The Missing Species Problem – A computational Note

2024-01-17

Front Matters

The problem of missing species

- Reference: Fisher et al (1943)
- Data: numbers of individuals observed from S species $(n_1, n_2, \cdots, n_S)$ in a set time
- Question: what is the expected number of "new" species when the same survey is conducted again.
- Fisher's model – a parametric hierarchical model:
  1. Assuming the number of observed individuals are Poisson random variables:

$$n_i \sim Poisson(\lambda_i)$$

  2. Assuming λ_i i.i.d.:

$$\lambda_i \sim gamma(\mu, r)$$

  which leads to predictive distribution of counts of individuals to be negative binomial:

$$n_i \sim NB(n_i \mid \mu, r)$$

  with a probability distribution function

$$\Pr(n_i = k) = \frac{\Gamma(r+k)}{k!\Gamma(r)} \left(\frac{r}{\mu+r}\right)^r \left(\frac{\mu}{\mu+r}\right)^k$$

The observed data (counts of individuals of each species), however, do not include 0. As a result, the likelihood function should be the rescaled probability distribution excluding 0. Given that the probability of observing a 0 is

$$p_0 = \left(\frac{r}{\mu+r}\right)^r,$$

The likelihood function of observing $n_i = k$ should be:

$$\Pr(n_i = k) = \frac{\frac{\Gamma(r+k)}{k!\Gamma(r)} \left(\frac{r}{\mu+r}\right)^r \left(\frac{\mu}{\mu+r}\right)^k}{1-p_0}, \text{ for } k > 0$$

This likelihood function can be easily implemented under Stan:

```
target += neg_binomial_2_log_lpmf(n | log_mu, r) - log1m(p0);
```

Applied to three data sets

Including - Western Basin trawl data

```
datWB_All <- read.csv(paste(dataDir, "WB_Catch.txt", sep="/")) %>% filter(season == "Autumn") %>%
  filter(life_stage == "ALL") %>% filter(count > 0) %>% group_by(year, species)
datWB_All$count <- as.integer(round(datWB_All$count,1))
dat_agAll <- summarize(datWB_All, count=sum(count))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## testing for year 2018
datWB <- group_by(datWB_All, species) %>% filter(year == 2018)
dat_ag <- summarize(datWB, count=sum(count))
y <- sort(dat_ag$count)
n <- length(y)
```

- Shakespeare word type frequency data

```
data (Shakespeare)

datSK <- Shakespeare
nSK <- sum(datSK$n_j)
ySK <- rep(datSK$j, datSK$n_j)

datSK2 <- Shakespeare[1:100,]
nSK2 <- sum(datSK2$n_j)
ySK2 <- rep(datSK2$j, datSK2$n_j)
```

and - the butterfly data from Fisher et al (1943)

```
data("WillButterfly")
datBF_W <- WillButterfly
nBF <- sum(datBF_W$n_j)
yBF <- rep(datBF_W$j, datBF_W$n_j)

data("butterfly")
datBF_C <- butterfly[1:24,]
nBF_C <- sum(datBF_C$n_j)
yBF_C <- rep(datBF_C$j, datBF_C$n_j)

datBF2 <- butterfly
nBF2 <- sum(datBF2$n_j)
yBF2 <- rep(datBF2$j, datBF2$n_j)
```

Stan input function (includes data and initial values)

```
species_count_in <- function(n, y, n.chains=nchains){
  n <- n
  y <- y
  data <- list(n=n, y=y)

  inits <- list()
  for (i in 1:n.chains){
    inits[[i]] <- list(log_mu = rnorm(1), r = runif(1,0.01,1))
```

```
  }
  paras <- c("mu","r","p0","N")
  return(list(data=data, init=inits, nchains=n.chains, para=paras ))
}

species_count_allyrBHM <- function(data=dat_agAll, n.chains=nchains){
  yr <- as.numeric(ordered(data$year))
  y <- data$count
  nsp <- summarize(data, n=length(year))
  data <- list(nobs=length(y), yr=yr, y=y, nyr=max(yr), n=nsp$n)

  inits <- list()
  for (i in 1:n.chains){
    inits[[i]] <- list(zlog_mu = rnorm(max(yr)), r=runif(max(yr)),
                       tau = runif(1), theta=rnorm(1))
  }
  paras <- c("mu","r","p0","N","tau", "theta")
  return(list(data=data, init=inits, nchains=n.chains, para=paras ))
}
```

