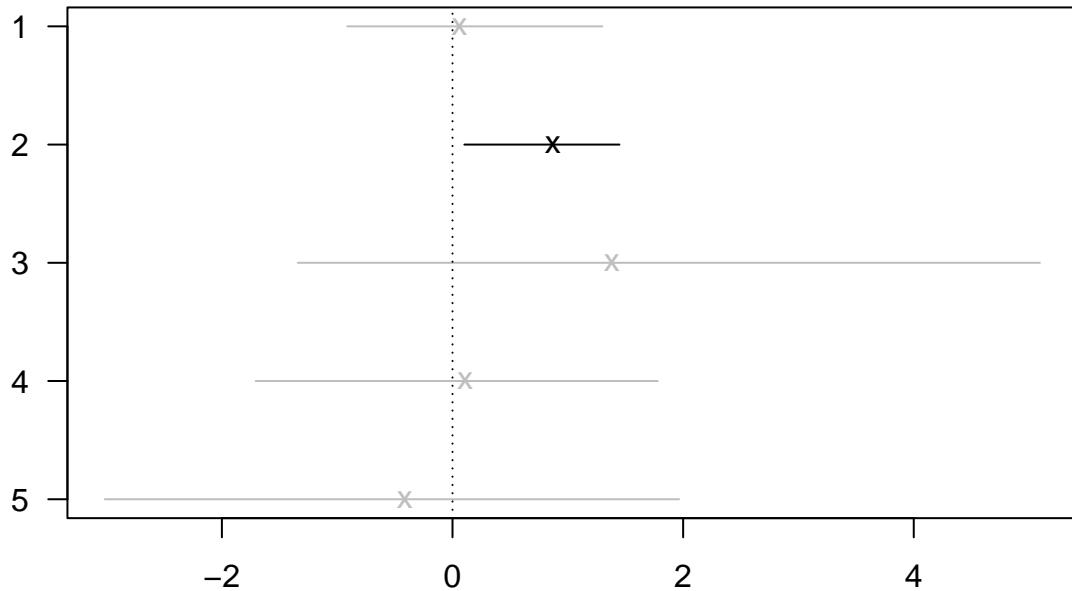
```
coefsplot(covname = "updist", object = paramod) #updist
```

## updist



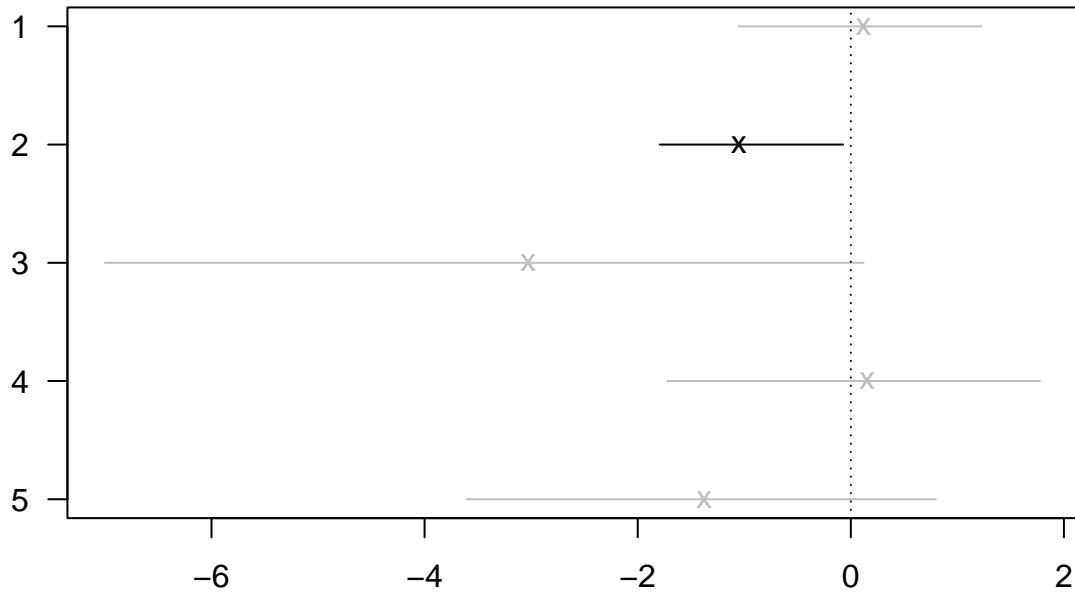
```
coefsplot(covname = "pool_riffle", object = paramod) #poolriffle
```

## pool\_riffle



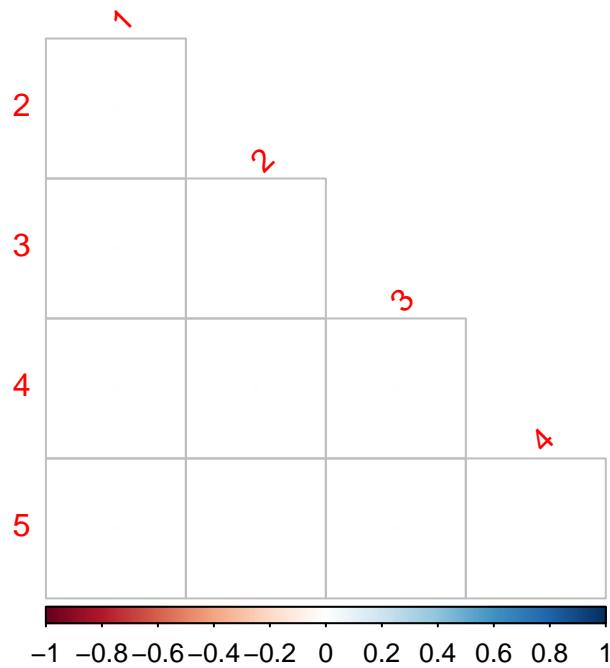
```
coefsplot(covname = "meander", object = paramod) #meander
```

## meander



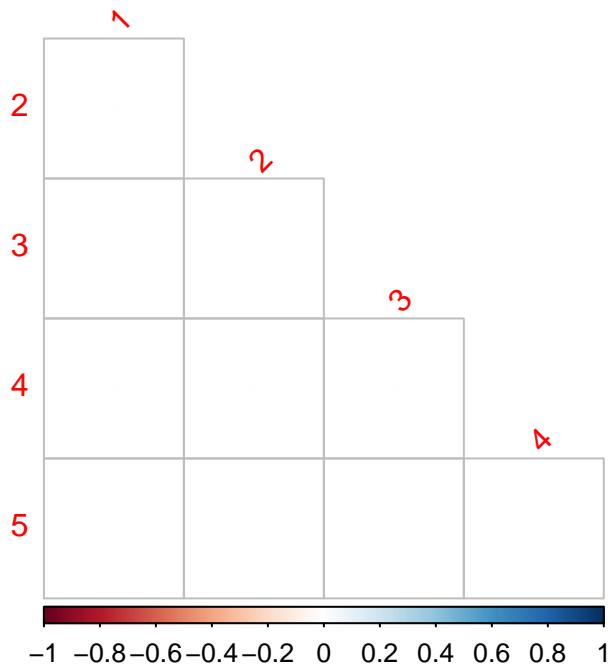
```
envcors <- get.enviro.cor(paramod)
rescors <- get.residual.cor(paramod)
corrplot(envcors$sig.cor, type = "lower", diag = FALSE, title = "Correlations due to covariates",
         mar = c(3, 0.5, 2, 1), tl.srt = 45)
```

## Correlations due to covariates



```
corrplot(rescors$sig.cor, type = "lower", diag = FALSE, title = "Residual correlations",
         mar = c(3, 0.5, 2, 1), tl.srt = 45)
```

