

MEE Paper Analysis

Fit msocc model

```
# prep data frames
site.df <- goby %>%
  distinct(site, .keep_all = TRUE) %>%
  select(-c(pcr1:pcr6), -sample)
sample.df <- goby %>%
  select(-c(pcr1:pcr6))
detect.df <- goby %>%
  select(-c(twg:veg))

start_msocc <- Sys.time()
# fit model
goby_mod <- msocc_mod(detect.df, num.mcmc = num_mcmc,
  site = list(model = ~ veg, cov_tbl = site.df),
  sample = list(model = ~ sal + twg, cov_tbl = sample.df),
  rep = list(model = ~ sal + fish + turb, cov_tbl = sample.df),
  priors = list(site = list(mu0 = 0, Sigma0 = 4),
    sample = list(mu0 = 0, Sigma0 = 4),
    rep = list(mu0 = 0, Sigma0 = 4),
    a0 = 1, b0 = 1),
  progress = F, print = NULL, beta_bin = T) # , seed = 02042020
end_msocc <- Sys.time()
end_msocc - start_msocc

# posterior summary
head(posterior_summary(goby_mod, level = "overall", burnin = 1000))
head(posterior_summary(goby_mod, level = "sample", burnin = 1000))
cred_plot(goby_mod, level = "sample", n = 6)[[1]]
```

Fit eDNAoccupancy model

```
gobyDetections = occData(gobyDetectionData, siteColName = "site", sampleColName = "sample")
# set.seed(02042020)

start_edna <- Sys.time()
fit = occModel(formulaSite = ~ veg,
  formulaSiteAndSample = ~ sal + twg,
  formulaReplicate = ~ sal + fish + turb,
  detectionMats = gobyDetections,
  siteData = gobySurveyData,
  niter = num_mcmc,
  niterInterval = 2000,
  siteColName = 'site')
end_edna <- Sys.time()
```

```
end_edna - start_edna
```

```
# posterior summary
posteriorSummary(fit, burnin = 1000, mcError = TRUE)
plotTrace(fit, c('beta.(Intercept)'), burnin = 1000)
```

Scale data

```
# for eDNAoccupancy
gobyDetections <- occData(gobyDetectionData, siteColName = "site", sampleColName = "sample")
gobySurveyData.sc <- scaleData(gobySurveyData)
survey_data <- gobySurveyData.sc %>%
  as_tibble() %>%
  mutate(site = as.character(site)) %>%
  mutate(site = ifelse(site == "Gannon_Slough_", "Gannon_Slough",
                        ifelse(site == "Martin_Slough_", "Martin_Slough",
                                ifelse(site == "Jughandle_Creek_", "Jughandle_Creek",
                                        ifelse(site == "Big_River_", "Big_River", site)))))

# for msocc
site.df.sc <- site.df %>%
  select(-c(twg:veg)) %>%
  left_join(., survey_data, by = "site")

sample.df.sc <- sample.df %>%
  select(-c(twg:veg)) %>%
  left_join(., survey_data, by = "site")
```

Loop - unscaled

```
# setup
num_mcmc <- 11000
nsims <- 10

ess <- array(0, dim = c(nsims, 9, 2))
time <- matrix(0, nsims, 4)

for(i in 1:nsims){
  # msocc
  start_msocc <- Sys.time()
  # fit model
  goby_mod <- msocc_mod(detect.df, num.mcmc = num_mcmc,
                        site = list(model = ~ veg, cov_tbl = site.df),
                        sample = list(model = ~ sal + twg, cov_tbl = sample.df),
                        rep = list(model = ~ sal + fish + turb, cov_tbl = sample.df),
                        priors = list(site = list(mu0 = 0, Sigma0 = 4),
                                      sample = list(mu0 = 0, Sigma0 = 4),
                                      rep = list(mu0 = 0, Sigma0 = 4),
                                      a0 = 1, b0 = 1),
                        progress = F, print = NULL, beta_bin = T)
  end_msocc <- Sys.time()
```

```

time[i,1] <- end_msocc - start_msocc; time[i,2] <- attr(end_msocc - start_msocc, "units")

# eDNA
start_edna <- Sys.time()
fit = occModel(formulaSite = ~ veg,
               formulaSiteAndSample = ~ sal + twg,
               formulaReplicate = ~ sal + fish + turb,
               detectionMats = gobyDetections,
               siteData = gobySurveyData,
               niter = num_mcmc,
               niterInterval = 500,
               siteColName = 'site')
end_edna <- Sys.time()
time[i,3] <- end_edna - start_edna; time[i,4] <- attr(end_edna - start_edna, "units")

# ess
msocc_post <- cbind(goby_mod$beta, goby_mod$alpha, goby_mod$delta) %>%
  as.mcmc

edna_post <- read.csv("mc.csv") %>%
  as.matrix %>%
  as.mcmc

ess[i,,1] <- effectiveSize(msocc_post)
ess[i,,2] <- effectiveSize(edna_post)

print(i)
}

# save results
save(time, file = "time_unscaled.Rdata")
save(ess, file = "ess_unscaled.Rdata")