The Stan Model

```
## one year model

species_count <- "
  data {
    int<lower=1> n;                 // total number of species
    int<lower=0> y[n];              // catch per species
  }

  parameters {
    real log_mu;
    real<lower = 0> r;
  }

  transformed parameters {
    real<lower=0,upper=1> p0;
    real<lower=0> mu;
    mu = exp(log_mu);
    p0 = (r/(mu+r))^r;
  }

  model {
    log_mu ~ normal(0,5);
    r ~ cauchy(0,2);

    target += neg_binomial_2_log_lpmf(y | log_mu, r) - log1m(p0);
  }

  generated quantities {
    real N;         // estimated total number of species
    N = n/(1-p0);
  }
```

```
"
fit <- stan_model(model_code = species_count)     ## compile model

## now multi-year BHM

species_count_nyear2 <- "
  data {
    int<lower=1> nobs;              // number of observations
    int<lower=1> nyr;               // total number of years
    int<lower=1> n[nyr];            // total number of species per year
    int<lower=1> y[nobs];           // catch per species each year
    int<lower=1> yr[nobs];          // year index for y
  }
  parameters {
    real<lower=0> r[nyr];
    real zlog_mu[nyr];
    real<lower=0> tau;
    real theta;
  }
  transformed parameters {
    real<lower=0,upper=1> p0[nyr];
    real<lower=0> mu[nyr];
    real log_mu[nyr];
    for (i in 1:nyr){
      log_mu[i] = theta+tau*zlog_mu[i];
      mu[i] = exp(log_mu[i]);
      p0[i] = (r[i]/(mu[i]+r[i]))^r[i];
    }
  }
  model {
    r ~ cauchy(0,2);
    tau ~ cauchy(0,2);
    theta ~ normal(0,5);
    zlog_mu ~ std_normal();
    for (k in 1:nobs){
      target += neg_binomial_2_log_lpmf(y[k] | log_mu[yr[k]], r[yr[k]]) - log1m(p0[yr[k]]);
    }
  }

  generated quantities {
    real<lower=0> N[nyr];         // estimated total number of species
    for (j in 1:nyr)
    N[j] = n[j]/(1-p0[j]);
  }
"

fit3 <- stan_model(model_code = species_count_nyear2)     ## compile model
save(fit,fit3, file=paste(saveDIR, "stan_models.RData", sep="/"))
```

Examples

the Western Basin trawl data

```
## testing for year 2013-2021

p0_WB <- rv(0)
N0_WB <- rv(0)
mu_WB <- rv(0)
r_WB <- rv(0)
n_WB <- numeric()
for (i in 2013:2021){
  print(i)
  dat <- group_by(datWB_All, species) %>% filter(year == i)
  dat_ag <- summarize(dat, count=sum(count))
  y <- sort(dat_ag$count)
  n <- length(y)
  n_WB[i-2012] <- n
  input.to.stan <- species_count_in(n,y)            ## package input
# sample
  keep <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))
  fitcoef <- rvsims(as.matrix(as.data.frame(extract(keep, permute=T))))
  p0_WB <- c(p0_WB, fitcoef[3])
  N0_WB <- c(N0_WB, fitcoef[4])
  mu_WB <- c(mu_WB, fitcoef[1])
  r_WB  <- c(r_WB, fitcoef[2])
}

save(n_WB, r_WB, mu_WB, p0_WB, N0_WB, file=paste(saveDIR, "OneYear_WB.RData", sep="/"))

input.to.stan <- species_count_allyrBHM()            ## package input
# sample
keep3 <- sampling(fit3, data=input.to.stan$data,
                  init=input.to.stan$init,
                  pars=input.to.stan$para,
                  iter=niters*10,thin=nthin*10,
                  chains=input.to.stan$nchains,
                  show_messages=F, verbose=F) #,
##                    control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))
print(keep3)

save(keep3, file=paste(saveDIR, "BHM_WB.RData", sep="/"))
```

```
fitcoefBHM <- rvsims(as.matrix(as.data.frame(extract(keep3, permute=T))))
p0_BHM <- fitcoefBHM[19:27]
N0_BHM <- fitcoefBHM[28:36]
mu_BHM <- fitcoefBHM[1:9]
r_BHM <-  fitcoefBHM[10:18]
```

```
## presenting results
##tikz(file=paste(workDir, "Figs","WBbhm.tex", sep="/"), height=4,
##     width=3.25, standAlone=F)
par(mfrow=c(3,1), mar=c(0, 3, 0.25, 0.5), oma=c(3,1,1,1), mgp=c(1.75,0.25,0), las=1, tck=-0.01)
## 1. showing shrinkage in mu
Ylim <- range(cbind(summary(mu_WB)[,c(4,8)], summary(mu_BHM)[, c(4,8)]))
plot(c(0,1), c(0,1), type="n", axes=F, ylab="$\\mu$", xlab="", log="y",
     ylim=Ylim, xlim=c(0.5,9.5))
```