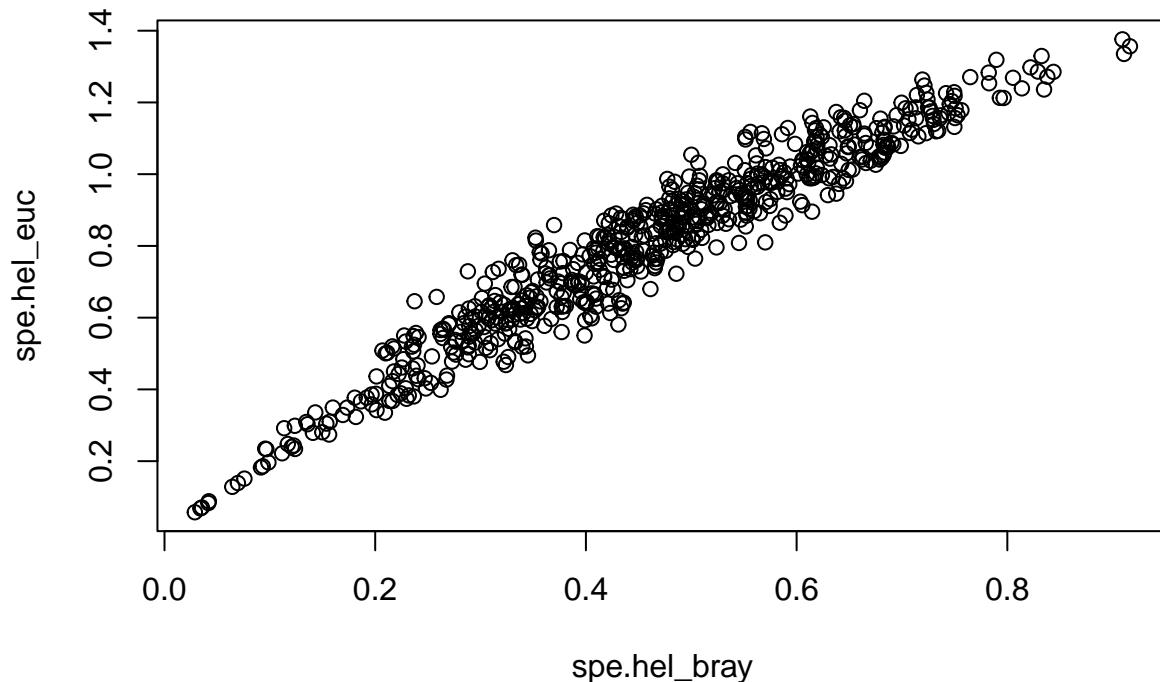
## Residual correlations



## 8. Multivariate analysis

### 8.1 Component communities

```
# Component communities: Bray-Curtis dissimilarities based on Hellinger  
# transformed average abundance data  
spe.hel_bray <- vegdist(decostand(avab[, -1], na.rm = T, method = "hellinger"), method = "bray",  
na.rm = T)  
  
# Check whether Euclidean and Bray-Curtis distances are comparable  
spe.hel_euc <- vegdist(decostand(avab[, -1], na.rm = T, method = "hellinger"), method = "euc",  
na.rm = T)  
plot(spe.hel_bray, spe.hel_euc)
```



```

mantel(spe.hel_bray, spe.hel_euc)

##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = spe.hel_bray, ydis = spe.hel_euc)
##
## Mantel statistic r: 0.9648
##      Significance: 0.001
##
## Upper quantiles of permutations (null model):
##    90%   95% 97.5%   99%
## 0.118 0.149 0.175 0.201
## Permutation: free
## Number of permutations: 999
model_adonis = adonis(spe.hel_bray ~ avlength + avcondition + T_av + O2_sat_av +
  Con_av + COD_av + NH4._av + Nt_av + pool_riffle + meander + netcen + updist,
  data = environment2)

## 'adonis' will be deprecated: use 'adonis2' instead
model_adonis$aov.tab

## Permutation: free
## Number of permutations: 999
##

```

```

## Terms added sequentially (first to last)
##
##          Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## avlength     1  0.1863  0.18629  1.8398  0.04303  0.119
## avcondition  1  0.1305  0.13055  1.2893  0.03016  0.278
## T_av         1  0.3204  0.32039  3.1643  0.07401  0.024 *
## O2_sat_av   1  0.1368  0.13678  1.3509  0.03160  0.247
## Con_av       1  0.1326  0.13262  1.3098  0.03064  0.270
## COD_av       1  0.0657  0.06568  0.6486  0.01517  0.691
## NH4._av      1  0.2040  0.20399  2.0146  0.04712  0.114
## Nt_av         1  0.1451  0.14509  1.4329  0.03352  0.243
## pool_riffle  1  0.0853  0.08529  0.8424  0.01970  0.523
## meander       1  0.2029  0.20286  2.0035  0.04686  0.099 .
## netcen        1  0.2173  0.21728  2.1459  0.05019  0.060 .
## updist        1  0.0719  0.07193  0.7104  0.01662  0.608
## Residuals    24  2.4301  0.10125           0.56137
## Total         36  4.3288                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

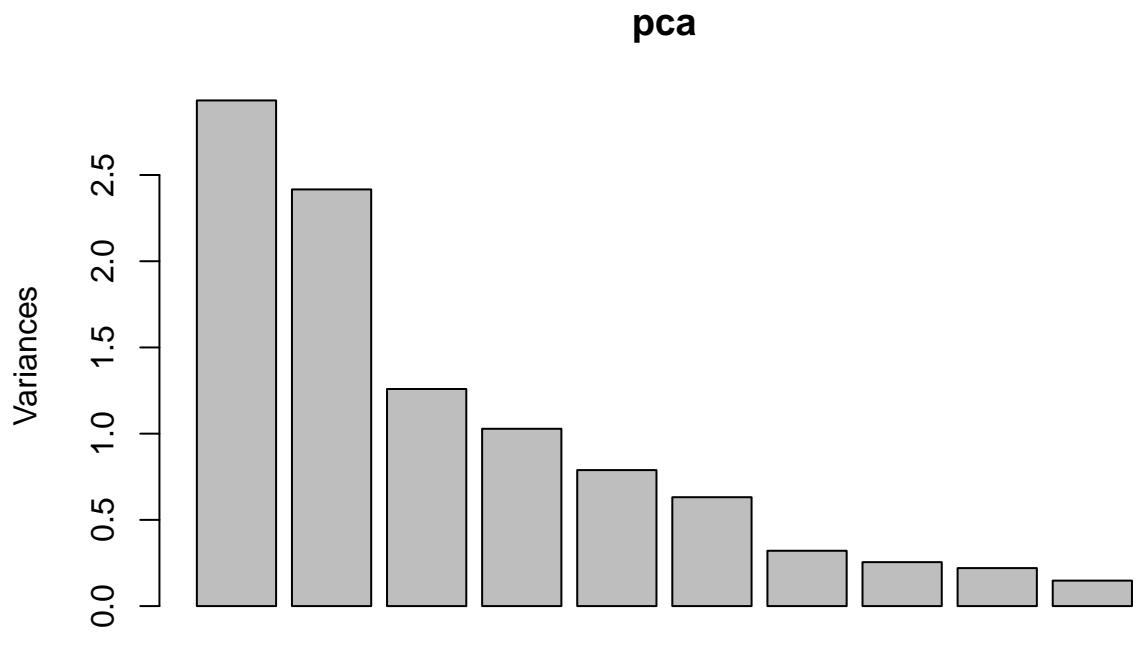
# environmental variables
env_select <- environment2[, c("T_av", "O2_sat_av", "Con_av", "COD_av", "NH4._av",
  "Nt_av", "pool_riffle", "meander", "netcen", "updist")]
env_select$pool_riffle <- as.numeric(env_select$pool_riffle)
env_select$meander <- as.numeric(env_select$meander)

pca <- prcomp(env_select, scale. = T)
summary(pca)

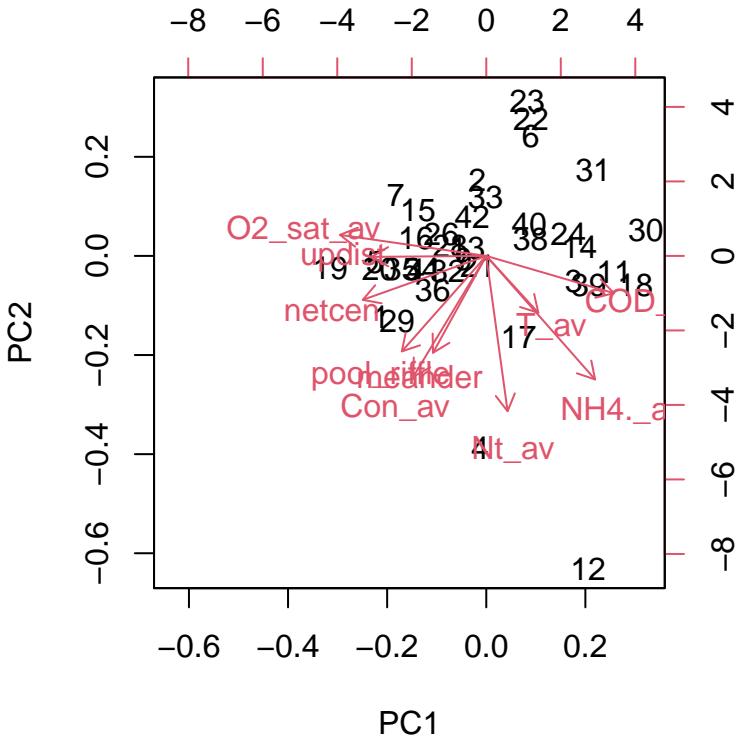
## Importance of components:
##                  PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation 1.7124 1.5545 1.1221 1.0140 0.88807 0.79463 0.56647
## Proportion of Variance 0.2933 0.2416 0.1259 0.1028 0.07887 0.06314 0.03209
## Cumulative Proportion 0.2933 0.5349 0.6608 0.7636 0.84248 0.90563 0.93771
##                  PC8     PC9     PC10
## Standard deviation 0.50483 0.46939 0.38429
## Proportion of Variance 0.02549 0.02203 0.01477
## Cumulative Proportion 0.96320 0.98523 1.00000

plot(pca)