```

Loop - scaled

```

# setup
num_mcmc <- 11000
nsims <- 10

ess <- array(0, dim = c(nsims, 9, 2))
time <- matrix(0, nsims, 4)

for(i in 1:nsims){
  # msocc
  start_msocc <- Sys.time()
  # fit model
  goby_mod <- msocc_mod(detect.df, num.mcmc = num_mcmc,
                       site = list(model = ~ veg, cov_tbl = site.df.sc),
                       sample = list(model = ~ sal + twg, cov_tbl = sample.df.sc),
                       rep = list(model = ~ sal + fish + turb, cov_tbl = sample.df.sc),
                       priors = list(site = list(mu0 = 0, Sigma0 = 4),
                                      sample = list(mu0 = 0, Sigma0 = 4)),

```

```

        rep = list(mu0 = 0, Sigma0 = 4),
        a0 = 1, b0 = 1),
    progress = F, print = NULL, beta_bin = T)
end_msocc <- Sys.time()
time[i,1] <- end_msocc - start_msocc; time[i,2] <- attr(end_msocc - start_msocc, "units")

# eDNA
start_edna <- Sys.time()
fit = occModel(formulaSite = ~ veg,
               formulaSiteAndSample = ~ sal + twg,
               formulaReplicate = ~ sal + fish + turb,
               detectionMats = gobyDetections,
               siteData = gobySurveyData.sc,
               niter = num_mcmc,
               niterInterval = 500,
               siteColName = 'site')
end_edna <- Sys.time()
time[i,3] <- end_edna - start_edna; time[i,4] <- attr(end_edna - start_edna, "units")

# ess
msocc_post <- cbind(goby_mod$beta, goby_mod$alpha, goby_mod$delta) %>%
  as.mcmc

edna_post <- read.csv("mc.csv") %>%
  as.matrix %>%
  as.mcmc

ess[i,,1] <- effectiveSize(msocc_post)
ess[i,,2] <- effectiveSize(edna_post)

print(i)
}

# save results
save(time, file = "time_scaled.Rdata")
save(ess, file = "ess_scaled.Rdata")

```

Make figure

```

load("time_unscaled.Rdata")
time_unscaled <- time

load("time_scaled.Rdata")
time_scaled <- time

load("ess_unscaled.Rdata")
ess_unscaled <- ess

load("ess_scaled.Rdata")
ess_scaled <- ess
rm(ess, time)