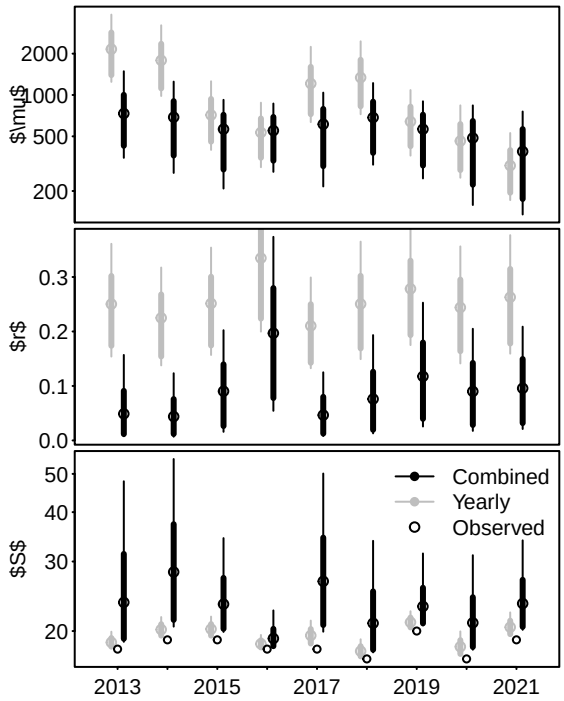
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 y value <= 0 omitted from
## logarithmic plot
```

```
segments(y0=summary(mu_WB)[,4], x0=(1:9)-0.125, y1=summary(mu_WB)[,8], x1=(1:9)-0.125, col="gray" )
segments(y0=summary(mu_WB)[,5], x0=(1:9)-0.125, y1=summary(mu_WB)[,7], x1=(1:9)-0.125, col="gray" , lwd=
points(y=summary(mu_WB)[,6], x=(1:9)-0.125, col="gray")

segments(y0=summary(mu_BHM)[,4], x0=(1:9)+0.125, y1=summary(mu_BHM)[,8], x1=(1:9)+0.125 )
segments(y0=summary(mu_BHM)[,5], x0=(1:9)+0.125, y1=summary(mu_BHM)[,7], x1=(1:9)+0.125 , lwd=3)
points(y=summary(mu_BHM)[,6], x=(1:9)+0.125)
axis(2)
#axis(1, at=1:9, labels=2013:2021, las=1)
box()

## Now r
Ylim <- range(summary(r_BHM)[, c(4,8)])
plot(c(0,1), c(0,1), xlim=c(0.5,9.5), ylim=Ylim, xlab="", ylab="$r$", axes=F)
segments(y0=summary(r_WB)[,4], x0=(1:9)-0.125, y1=summary(r_WB)[,8], x1=(1:9)-0.125, col="gray" )
segments(y0=summary(r_WB)[,5], x0=(1:9)-0.125, y1=summary(r_WB)[,7], x1=(1:9)-0.125, col="gray" , lwd=3
points(y=summary(r_WB)[,6], x=(1:9)-0.125, col="gray")

segments(y0=summary(r_BHM)[,4], x0=(1:9)+0.125, y1=summary(r_BHM)[,8], x1=(1:9)+0.125)
segments(y0=summary(r_BHM)[,5], x0=(1:9)+0.125, y1=summary(r_BHM)[,7], x1=(1:9)+0.125, lwd=3)
points(y=summary(r_BHM)[,6], x=(1:9)+0.125)

##axis(1, at=1:9, labels=2013:2021)
axis(2)
box()