```



```
biplot(pca)
```



```

# Assess the effect of environmental variables on parasite component community
# dissimilarities using distance based RDA
spe.rda <- dbrda(spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av +
    pool_riffle + meander, data = env_select)
anova(spe.rda)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_ri
##                 Df SumOfSqs      F Pr(>F)
## Model     8   1.1664 1.2909  0.141
## Residual 28   3.1624

RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.06072755

mod0 <- dbrda(spe.hel_bray ~ 1, env_select[, -c(9:10)]) # Model with intercept only #edit_PH
mod1 <- dbrda(spe.hel_bray ~ ., env_select[, -c(9:10)]) # Model with all explanatory variables #edit_
step.res <- ordiR2step(mod0, mod1, direction = "both", perm.max = 200)

## Step: R2.adj= 0
## Call: spe.hel_bray ~ 1
##
##                    R2.adjusted
## <All variables> 0.060727547

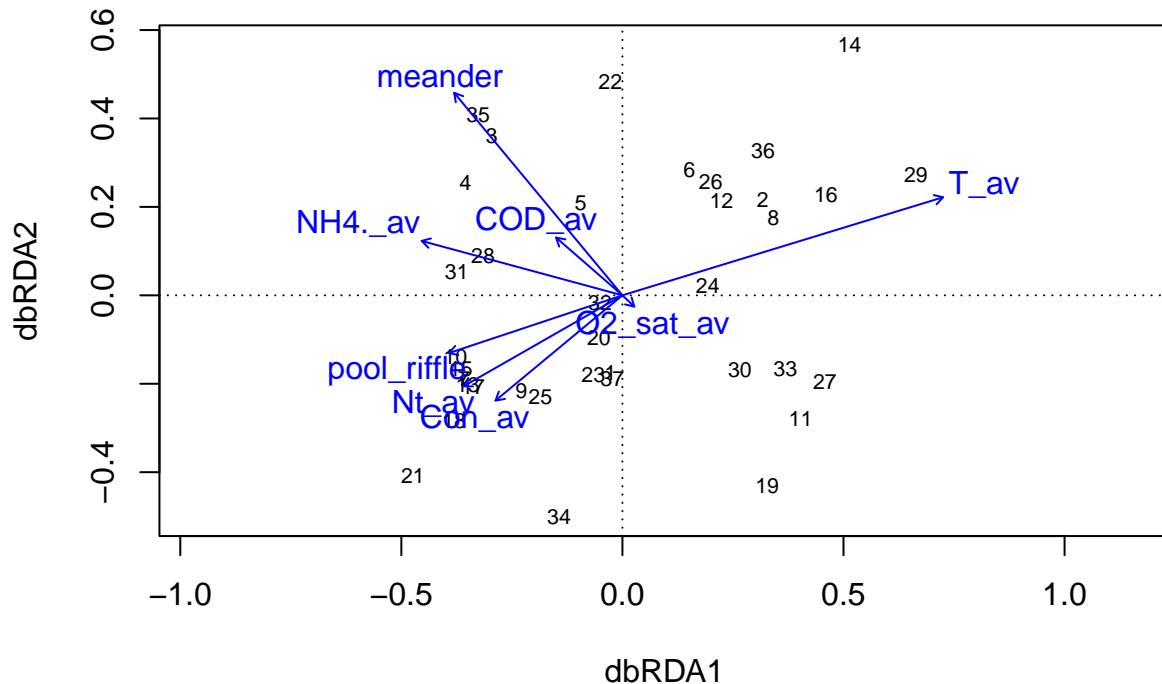
```

```

## + T_av          0.036398792
## + NH4._av      0.020208612
## + meander      0.018502880
## + O2_sat_av   0.004277611
## + Con_av       0.001872668
## + pool_riffle  0.001742860
## <none>          0.000000000
## + Nt_av         -0.002060170
## + COD_av       -0.017968936
##
##              Df      AIC      F Pr(>F)
## + T_av    1 54.788 2.3599   0.06 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
step.res$anova # Summary table

## NULL
plot(spe.rda, scaling = 1) # it is for technical reasons not possible to plot both site and species sc

```



```

summary(spe.rda)

##
## Call:
## dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av +      NH4._av + Nt_av + pool_riffle)
## 
## Partitioning of squared Bray distance:

```

```

## Inertia Proportion
## Total      4.329    1.0000
## Constrained 1.166    0.2695
## Unconstrained 3.162    0.7305
##
## Eigenvalues, and their contribution to the squared Bray distance
##
## Importance of components:
##          dbRDA1 dbRDA2 dbRDA3 dbRDA4 dbRDA5 dbRDA6
## Eigenvalue 0.5740 0.3372 0.17566 0.09415 0.042525 0.0020670
## Proportion Explained 0.1326 0.0779 0.04058 0.02175 0.009824 0.0004775
## Cumulative Proportion NA NA NA NA NA NA
##          idbRDA1 idbRDA2 MDS1 MDS2 MDS3 MDS4 MDS5
## Eigenvalue -0.014797 -0.04441 1.3249 0.8027 0.4680 0.31323 0.29121
## Proportion Explained 0.003418 0.01026 0.3061 0.1854 0.1081 0.07236 0.06727
## Cumulative Proportion NA NA NA NA NA NA NA
##          MDS6 MDS7 MDS8 MDS9 MDS10 MDS11 MDS12
## Eigenvalue 0.13644 0.11438 0.08987 0.07065 0.06425 0.02766 0.01576
## Proportion Explained 0.03152 0.02642 0.02076 0.01632 0.01484 0.00639 0.00364
## Cumulative Proportion NA NA NA NA NA NA NA
##          MDS13 MDS14 iMDS1 iMDS2 iMDS3
## Eigenvalue 0.011872 2.725e-04 -0.0020314 -0.007356 -0.01078
## Proportion Explained 0.002743 6.295e-05 0.0004693 0.001699 0.00249
## Cumulative Proportion NA NA NA NA NA NA
##          iMDS4 iMDS5 iMDS6 iMDS7 iMDS8 iMDS9
## Eigenvalue -0.014744 -0.01935 -0.022495 -0.029849 -0.03342 -0.041489
## Proportion Explained 0.003406 0.00447 0.005197 0.006895 0.00772 0.009584
## Cumulative Proportion NA NA NA NA NA NA
##          iMDS10 iMDS11 iMDS12 iMDS13 iMDS14
## Eigenvalue -0.05984 -0.06439 -0.06995 -0.09116 -0.10203
## Proportion Explained 0.01382 0.01487 0.01616 0.02106 0.02357
## Cumulative Proportion NA NA NA NA NA
##
## Accumulated constrained eigenvalues
## Importance of components:
##          dbRDA1 dbRDA2 dbRDA3 dbRDA4 dbRDA5 dbRDA6 idbRDA1
## Eigenvalue 0.5740 0.3372 0.1757 0.09415 0.04252 0.002067 -0.01480
## Proportion Explained 0.4921 0.2891 0.1506 0.08072 0.03646 0.001772 0.01269
## Cumulative Proportion NA NA NA NA NA NA NA
##          idbRDA2
## Eigenvalue -0.04441
## Proportion Explained 0.03807
## Cumulative Proportion NA
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 3.533199
##
## Site scores (weighted sums of species scores)
##
##          dbRDA1 dbRDA2 dbRDA3 dbRDA4 dbRDA5 dbRDA6
## 1 -0.07530 -0.62580 -0.63602 -2.643196 -0.34427 -23.8946