```

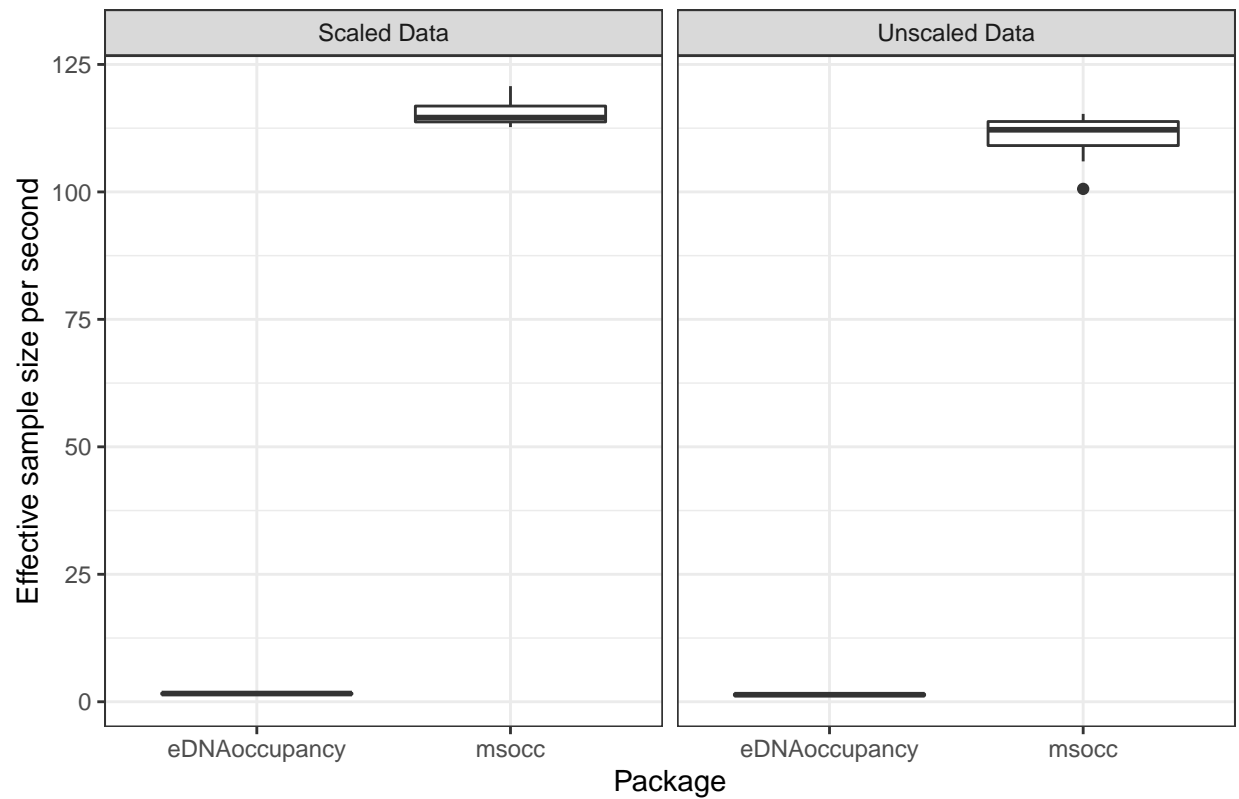
```

# plotting data frame
plot_tbl <- tibble(time = c(
  as.numeric(time_unscaled[,1]),
  as.numeric(time_unscaled[,3]) * 60,
  as.numeric(time_scaled[,1]),
  as.numeric(time_scaled[,3]) * 60
),
  ess = c(
    apply(ess_unscaled[,1], 1, min),
    apply(ess_unscaled[,2], 1, min),
    apply(ess_scaled[,1], 1, min),
    apply(ess_scaled[,2], 1, min)
  ),
  method = rep(c(rep("msocc", 10), rep("eDNAoccupancy", 10)), 2)
) %>%
  mutate(essps = ess / time,
         esspm = essps * 60,
         data_type = c(rep("Unscaled Data", 20), rep("Scaled Data", 20)))

# plot
plot_tbl %>%
  ggplot(aes(x = method, y = essps)) +
  geom_boxplot() +
  facet_wrap(~ data_type) +
  theme_bw() +
  labs(title = "Effective sample size per second",
       x = "Package",
       y = "Effective sample size per second")

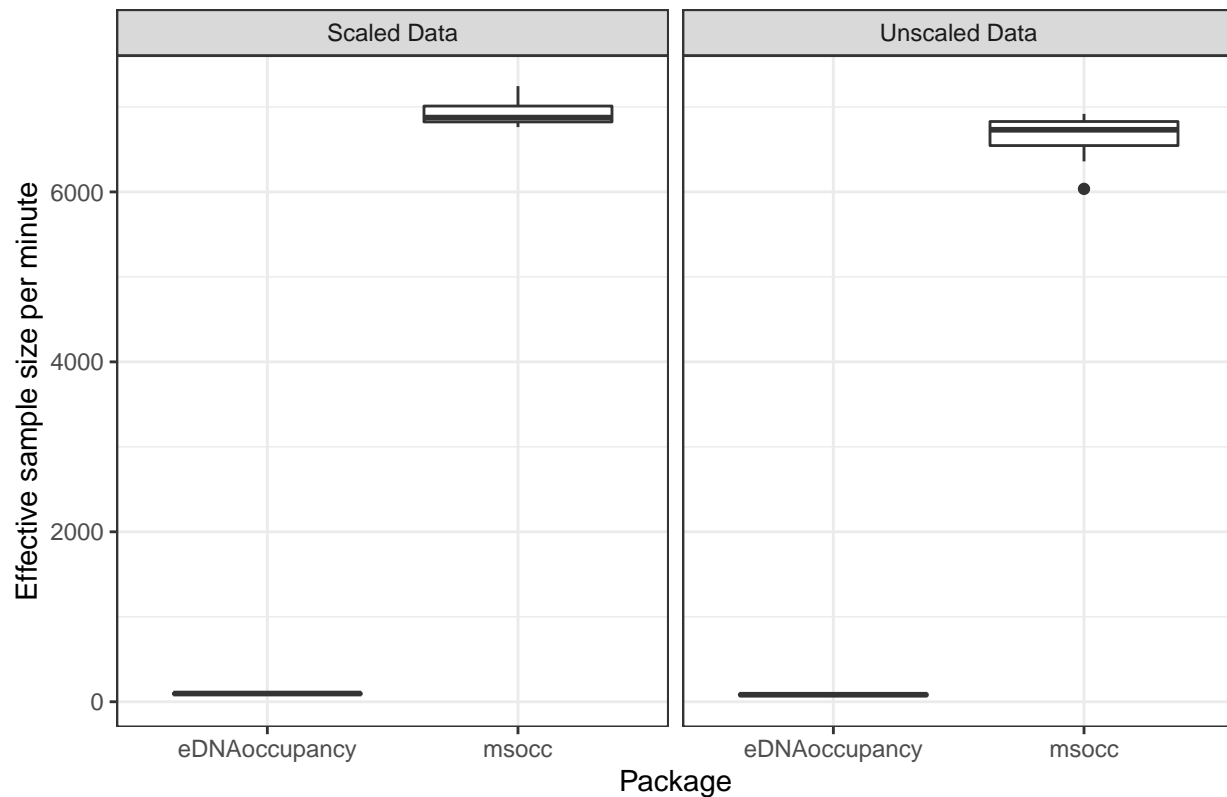
```