## Now N0
Ylim <- range(c(unlist(summary(N0_BHM)[, c(4,8)]), n_WB))
plot(c(0,1), c(0,1), xlim=c(0.5,9.5), ylim=Ylim, xlab="", ylab="$S$",axes=F, log="y")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 y value <= 0 omitted from
## logarithmic plot
```

```
segments(y0=summary(N0_WB)[,4], x0=(1:9)-0.125, y1=summary(N0_WB)[,8], x1=(1:9)-0.125, col="gray" )
segments(y0=summary(N0_WB)[,5], x0=(1:9)-0.125, y1=summary(N0_WB)[,7], x1=(1:9)-0.125, col="gray" , lwd=
points(y=summary(N0_WB)[,6], x=(1:9)-0.125, col="gray")

segments(y0=summary(N0_BHM)[,4], x0=(1:9)+0.125, y1=summary(N0_BHM)[,8], x1=(1:9)+0.125)
segments(y0=summary(N0_BHM)[,5], x0=(1:9)+0.125, y1=summary(N0_BHM)[,7], x1=(1:9)+0.125, lwd=3)
points(y=summary(N0_BHM)[,6], x=(1:9)+0.125)

points(y=n_WB, x=1:9, pch=1, cex=0.75)
axis(1, at=1:9, labels=2013:2021, outer=T)
```

```
axis(2)
box()
legend("topright", lty=c(1,1,0), col=c("black", "gray","black"), pch=c(16,16,1), legend=c("Combined","Y
```




```
##dev.off()
```

```
## the Shakespeare word frequency data
input.to.stan <- species_count_in(nSK,ySK)            ## package input
# sample
keepSK <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))
print(keepSK)

input.to.stan <- species_count_in(nSK2,ySK2)            ## package input
# sample
keepSK2 <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))
print(keepSK2)

## the butterfly data data
input.to.stan <- species_count_in(nBF,yBF)            ## package input
# sample
```

```
keepBF <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))

print(keepBF)

input.to.stan <- species_count_in(nBF2,yBF2)            ## package input
# sample
keepBF2 <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))

print(keepBF2)

input.to.stan <- species_count_in(nBF_C,yBF_C)            ## package input
# sample
keepBF_C <- sampling(fit, data=input.to.stan$data,
                   init=input.to.stan$init,
                   pars=input.to.stan$para,
                   iter=niters,thin=nthin,
                   chains=input.to.stan$nchains,
                   show_messages=F, verbose=F,
                   control = list(adapt_delta = 0.85, stepsize = 1, max_treedepth = 15))

print(keepBF_C)

save(keepSK, keepSK2, keepBF, keepBF2, keepBF_C,
     file=paste(saveDIR, "Eval_SK_BF.RData", sep="/"))
```

extract posterior estimations

1. Posterior simulation function for estimating δ_∞

```
fitcoefSK <- rvsims(as.matrix(as.data.frame(extract(keepSK, permute=T))))
fitcoefSK2 <- rvsims(as.matrix(as.data.frame(extract(keepSK2, permute=T))))
fitcoefBF <- rvsims(as.matrix(as.data.frame(extract(keepBF, permute=T))))
fitcoefBF2 <- rvsims(as.matrix(as.data.frame(extract(keepBF2, permute=T))))
fitcoefBF_C <- rvsims(as.matrix(as.data.frame(extract(keepBF_C, permute=T))))
```

2. Examples from package `SPECIES`

```
## Shakespeare
## 1. Jackknife
SK_jknf <- jackknife(Shakespeare, k=5)
## 2. ACE coverage method
SK_ace <- ChaoLee1992(Shakespeare, t=10)
## 3. Chao 1984 lower bound
```

```
SK_lowerbound <- chao1984(Shakespeare)
## 4. Coverage-duplication
SK_covdur <- ChaoBunge(Shakespeare, t=10)
## 5. Penalized NPMLE
SK_pnpmle <- pnpmle(Shakespeare, t=15, C=1, b=200)
## 6. unconditional NPMLE
SK_unpmle <- unpmle(Shakespeare, t=10, C=1, b=200)
## 7. Pois-Comp Gamma
SK_pcg <- pcg(Shakespeare, t=20, C=1, b=200)