```

```

## 2   0.87242  0.77832 -0.32068  2.074749 -0.25800  41.3754
## 3  -0.81094  1.29200 -0.57735 -0.708420 -2.35800  -5.0035
## 4  -0.97450  0.91051  1.79849 -2.192492 -1.99248  -4.7134
## 5  -0.25889  0.74559 -1.30995 -1.418227  0.33185 -21.9720
## 6   0.41640  1.01888 -0.34551 -2.318324  1.61924 -24.8949
## 7  -0.98286 -0.67495  0.62017  1.174218  1.55155 -0.8974
## 8   0.93807  0.62635 -3.10878 -0.765073  2.41981 -3.0764
## 9  -0.62816 -0.77245 -1.30540  1.806170  1.30543 -41.2679
## 10 -1.03727 -0.50089  0.62572  0.369727  1.00378  4.7819
## 11  1.10743 -0.99448 -0.66515 -0.071830  0.83402 -14.8528
## 12  0.61624  0.76828  0.87110  0.357836  0.12125  38.2552
## 13 -0.96088 -0.72453  0.47977  1.569869  1.43399 -6.8692
## 14  1.41128  2.03453 -1.67256  0.776255  0.27648  29.9435
## 15 -1.00448 -0.59302  0.65010  0.833950  1.44643  3.4902
## 16  1.26382  0.81623  3.19230  1.380128  1.12523 -10.4959
## 17 -0.92726 -0.73637  0.33774  1.892276  1.07787 -8.2400
## 18 -1.04523 -1.01280 -0.59052  2.897871  0.35870 -14.2865
## 19  0.89987 -1.54255 -0.54435  1.738188 -2.61939  32.8867
## 20 -0.14608 -0.34079  0.78630  1.316880  1.70249 -9.1246
## 21 -1.30124 -1.46318 -1.27370  3.739798  0.91928 -16.7232
## 22 -0.07485  1.73636  0.30728 -1.760677 -0.46041  6.1504
## 23 -0.18008 -0.64105  0.55428 -1.310793 -0.82437 -3.6579
## 24  0.53054  0.08088  1.31623 -3.529128 -1.96164  41.6937
## 25 -0.50901 -0.82222 -1.26480  2.167458 -6.71178 -22.4535
## 26  0.54874  0.92050  1.12409  0.009258  0.07714  34.0773
## 27  1.25804 -0.69768  2.37255 -1.697654 -3.24936 -16.2349
## 28 -0.86554  0.32492 -0.68347  0.704689 -0.16329  13.0277
## 29  1.82276  0.98120  0.33891 -1.852448  4.33581 -8.6587
## 30  0.72974 -0.59907 -1.00943 -0.201449  1.77124 -4.5186
## 31 -1.02904  0.19515  1.05102 -1.468951 -1.36610  23.1359
## 32 -0.13877 -0.06050 -0.19569 -1.899083 -1.10403 -31.4856
## 33  1.01209 -0.59332  2.58400 -0.318337 -1.43566  39.0635
## 34 -0.39319 -1.79743  0.20497 -0.362972 -0.59693 -19.8004
## 35 -0.89506  1.46497 -0.05622 -1.876722 -1.86120  3.5844
## 36  0.87386  1.17335 -3.50567  1.196922  1.51036 -9.4970
## 37 -0.06267 -0.67495 -0.14975  0.389535  2.08495  11.1529
##
##
## Site constraints (linear combinations of constraining variables)
##
##      dbRDA1   dbRDA2   dbRDA3   dbRDA4   dbRDA5   dbRDA6
## 1  -0.16605 -0.69736 -0.45052  0.39949 -1.140870  0.26766
## 2  -0.07683 -0.33692  0.61638  0.50606 -0.381728  1.66676
## 3  -0.10071  0.76299  0.51759  0.02910  0.003366 -0.26259
## 4  -1.55548  0.90013  0.66933 -0.51241 -0.415318 -0.13093
## 5  -0.41914  0.27994 -0.66104 -1.10909  0.384267  0.50060
## 6   0.63090  0.30135 -0.07724  0.37780  0.857007 -0.33310
## 7  -0.15252 -0.25647 -0.20458 -0.38652  0.792931  0.48033
## 8  -0.14874  0.12938 -0.39866 -0.78891  0.083814  0.27677
## 9  -0.14359  0.11628  0.95439 -0.04629  0.609992 -0.64671
## 10 -0.74154 -0.65088  0.87072 -1.17151 -0.875610 -1.73263
## 11  0.77571 -0.48314  0.02321 -0.76003  0.259283 -0.01494
## 12 -0.44387 -0.28442  0.54938 -0.01813  1.062736 -0.71676
## 13 -0.25608 -0.61494 -0.47847  0.74269  0.180828 -0.14346

```

```

## 14 -0.23636 0.67472 -0.87185 0.72759 0.046993 -0.61306
## 15 -0.75508 0.19781 0.41035 0.89449 -0.744271 -0.25040
## 16 0.97946 1.50039 0.74911 0.12808 -0.657115 -0.30363
## 17 -1.19679 -0.18091 -0.90174 0.16093 -0.229189 0.70454
## 18 0.23197 0.39183 -0.42932 -0.82808 -0.242760 0.64991
## 19 0.80868 -0.55920 -0.22435 0.32426 -0.445610 -0.51746
## 20 -0.19206 0.12003 0.24728 0.99541 0.747110 0.40576
## 21 -0.38477 0.09371 -0.03632 1.07969 1.383306 -0.25819
## 22 0.02352 0.34014 0.07462 0.03662 -0.352806 0.35626
## 23 0.18716 -0.91860 -0.02126 -0.69548 0.117329 0.60753
## 24 0.03851 0.57991 -0.70662 -1.10691 0.533700 -0.10095
## 25 0.61401 -0.01017 -0.51875 0.25795 -1.076732 -0.24901
## 26 0.42275 0.07490 0.99168 0.37950 -0.249480 0.19511
## 27 0.54478 -0.12356 0.31406 0.41569 0.237094 0.09772
## 28 0.02044 0.43106 -0.71781 0.33274 -0.428303 -0.26278
## 29 1.23235 0.15293 0.10463 -0.07285 0.107338 0.07885
## 30 0.02214 -1.33353 -0.37271 0.38762 -0.062960 -0.48751
## 31 -0.71637 0.11204 -0.85950 0.24734 -0.066602 -0.17455
## 32 0.01756 -0.71899 -0.27770 0.01727 0.106473 -0.73172
## 33 0.78296 -0.75806 0.27650 -0.27767 -0.031297 -0.01895
## 34 -0.33590 -0.63554 1.07989 -0.07860 -0.526645 0.97933
## 35 -0.15761 0.49842 0.66426 -0.05091 0.331527 0.71522
## 36 0.53360 0.88375 -0.84613 0.03571 -0.630019 -0.08456
## 37 0.31300 0.02099 -0.05877 -0.57261 0.712218 0.05155
##
##
## Biplot scores for constraining variables
##
##          dbRDA1   dbRDA2   dbRDA3   dbRDA4   dbRDA5   dbRDA6
## T_av      0.63982  0.25527  0.1662 -0.34674 -0.2289 -0.48193
## O2_sat_av  0.02371 -0.02938 -0.8854 -0.18557  0.2035 -0.13356
## Con_av     -0.25315 -0.27391 -0.1407 -0.76998 -0.4264  0.07657
## COD_av     -0.13235  0.15034  0.4807  0.08407  0.1430 -0.57341
## NH4._av    -0.40051  0.14159  0.7569 -0.21438 -0.2051 -0.37005
## Nt_av      -0.31714 -0.23687  0.2952 -0.33830 -0.5347 -0.40917
## pool_riffle -0.34609 -0.15029 -0.4384  0.27287 -0.5547 -0.42271
## meander    -0.33563  0.52703 -0.3800 -0.21165 -0.5422 -0.13159
anova(spe.rda)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_riffle)
##          Df SumOfSqs      F Pr(>F)
## Model      8  1.1664 1.2909  0.177
## Residual  28  3.1624

anova(spe.rda, by = "term")

## Permutation test for dbrda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999

```

```

## 
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_rif
##              Df SumOfSqs      F Pr(>F)
## T_av          1  0.2734 2.4210  0.053 .
## O2_sat_av    1  0.1377 1.2195  0.328
## Con_av        1  0.1613 1.4283  0.223
## COD_av        1  0.0676 0.5990  0.673
## NH4._av       1  0.1709 1.5131  0.191
## Nt_av         1  0.0501 0.4439  0.783
## pool_riffle   1  0.0657 0.5818  0.684
## meander       1  0.2396 2.1210  0.074 .
## Residual     28  3.1624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova.cca(spe.rda, step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_rif
##              Df SumOfSqs      F Pr(>F)
## Model        8  1.1664 1.2909  0.144
## Residual   28  3.1624
anova.cca(spe.rda, step = 1000, by = "term")

## Permutation test for dbrda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_rif
##              Df SumOfSqs      F Pr(>F)
## T_av          1  0.2734 2.4210  0.049 *
## O2_sat_av    1  0.1377 1.2195  0.317
## Con_av        1  0.1613 1.4283  0.241
## COD_av        1  0.0676 0.5990  0.704
## NH4._av       1  0.1709 1.5131  0.190
## Nt_av         1  0.0501 0.4439  0.776
## pool_riffle   1  0.0657 0.5818  0.707
## meander       1  0.2396 2.1210  0.072 .
## Residual     28  3.1624
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.06072755
RsquareAdj(spe.rda)$r.squared

## [1] 0.2694548

```

### 8.1.2 Effect of space on component community structure

```

# Same for spatial predictors
spe.rda <- dbrda(spe.hel_bray ~ netcen + updist, data = env_select)
anova(spe.rda)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Model      2    0.5154 2.2975  0.022 *
## Residual  34   3.8135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.06723433

mod0 <- dbrda(spe.hel_bray ~ 1, env_select[, c(9:10)]) # Model with intercept only #edit_PH
mod1 <- dbrda(spe.hel_bray ~ ., env_select[, c(9:10)]) # Model with all explanatory variables #edit_P
step.res <- ordiR2step(mod0, mod1, direction = "both", perm.max = 200)

## Step: R2.adj= 0
## Call: spe.hel_bray ~ 1
##
##          R2.adjusted
## <All variables> 0.06723433
## + updist        0.04867127
## + netcen        0.04317133
## <none>          0.00000000
##
##          Df      AIC      F Pr(>F)
## + updist  1 54.314 2.8418  0.02 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.04867127
## Call: spe.hel_bray ~ updist
##
##          R2.adjusted
## <All variables> 0.06723433
## + netcen        0.06723433
## <none>          0.04867127
##
##          Df      AIC      F Pr(>F)
## + netcen  1 54.512 1.6965  0.152
step.res$anova # Summary table

##          R2.adj Df      AIC      F Pr(>F)
## + updist     0.048671  1 54.314 2.8418  0.02 *
## <All variables> 0.067234
## ---

```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.06723433

anova.cca(spe.rda, step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Model      2    0.5154 2.2975   0.03 *
## Residual  34    3.8135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova.cca(spe.rda, step = 1000, by = "term")