Effective sample size per second



```
plot_tbl %>%  
  ggplot(aes(x = method, y = esspm)) +  
  geom_boxplot() +  
  facet_wrap(~ data_type) +  
  theme_bw() +  
  labs(title = "Comparison of effective sample size per minute",  
        x = "Package",  
        y = "Effective sample size per minute")
```

Comparison of effective sample size per minute



```
# create table
table <- plot_tbl %>%
  mutate(time_sec = time,
         time_minutes = time / 60) %>%
  select(-time) %>%
  select(method, data_type, everything()) %>%
  group_by(method, data_type)

with(table, tapply(time_sec, list(method, data_type), mean))

##           Scaled Data Unscaled Data
## eDNAoccupancy 1839.44724   2198.72291
## msocc         36.07299    36.40659

with(table, tapply(time_minutes, list(method, data_type), mean))

##           Scaled Data Unscaled Data
## eDNAoccupancy 30.6574539   36.6453818
## msocc         0.6012165    0.6067765

with(table, tapply(ess, list(method, data_type), mean))

##           Scaled Data Unscaled Data
## eDNAoccupancy 2908.973    2958.186
## msocc         4161.010    4029.611

table_out <- tibble(
  Package = c(rep("msocc", 2), rep("eDNAoccupancy", 2)),
```

```

Data = rep(c("Unscaled", "Scaled"), 2),
`Average time` = c("36.41 seconds", "36.07 seconds", "36.65 minutes", "30.66 minutes"),
`Average ESS` = c(4029.61, 4161.01, 2958.19, 2908.97),
`Average ESS/min` = c(6640.39, 6921.56, 80.71, 94.87)
)

print(xtable(table_out), include.rownames = FALSE)

## % latex table generated in R 3.6.2 by xtable 1.8-4 package
## % Wed Feb 19 11:05:07 2020
## \begin{table}[ht]
## \centering
## \begin{tabular}{lllrr}
## \hline
## Package & Data & Average time & Average ESS & Average ESS/min \\
## \hline
## msocc & Unscaled & 36.41 seconds & 4029.61 & 6640.39 \\
## msocc & Scaled & 36.07 seconds & 4161.01 & 6921.56 \\
## eDNAoccupancy & Unscaled & 36.65 minutes & 2958.19 & 80.71 \\
## eDNAoccupancy & Scaled & 30.66 minutes & 2908.97 & 94.87 \\
## \hline
## \end{tabular}
## \end{table}

```