## Butterfly 1
## 1. Jackknife
BF1_jkknf <- jackknife(butterfly, k=5)
## 2. ACE coverage method
BF1_ace <- ChaoLee1992(butterfly, t=10)
## 3. Chao 1984 lower bound
BF1_lowerbound <- chao1984(butterfly)
## 4. Coverage-duplication
BF1_covdur <- ChaoBunge(butterfly, t=10)
## 5. Penalized NPMLE
BF1_pnpmle <- pnpmle(butterfly, t=15, C=1, b=200)
## 6. unconditional NPMLE
BF1_unpmle <- unpmle(butterfly, t=10, C=1, b=200)
## 7. Pois-Comp Gamma
BF1_pcg <- pcg(butterfly, t=20, C=1, b=200)

#################
## Removing aggregated last species

## Shakespeare
## 1. Jackknife
SK_jkknf2 <- jackknife(Shakespeare[1:100,], k=5)
## 2. ACE coverage method
SK_ace2 <- ChaoLee1992(Shakespeare[1:100,], t=10)
## 3. Chao 1984 lower bound
SK_lowerbound2 <- chao1984(Shakespeare[1:100,])
## 4. Coverage-duplication
SK_covdur2 <- ChaoBunge(Shakespeare[1:100,], t=10)
## 5. Penalized NPMLE
SK_pnpmle2 <- pnpmle(Shakespeare[1:100,], t=15, C=1, b=200)
## 6. unconditional NPMLE
SK_unpmle2 <- unpmle(Shakespeare[1:100,], t=10, C=1, b=200)
## 7. Pois-Comp Gamma
SK_pcg2 <- pcg(Shakespeare[1:100,], t=20, C=1, b=200)

## Butterfly 1
## 1. Jackknife
BF1_jkknf2 <- jackknife(butterfly[1:24,], k=5)
## 2. ACE coverage method
BF1_ace2 <- ChaoLee1992(butterfly[1:24,], t=10)
## 3. Chao 1984 lower bound
BF1_lowerbound2 <- chao1984(butterfly[1:24,])
## 4. Coverage-duplication
```

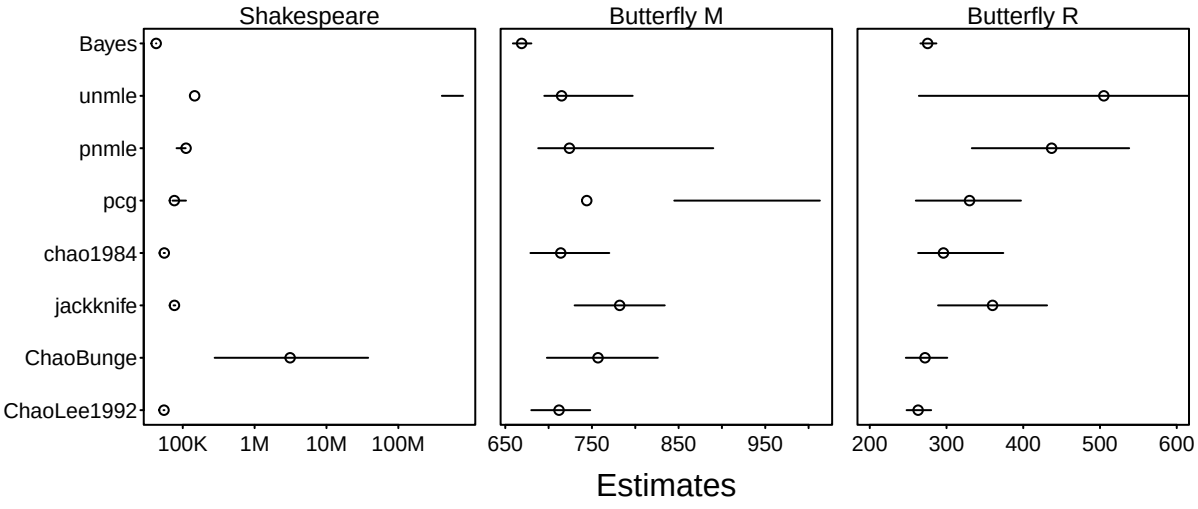
```r
BF1_covdur2 <- ChaoBunge(butterfly[1:24,], t=10)
## 5. Penalized NPMLE
BF1_pnpmle2 <- pnpmle(butterfly[1:24,], t=15, C=1, b=200)
## 6. unconditional NPMLE
BF1_unpmle2 <- unpmle(butterfly[1:24,], t=10, C=1, b=200)
## 7. Pois-Comp Gamma
BF1_pcg2 <- pcg(butterfly[1:24,], t=20, C=1, b=200)