## Permutation test for dbrda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## netcen     1    0.3019 2.6920   0.037 *
## updist     1    0.2134 1.9029   0.110
## Residual  34    3.8135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.06723433

RsquareAdj(spe.rda)$r.squared

## [1] 0.1190546

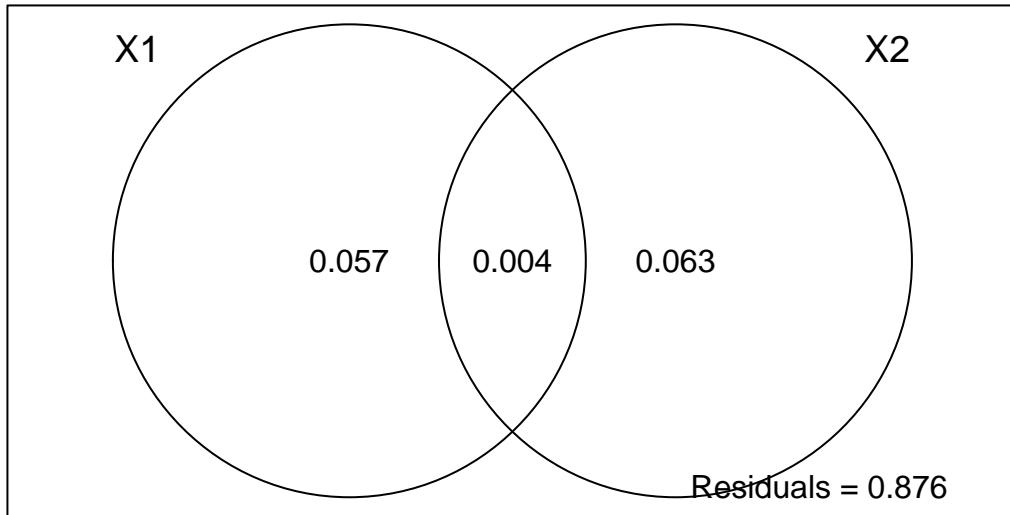
```

### 8.1.3 Variation partitioning

```

# Variation partitioning
spe.varpart1 <- varpart(spe.hel_bray, env_select[, 1:8], env_select[, 9:10])
plot(spe.varpart1, digits = 2)

```



```
spe.varpart1
```

```
##
## Partition of squared Bray distance in dbRDA
##
## Call: varpart(Y = spe.hel_bray, X = env_select[, 1:8], env_select[, 9:10])
##
## Explanatory tables:
## X1: env_select[, 1:8]
## X2: env_select[, 9:10]
##
## No. of explanatory tables: 2
## Total variation (SS): 4.3288
## No. of observations: 37
##
## Partition table:
##                               Df R.squared Adj.R.squared Testable
## [a+c] = X1                 8  0.26945     0.06073    TRUE
## [b+c] = X2                 2  0.11905     0.06723    TRUE
## [a+b+c] = X1+X2            10  0.36716     0.12376    TRUE
## Individual fractions
## [a] = X1|X2                8           0.05653    TRUE
## [b] = X2|X1                2           0.06303    TRUE
## [c]                         0           0.00420   FALSE
## [d] = Residuals             0           0.87624   FALSE
```

```

## ---
## Use function 'dbrda' to test significance of fractions of interest
anova.cca(dbrda(spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av +
    pool_riffle + meander + Condition(netcen + updist), data = env_select), step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_riffle + meander + Condition(netcen + updist), data = env_select), step = 1000

## Df SumOfSqs      F Pr(>F)
## Model     8   1.0740 1.2742  0.174
## Residual 26   2.7395

anova.cca(dbrda(spe.hel_bray ~ netcen + updist + Condition(T_av + O2_sat_av + Con_av +
    COD_av + NH4._av + Nt_av + pool_riffle + meander), data = env_select), step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = spe.hel_bray ~ netcen + updist + Condition(T_av + O2_sat_av + Con_av + COD_av +
    COD_av + NH4._av + Nt_av + pool_riffle + meander), data = env_select), step = 1000

## Df SumOfSqs      F Pr(>F)
## Model     2   0.42295 2.0071  0.065 .
## Residual 26   2.73945
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

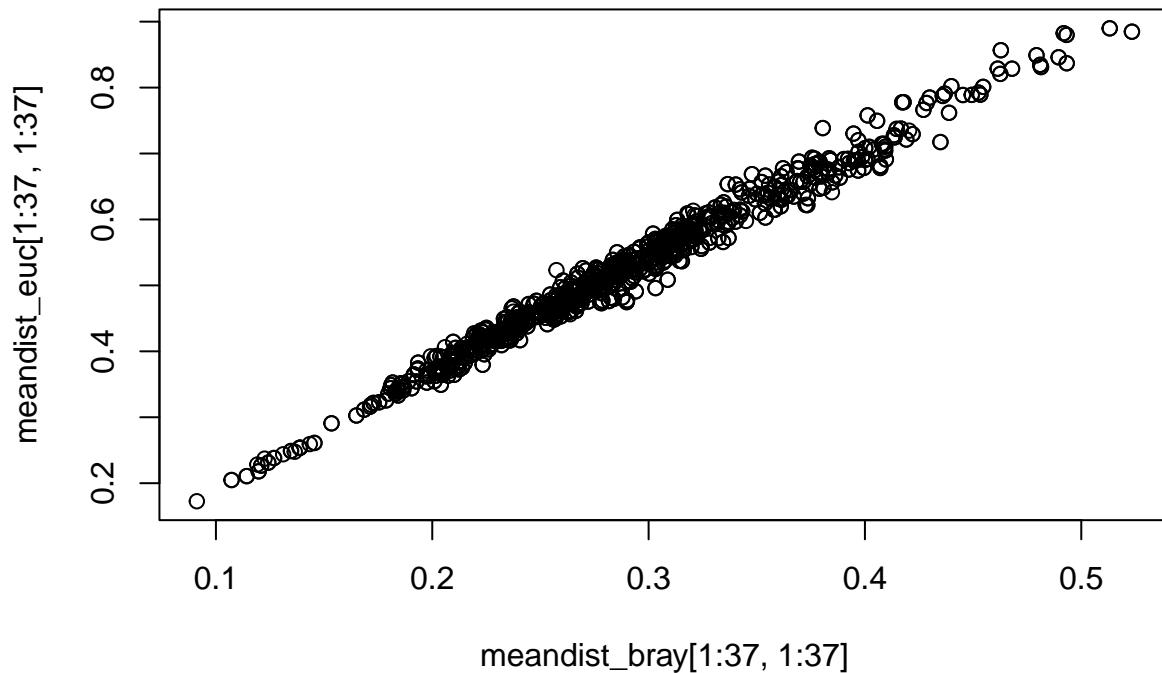
## 8.2 Infra-communities

```

# Infracommunities: Bray-Curtis dissimilarities are calculated at the
# individual host level Hellinger-transformed parasite data and then averaged
# within site A dummy parasite species is added to avoid problems with
# non-infected fishes
data_infra <- na.omit(data[, c(1, 22:24, 26:32)])
data_infra_disp <- dispweight(data_infra[, -1])
braycurtis <- vegdist(decostand(cbind(data_infra_disp, rep(1, nrow(data_infra)))), 
    na.rm = T, method = "hellinger"), method = "bray", na.rm = T)
meandist_bray <- meandist(braycurtis, data_infra[, 1])

# Check whether Euclidean and Bray-Curtis distances are comparable
braycurtis <- vegdist(decostand(cbind(data_infra_disp, rep(1, nrow(data_infra)))), 
    na.rm = T, method = "hellinger"), method = "bray", na.rm = T)
meandist_bray <- meandist(braycurtis, data_infra[, 1])
euc <- vegdist(decostand(cbind(data_infra_disp, rep(1, nrow(data_infra)))), na.rm = T,
    method = "hellinger"), method = "euc", na.rm = T)
meandist_euc <- meandist(euc, data_infra[, 1])
plot(meandist_bray[1:37, 1:37], meandist_euc[1:37, 1:37])

```



```

mantel(meandist_bray[1:37, 1:37], meandist_euc[1:37, 1:37])

##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = meandist_bray[1:37, 1:37], ydis = meandist_euc[1:37, 1:37])
##
## Mantel statistic r: 0.9906
##      Significance: 0.001
##
## Upper quantiles of permutations (null model):
##    90%   95% 97.5%   99%
## 0.187 0.242 0.283 0.351
## Permutation: free
## Number of permutations: 999
model_adonis = adonis(meandist_bray ~ avlength + avcondition + T_av + O2_sat_av +
  Con_av + COD_av + NH4_.av + Nt_av + pool_riffle + meander + netcen + updist,
  data = environment2)

## 'adonis' will be deprecated: use 'adonis2' instead
model_adonis$aov.tab

##
## Permutation: free
## Number of permutations: 999
##