## Butterfly 2
## 1. Jackknife
BF2_jkknf <- jackknife(WillButterfly, k=5)
## 2. ACE coverage method
BF2_ace <- ChaoLee1992(WillButterfly, t=10)
## 3. Chao 1984 lower bound
BF2_lowerbound <- chao1984(WillButterfly)
## 4. Coverage-duplication
BF2_covdur <- ChaoBunge(WillButterfly, t=10)
## 5. Penalized NPMLE
BF2_pnpmle <- pnpmle(WillButterfly, t=15, C=1, b=200)
## 6. unconditional NPMLE
BF2_unpmle <- unpmle(WillButterfly, t=10, C=1, b=200)
## 7. Pois-Comp Gamma
BF2_pcg <- pcg(WillButterfly, t=20, C=1, b=200)

save(BF1_pcg, BF1_unpmle, BF1_pnpmle, BF1_lowerbound, BF1_covdur, BF1_jkknf, BF1_ace, BF1_pcg2, BF1_unpm
```

A figure for comparing methods

```r
comp_figBF <- data.frame(mean = NULL, lower= NULL, upper=NULL)
comp_figBF <- rbind(comp_figBF, c(BF1_ace$Nhat[1], BF1_ace$CI[1,]))
comp_figBF <- rbind(comp_figBF, c(BF1_covdur$Nhat, BF1_covdur$CI))
comp_figBF <- rbind(comp_figBF, c(BF1_jkknf$Nhat, BF1_jkknf$CI))
comp_figBF <- rbind(comp_figBF, c(BF1_lowerbound$Nhat, BF1_lowerbound$CI))
comp_figBF <- rbind(comp_figBF, c(BF1_pcg$Nhat, BF1_pcg$CI))
comp_figBF <- rbind(comp_figBF, c(BF1_pnpmle$Nhat, BF1_pnpmle$CI))
comp_figBF <- rbind(comp_figBF, c(BF1_unpmle$Nhat, BF1_unpmle$CI))
comp_figBF <- rbind(comp_figBF, unlist(summary(fitcoefBF2)[4, c(2, 5, 9)]))
names(comp_figBF) <- c("Mean", "Lower", "Upper")
comp_figBF$Methods <- c("ChaoLee1992","ChaoBunge", "jackknife", "chao1984", "pcg","pnmle","unmle", "Bay

comp_figBF_c <- data.frame(mean = NULL, lower= NULL, upper=NULL)
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_ace2$Nhat[1], BF1_ace2$CI[1,]))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_covdur2$Nhat, BF1_covdur2$CI))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_jkknf2$Nhat, BF1_jkknf2$CI))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_lowerbound2$Nhat, BF1_lowerbound2$CI))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_pcg2$Nhat, BF1_pcg2$CI))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_pnpmle2$Nhat, BF1_pnpmle2$CI))
comp_figBF_c <- rbind(comp_figBF_c, c(BF1_unpmle2$Nhat, BF1_unpmle2$CI))
comp_figBF_c <- rbind(comp_figBF_c, unlist(summary(fitcoefBF_C)[4, c(2, 5, 9)]))
names(comp_figBF_c) <- c("Mean", "Lower", "Upper")
comp_figBF_c$Methods <- c("ChaoLee1992","ChaoBunge", "jackknife", "chao1984", "pcg","pnmle","unmle", "Ba