```

```

## Terms added sequentially (first to last)
##
##          Df SumsOfSqs  MeanSqs F.Model       R2 Pr(>F)
## avlength     1  0.05123  0.051226  6.1160 0.11570  0.008 **
## avcondition  1  0.00393  0.003927  0.4688 0.00887  0.609
## T_av         1  0.00874  0.008741  1.0437 0.01974  0.392
## O2_sat_av   1  0.02116  0.021158  2.5262 0.04779  0.084 .
## Con_av       1  0.05861  0.058606  6.9971 0.13237  0.007 **
## COD_av       1  0.03543  0.035428  4.2298 0.08002  0.027 *
## NH4._av      1  0.00275  0.002754  0.3288 0.00622  0.685
## Nt_av         1  0.01049  0.010488  1.2521 0.02369  0.310
## pool_riffle  1  0.00491  0.004914  0.5867 0.01110  0.565
## meander       1  0.00959  0.009588  1.1447 0.02166  0.369
## netcen        1  0.02319  0.023187  2.7684 0.05237  0.075 .
## updist        1  0.01171  0.011707  1.3977 0.02644  0.266
## Residuals    24  0.20102  0.008376           0.45403
## Total         36  0.44274                   1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

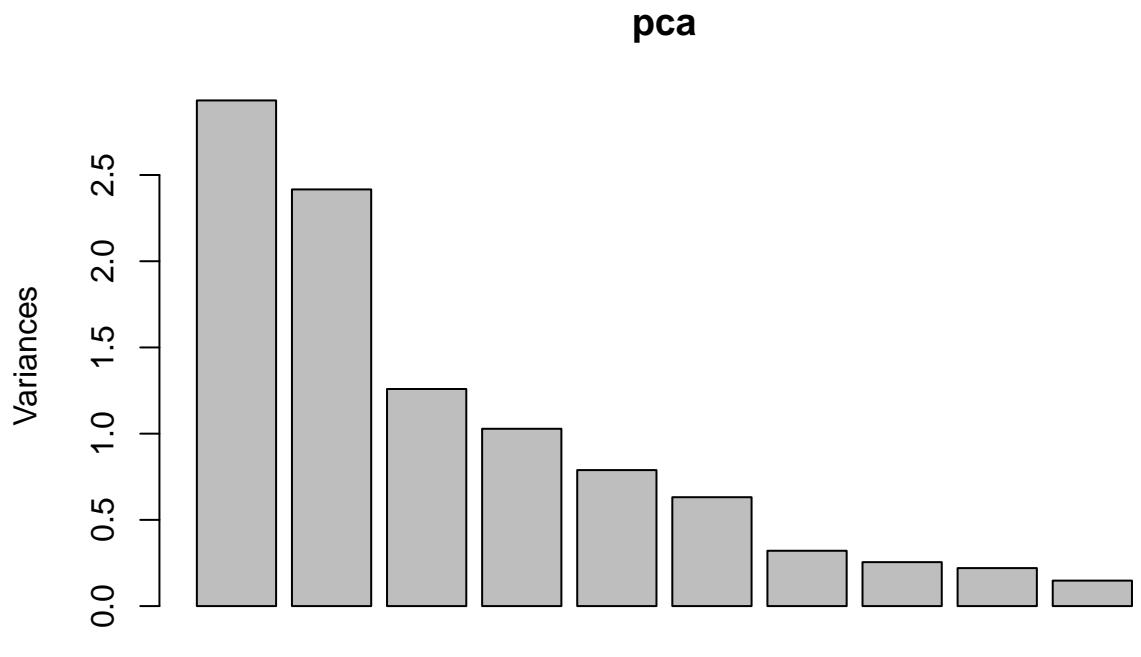
# environmental variables
env_select <- environment2[, c("T_av", "O2_sat_av", "Con_av", "COD_av", "NH4._av",
  "Nt_av", "pool_riffle", "meander", "netcen", "updist")]
env_select$pool_riffle <- as.numeric(env_select$pool_riffle)
env_select$meander <- as.numeric(env_select$meander)

pca <- prcomp(env_select, scale. = T)
summary(pca)

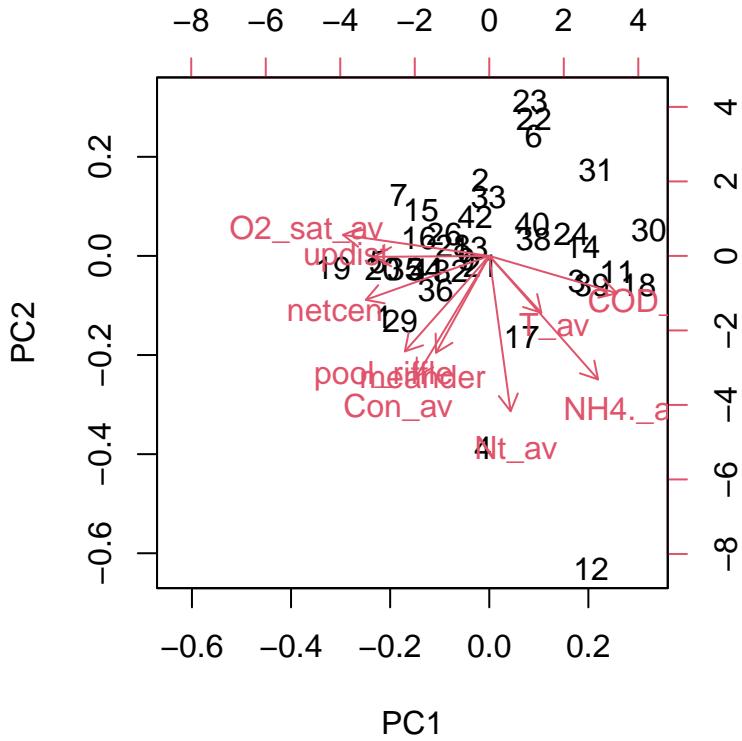
## Importance of components:
##                  PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation 1.7124 1.5545 1.1221 1.0140 0.88807 0.79463 0.56647
## Proportion of Variance 0.2933 0.2416 0.1259 0.1028 0.07887 0.06314 0.03209
## Cumulative Proportion 0.2933 0.5349 0.6608 0.7636 0.84248 0.90563 0.93771
##                  PC8     PC9     PC10
## Standard deviation 0.50483 0.46939 0.38429
## Proportion of Variance 0.02549 0.02203 0.01477
## Cumulative Proportion 0.96320 0.98523 1.00000

plot(pca)

```



```
biplot(pca)
```



### 8.2.1 Effect of environment on infracommunity structure

```
# Assess the effect of environmental variables on parasite infracommunity
# dissimilarities using distance based RDA
spe.rda <- dbRDA(meandist_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av +
  pool_riffle + meander, data = env_select)
anova(spe.rda)

## Permutation test for dbRDA under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbRDA(formula = meandist_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_r
##               Df SumOfSqs      F Pr(>F)
## Model      8  0.45036 1.3833  0.007 ***
## Residual  28  1.13946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.07850103

mod0 <- dbRDA(meandist_bray ~ 1, env_select) # Model with intercept only #edit_PH
mod1 <- dbRDA(meandist_bray ~ ., env_select) # Model with all explanatory variables #edit_PH
step.res <- ordiR2step(mod0, mod1, direction = "both", perm.max = 200)
```

```

## Step: R2.adj= 0
## Call: meandist_bray ~ 1
##
##          R2.adjusted
## <All variables> 0.0898822750
## + Con_av         0.0492715387
## + NH4._av        0.0376899022
## + Nt_av          0.0353578268
## + COD_av         0.0097867139
## + updist         0.0092890919
## + pool_riffle    0.0070398050
## + netcen          0.0034499960
## + O2_sat_av      0.0031240320
## <none>           0.0000000000
## + T_av            -0.0005031246
## + meander         -0.0037253892
##
##          Df      AIC      F Pr(>F)
## + Con_av   1 17.228 2.8657  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.04927154
## Call: meandist_bray ~ Con_av
##
##          R2.adjusted
## <All variables> 0.08988227
## + COD_av         0.08426096
## + NH4._av        0.07058808
## + updist         0.06800179
## + O2_sat_av      0.06266812
## + netcen          0.05455352
## + meander         0.05428744
## + Nt_av           0.05273614
## <none>           0.04927154
## + pool_riffle    0.04923534
## + T_av            0.04694754
##
##          Df      AIC      F Pr(>F)
## + COD_av   1 16.768 2.3373  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.08426096
## Call: meandist_bray ~ Con_av + COD_av
##
##          R2.adjusted
## + updist         0.09961037
## + meander         0.09248673
## <All variables> 0.08988227
## + netcen          0.08626581
## + pool_riffle    0.08559576
## <none>           0.08426096
## + T_av            0.08115910