comp_figBF2 <- data.frame(mean = NULL, lower= NULL, upper=NULL)
comp_figBF2 <- rbind(comp_figBF2, c(BF2_ace$Nhat[1], BF2_ace$CI[1,]))
```

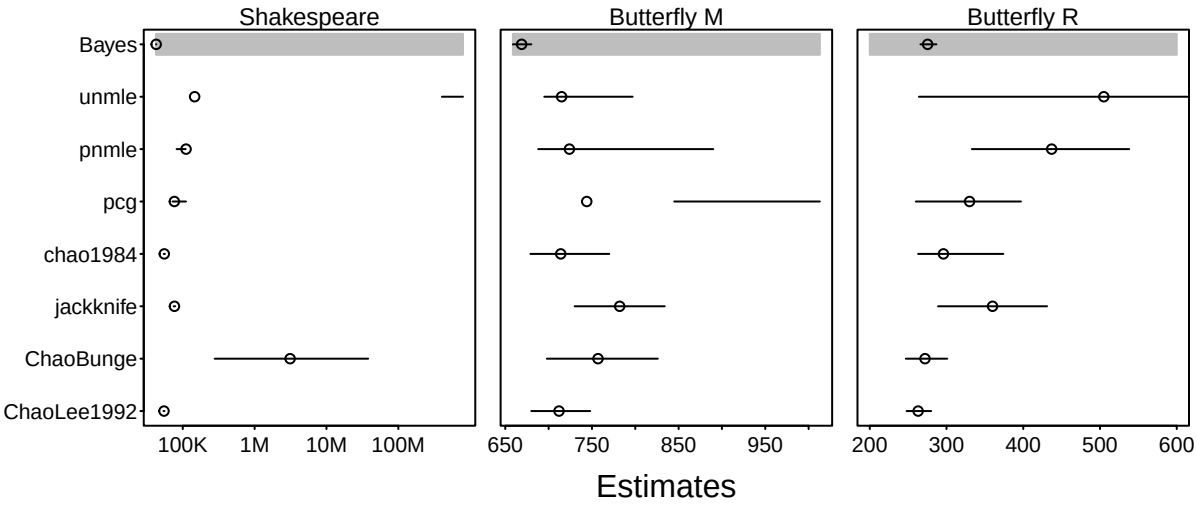
```
comp_figBF2 <- rbind(comp_figBF2, c(BF2_covdur$Nhat, BF2_covdur$CI))
comp_figBF2 <- rbind(comp_figBF2, c(BF2_jkknf$Nhat, BF2_jkknf$CI))
comp_figBF2 <- rbind(comp_figBF2, c(BF2_lowerbound$Nhat, BF2_lowerbound$CI))
comp_figBF2 <- rbind(comp_figBF2, c(BF2_pcg$Nhat, BF2_pcg$CI))
comp_figBF2 <- rbind(comp_figBF2, c(BF2_pnpmle$Nhat, BF2_pnpmle$CI))
comp_figBF2 <- rbind(comp_figBF2, c(BF2_unpmle$Nhat, BF2_unpmle$CI))
comp_figBF2 <- rbind(comp_figBF2, unlist(summary(fitcoefBF)[4, c(2, 5, 9)]))

names(comp_figBF2) <- c("Mean", "Lower", "Upper")
comp_figBF2$Methods <- c("ChaoLee1992","ChaoBunge", "jackknife", "chao1984", "pcg","pnmle","unmle","Baye

comp_figSK <- data.frame(mean = NULL, lower= NULL, upper=NULL)
comp_figSK <- rbind(comp_figSK, c(SK_ace$Nhat[1], SK_ace$CI[1,]))
comp_figSK <- rbind(comp_figSK, c(SK_covdur$Nhat, SK_covdur$CI))
comp_figSK <- rbind(comp_figSK, c(SK_jkknf$Nhat, SK_jkknf$CI))
comp_figSK <- rbind(comp_figSK, c(SK_lowerbound$Nhat, SK_lowerbound$CI))
comp_figSK <- rbind(comp_figSK, c(SK_pcg$Nhat, SK_pcg$CI))
comp_figSK <- rbind(comp_figSK, c(SK_pnpmle$Nhat, SK_pnpmle$CI))
comp_figSK <- rbind(comp_figSK, c(SK_unpmle$Nhat, SK_unpmle$CI))
comp_figSK <- rbind(comp_figSK, unlist(summary(fitcoefSK)[4, c(2, 5, 9)]))

names(comp_figSK) <- c("Mean", "Lower", "Upper")
comp_figSK$Methods <- c("ChaoLee1992","ChaoBunge", "jackknife", "chao1984", "pcg","pnmle","unmle","Bayes