```

```

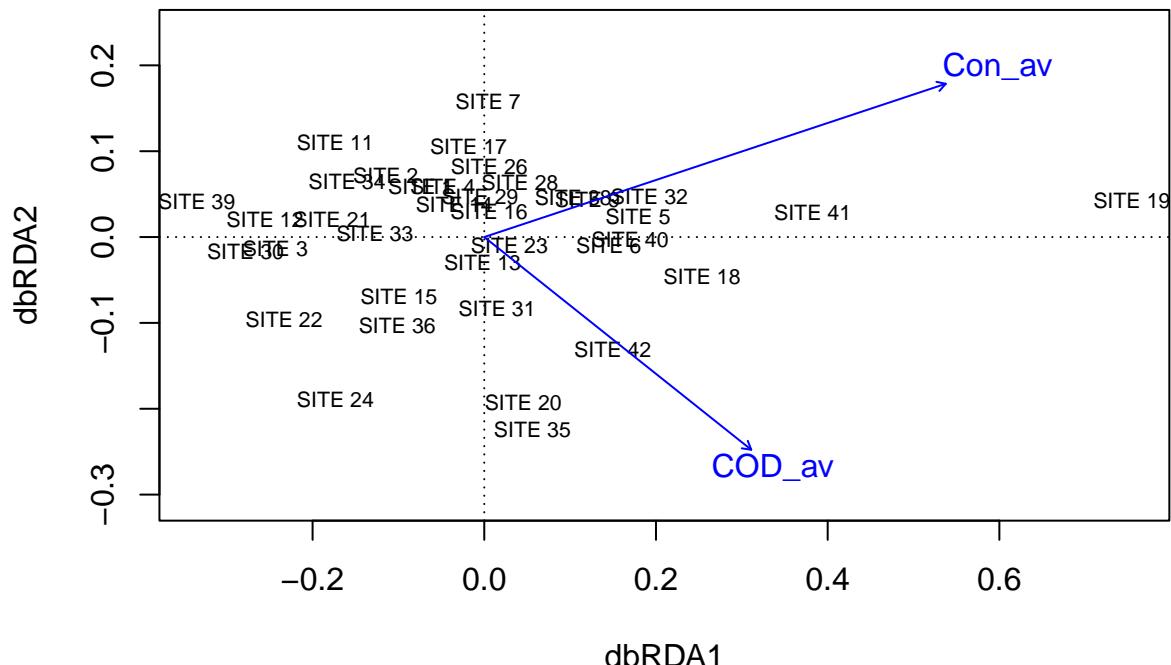
## + O2_sat_av      0.08092067
## + Nt_av          0.07979657
## + NH4._av        0.07591794

step.res$anova # Summary table

##              R2.adj Df     AIC      F Pr(>F)
## + Con_av      0.049272 1 17.228 2.8657 0.002 **
## + COD_av      0.084261 1 16.768 2.3373 0.002 **
## <All variables> 0.089882
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

spe.rda <- dbRDA(meandist_bray ~ Con_av + COD_av, env_select)
plot(spe.rda, scaling = 1) # it is for technical reasons not possible to plot both site and species sc

```



```

summary(spe.rda)

##
## Call:
## dbRDA(formula = meandist_bray ~ Con_av + COD_av, data = env_select)
##
## Partitioning of squared Unknown distance:
##                  Inertia Proportion
## Total           1.5898    1.0000
## Constrained    0.2148    0.1351
## Unconstrained  1.3750    0.8649
## 

```

```

## Eigenvalues, and their contribution to the squared Unknown distance
##
## Importance of components:
##          dbRDA1   dbRDA2    MDS1     MDS2     MDS3     MDS4     MDS5
## Eigenvalue      0.1863  0.02858  0.1972  0.15873  0.1207  0.08661  0.07076
## Proportion Explained 0.1172  0.01798  0.1240  0.09984  0.0759  0.05447  0.04451
## Cumulative Proportion    NA      NA      NA      NA      NA      NA      NA
##                           MDS6     MDS7     MDS8     MDS9     MDS10    MDS11    MDS12
## Eigenvalue      0.06231 0.04802 0.04613 0.04387 0.04231 0.03826 0.03703
## Proportion Explained 0.03919 0.03021 0.02901 0.02760 0.02661 0.02406 0.02329
## Cumulative Proportion    NA      NA      NA      NA      NA      NA      NA
##                           MDS13    MDS14    MDS15    MDS16    MDS17    MDS18    MDS19
## Eigenvalue      0.03557 0.03355 0.03200 0.02808 0.02797 0.02575 0.02487
## Proportion Explained 0.02237 0.02110 0.02013 0.01766 0.01759 0.01620 0.01564
## Cumulative Proportion    NA      NA      NA      NA      NA      NA      NA
##                           MDS20    MDS21    MDS22    MDS23    MDS24    MDS25    MDS26
## Eigenvalue      0.02377 0.02238 0.02182 0.02104 0.01990 0.01880 0.01782
## Proportion Explained 0.01495 0.01408 0.01372 0.01323 0.01252 0.01183 0.01121
## Cumulative Proportion    NA      NA      NA      NA      NA      NA      NA
##                           MDS27    MDS28    MDS29    MDS30    MDS31    MDS32
## Eigenvalue      0.01719 0.01622 0.011778 0.010499 0.007903 0.006958
## Proportion Explained 0.01081 0.01020 0.007408 0.006604 0.004971 0.004376
## Cumulative Proportion    NA      NA      NA      NA      NA      NA
##                           MDS33    iMDS1
## Eigenvalue      0.005670 -0.006426
## Proportion Explained 0.003566 0.004042
## Cumulative Proportion    NA      NA
##
## Accumulated constrained eigenvalues
## Importance of components:
##          dbRDA1   dbRDA2
## Eigenvalue      0.1863  0.02858
## Proportion Explained 0.8670  0.13304
## Cumulative Proportion 0.8670  1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 2.750505
##
## Site scores (weighted sums of species scores)
##
##          dbRDA1   dbRDA2    MDS1     MDS2     MDS3     MDS4
## SITE 1  -0.215744  0.44381  0.01302  0.31819 -0.28116 -0.178171
## SITE 11 -0.507248  0.82144 -0.09719 -0.28198 -0.04679  0.342258
## SITE 12 -0.746029  0.15775  0.31672  0.25334 -0.08234 -0.151368
## SITE 13 -0.005847 -0.21986 -0.02027  0.91847 -0.12233 -0.332031
## SITE 14 -0.101987  0.28502  0.01314  0.55873 -0.09066 -0.223950
## SITE 15 -0.290401 -0.51714 -0.23614 -0.10408 -0.36670 -0.263022
## SITE 16  0.016467  0.22318  0.14693 -0.09879 -0.21010  0.156923
## SITE 17 -0.053594  0.78979  0.68552  0.22623  0.55763  0.068555
## SITE 18  0.741297 -0.34140  0.34156 -0.78781 -0.33039  0.372025
## SITE 19  2.204147  0.31903  0.07576 -0.77176 -0.32619  0.237586

```

```

## SITE 2 -0.334822 0.53901 0.20181 0.23760 0.25255 0.460075
## SITE 20 0.132501 -1.43824 -0.36307 -0.15683 0.91526 0.652586
## SITE 21 -0.518164 0.15765 0.51672 0.22690 -0.05180 0.071838
## SITE 22 -0.681351 -0.71637 0.44536 0.61813 0.24285 -0.440713
## SITE 23 0.085803 -0.07384 -0.18742 -0.58764 -0.50280 0.146049
## SITE 24 -0.506368 -1.41199 -0.02580 0.53865 0.25884 0.040947
## SITE 26 0.017028 0.61108 0.37526 -0.36715 -0.10492 0.425754
## SITE 28 0.121498 0.47609 0.59944 -0.22080 -0.07107 0.465237
## SITE 29 -0.014961 0.35365 0.04914 -0.10076 -0.08378 0.641604
## SITE 3 -0.711132 -0.09645 0.08469 -0.08718 -0.21517 0.038217
## SITE 30 -0.812113 -0.12175 0.55582 -0.16918 -0.25866 0.087617
## SITE 31 0.043616 -0.61571 -0.31459 0.47663 -0.16031 -0.651535
## SITE 32 0.562390 0.35603 -0.38737 0.06837 -0.20621 -0.072343
## SITE 33 -0.372173 0.03501 0.10319 0.85337 -0.01445 -0.328696
## SITE 34 -0.466651 0.48378 0.50738 0.02349 -0.08407 0.306511
## SITE 35 0.163917 -1.66724 -0.04520 0.22652 0.57157 0.198776
## SITE 36 -0.296043 -0.76361 -0.59812 0.40106 -0.04152 -0.249681
## SITE 38 0.302702 0.34509 -0.16274 -0.49724 -0.25040 -0.005416
## SITE 39 -0.978171 0.31142 0.36167 0.40699 0.02355 0.020399
## SITE 4 -0.141634 0.44343 -0.02774 -0.11080 -0.07834 0.180628
## SITE 40 0.496377 -0.01681 -0.38350 -0.04028 -0.50240 -0.167352
## SITE 41 1.117637 0.21808 -0.82799 -0.60061 -0.49878 -0.637014
## SITE 42 0.437517 -0.97421 -1.46550 0.08641 1.33505 0.880493
## SITE 5 0.522987 0.17736 0.27513 0.18757 -0.04206 0.152978
## SITE 6 0.423826 -0.07352 -0.91744 -0.10155 -0.42928 -0.789691
## SITE 7 0.013170 1.17661 0.57227 -1.15454 1.52741 -1.616432
## SITE 9 0.351552 0.32386 -0.18044 -0.38769 -0.23204 0.160360
##
##
## Site constraints (linear combinations of constraining variables)
##
##          dbRDA1   dbRDA2     MDS1     MDS2     MDS3     MDS4
## SITE 1  -0.079483  0.44275  0.01302  0.31819 -0.28116 -0.178171
## SITE 11 -0.692558  0.69501 -0.09719 -0.28198 -0.04679  0.342258
## SITE 12 -0.387516 -0.29270  0.31672  0.25334 -0.08234 -0.151368
## SITE 13  0.692033  0.28074 -0.02027  0.91847 -0.12233 -0.332031
## SITE 14  0.273072  0.58517  0.01314  0.55873 -0.09066 -0.223950
## SITE 15 -0.503973 -0.49304 -0.23614 -0.10408 -0.36670 -0.263022
## SITE 16 -0.041630  0.31054  0.14693 -0.09879 -0.21010  0.156923
## SITE 17  0.286975  0.22462  0.68552  0.22623  0.55763  0.068555
## SITE 18  0.355899 -0.83587  0.34156 -0.78781 -0.33039  0.372025
## SITE 19  1.702755 -0.40556  0.07576 -0.77176 -0.32619  0.237586
## SITE 2 -0.068404  0.54101  0.20181  0.23760  0.25255  0.460075
## SITE 20  0.005618 -0.62143 -0.36307 -0.15683  0.91526  0.652586
## SITE 21 -0.085484 -0.08616  0.51672  0.22690 -0.05180  0.071838
## SITE 22 -0.015359 -0.63175  0.44536  0.61813  0.24285 -0.440713
## SITE 23 -0.475351 -0.06131 -0.18742 -0.58764 -0.50280  0.146049
## SITE 24 -0.138656 -0.63048 -0.02580  0.53865  0.25884  0.040947
## SITE 26 -0.086478  0.70999  0.37526 -0.36715 -0.10492  0.425754
## SITE 28  0.227529  0.47771  0.59944 -0.22080 -0.07107  0.465237
## SITE 29 -0.053389  0.16940  0.04914 -0.10076 -0.08378  0.641604
## SITE 3 -0.805011 -0.44279  0.08469 -0.08718 -0.21517  0.038217
## SITE 30 -0.711357 -0.97402  0.55582 -0.16918 -0.25866  0.087617
## SITE 31  0.197349 -0.41392 -0.31459  0.47663 -0.16031 -0.651535

```

```

## SITE 32  0.342870  0.50733 -0.38737  0.06837 -0.20621 -0.072343
## SITE 33  0.260419  0.25833  0.10319  0.85337 -0.01445 -0.328696
## SITE 34 -0.213627  0.34583  0.50738  0.02349 -0.08407  0.306511
## SITE 35  0.299656 -0.71657 -0.04520  0.22652  0.57157  0.198776
## SITE 36 -0.265626 -0.39128 -0.59812  0.40106 -0.04152 -0.249681
## SITE 38 -0.177490  0.11612 -0.16274 -0.49724 -0.25040 -0.005416
## SITE 39 -0.502932  0.21901  0.36167  0.40699  0.02355  0.020399
## SITE 4   -0.213933  0.31087 -0.02774 -0.11080 -0.07834  0.180628
## SITE 40  0.157625 -0.02847 -0.38350 -0.04028 -0.50240 -0.167352
## SITE 41  0.308647  0.03805 -0.82799 -0.60061 -0.49878 -0.637014
## SITE 42 -0.043511  0.13560 -1.46550  0.08641  1.33505  0.880493
## SITE 5   0.774214 -0.17770  0.27513  0.18757 -0.04206  0.152978
## SITE 6   -0.221478  0.23059 -0.91744 -0.10155 -0.42928 -0.789691
## SITE 7   -0.080948  0.35692  0.57227 -1.15454  1.52741 -1.616432
## SITE 9   -0.020466  0.24745 -0.18044 -0.38769 -0.23204  0.160360
##
##
## Biplot scores for constraining variables
##
##          dbRDA1  dbRDA2 MDS1  MDS2  MDS3  MDS4
## Con_av  0.7628  0.6467    0    0    0    0
## COD_av  0.4413 -0.8974    0    0    0    0
anova(spe.rda)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ Con_av + COD_av, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Model      2  0.21484 2.6563  0.001 ***
## Residual  34  1.37498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(spe.rda, by = "term")

## Permutation test for dbrda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ Con_av + COD_av, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Con_av     1  0.12032 2.9752  0.001 ***
## COD_av     1  0.09452 2.3373  0.004 **
## Residual  34  1.37498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 8.2.2 Effect of space on infracommunity structure

```

spe.rda <- dbrda(meandist_bray ~ netcen + updist, data = env_select)
anova(spe.rda)
```

```

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Model      2  0.10742 1.2319  0.135
## Residual  34  1.48240
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.01271734

anova.cca(spe.rda, step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## Model      2  0.10742 1.2319  0.158
## Residual  34  1.48240
anova.cca(spe.rda, step = 1000, by = "term")

## Permutation test for dbrda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ netcen + updist, data = env_select)
##          Df SumOfSqs      F Pr(>F)
## netcen     1  0.04949 1.1352  0.253
## updist     1  0.05792 1.3285  0.124
## Residual  34  1.48240
RsquareAdj(spe.rda)$adj.r.squared

## [1] 0.01271734

RsquareAdj(spe.rda)$r.squared

## [1] 0.06756637

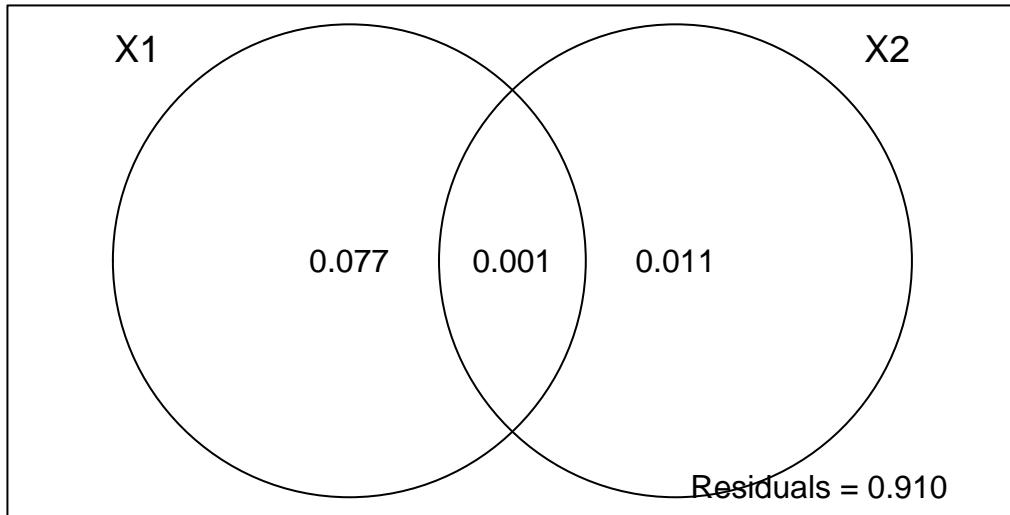
```

### 8.2.3 Variation partitioning

```

# Variation partitioning
spe.varpart1 <- varpart(meandist_bray, env_select[, 1:8], env_select[, 9:10])
plot(spe.varpart1, digits = 2)

```



```
spe.varpart1
```

```
##
## Partition of squared Unknown user-supplied distance in dbRDA
##
## Call: varpart(Y = meandist_bray, X = env_select[, 1:8], env_select[, 9:10])
##
## Explanatory tables:
## X1: env_select[, 1:8]
## X2: env_select[, 9:10]
##
## No. of explanatory tables: 2
## Total variation (SS): 1.5898
## No. of observations: 37
##
## Partition table:
##                               Df R.squared Adj.R.squared Testable
## [a+c] = X1                 8  0.28328     0.07850    TRUE
## [b+c] = X2                 2  0.06757     0.01272    TRUE
## [a+b+c] = X1+X2            10 0.34269     0.08988    TRUE
## Individual fractions
## [a] = X1|X2                8           0.07716    TRUE
## [b] = X2|X1                2           0.01138    TRUE
## [c]                         0           0.00134   FALSE
## [d] = Residuals             0           0.91012   FALSE
```

```

## ---
## Use function 'dbrda' to test significance of fractions of interest
anova.cca(dbrda(meandist_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av +
    pool_riffle + meander + Condition(netcen + updist), data = env_select), step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_riffle + meander + Condition(netcen + updist), data = env_select), step = 1000)
## Df SumOfSqs      F Pr(>F)
## Model     8   0.4374 1.3603  0.016 *
## Residual 26   1.0450
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova.cca(dbrda(meandist_bray ~ netcen + updist + Condition(T_av + O2_sat_av + Con_av +
    COD_av + NH4._av + Nt_av + pool_riffle + meander), data = env_select), step = 1000)

## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = meandist_bray ~ netcen + updist + Condition(T_av + O2_sat_av + Con_av + COD_av + NH4._av + Nt_av + pool_riffle + meander), data = env_select), step = 1000)
## Df SumOfSqs      F Pr(>F)
## Model     2   0.09446 1.1751  0.183
## Residual 26   1.04500

```