comp_figSK2 <- data.frame(mean = NULL, lower= NULL, upper=NULL)
comp_figSK2 <- rbind(comp_figSK2, c(SK_ace2$Nhat[1], SK_ace2$CI[1,]))
comp_figSK2 <- rbind(comp_figSK2, c(SK_covdur2$Nhat, SK_covdur2$CI))
comp_figSK2 <- rbind(comp_figSK2, c(SK_jkknf2$Nhat, SK_jkknf2$CI))
comp_figSK2 <- rbind(comp_figSK2, c(SK_lowerbound2$Nhat, SK_lowerbound2$CI))
comp_figSK2 <- rbind(comp_figSK2, c(SK_pcg2$Nhat, SK_pcg2$CI))
comp_figSK2 <- rbind(comp_figSK2, c(SK_pnpmle2$Nhat, SK_pnpmle2$CI))
comp_figSK2 <- rbind(comp_figSK2, c(SK_unpmle2$Nhat, SK_unpmle2$CI))
comp_figSK2 <- rbind(comp_figSK2, unlist(summary(fitcoefSK2)[4, c(2, 5, 9)]))

names(comp_figSK2) <- c("Mean", "Lower", "Upper")
comp_figSK2$Methods <- c("ChaoLee1992","ChaoBunge", "jackknife", "chao1984", "pcg","pnmle","unmle","Baye


##tikz(file=paste(workDir, "Figs","Comp_fig0.tex", sep="/"), height=3, width=6.5, standAlone=F)
par(mfrow=c(1,3),oma=c(4, 6, 3, 1), mar=c(0,0.5,0.1,0.5), mgp=c(1.5,0.25,0),las=1, tck=-0.01)
plot(c(1,10), c(1,10), xlim=range(comp_figSK[,c(2,3)]), log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
##polygon(x=c(range(comp_figSK[,c(2,3)]), rev(range(comp_figSK[,c(2,3)]))), y=c(7.8,7.8,8.2,8.2), col="
for (i in 1:8){
  segments(x0=comp_figSK[i,2], x1=comp_figSK[i,3], y0=i, y1=i)
  points(x=comp_figSK[i,1], y=i)
}
axis(2, outer=F, at=1:8, labels = comp_figSK$Methods, las=1)
box()
axis(1, outer=T, at=c(100,1000,10000,100000)*1000, labels = paste(c("100","1","10","100"), c("K", rep("
mtext("Shakespeare", cex=0.75)
plot(c(1,10), c(1,10), xlim=range(comp_figBF[,c(2,3)]), ##log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
```

```
##polygon(x=c(range(comp_figBF[,c(2,3)]), rev(range(comp_figBF[,c(2,3)]))), y=c(7.8,7.8,8.2,8.2), col="
for (i in 1:8){
  segments(x0=comp_figBF[i,2], x1=comp_figBF[i,3], y0=i, y1=i)
  points(x=comp_figBF[i,1], y=i)
}
##axis(2, at=1:8, labels = comp_figBF$Methods, las=1)
axis(1, outer = T)
box()
mtext("Butterfly M", cex=0.75)

plot(c(1,10), c(1,10), xlim=c(200,600),#range(comp_figBF2[,c(2,3)]), ##log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
##polygon(x=c(200,600,600,200), y=c(7.8,7.8,8.2,8.2), col="gray",border = "gray")
for (i in 1:8){
  segments(x0=comp_figBF2[i,2], x1=comp_figBF2[i,3], y0=i, y1=i)
  points(x=comp_figBF2[i,1], y=i)
}
##axis(2, at=1:8, labels = comp_figBF2$Methods, las=1)
box()
axis(1, outer = T)
mtext("Butterfly R", cex=0.75)
mtext("Estimates", side=1, outer=T, line=2)
```




```
##dev.off()

##tikz(file=paste(workDir, "Figs","Comp_fig1.tex", sep="/"), height=3,
##     width=6.5, standAlone=F)
par(mfrow=c(1,3),oma=c(4, 6, 3, 1), mar=c(0,0.5,0.1,0.5), mgp=c(1.5,0.25,0),las=1, tck=-0.01)
plot(c(1,10), c(1,10), xlim=range(comp_figSK[,c(2,3)]), log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
polygon(x=c(range(comp_figSK[,c(2,3)]), rev(range(comp_figSK[,c(2,3)]))), y=c(7.8,7.8,8.2,8.2), col="gra
for (i in 1:8){
  segments(x0=comp_figSK[i,2], x1=comp_figSK[i,3], y0=i, y1=i)
  points(x=comp_figSK[i,1], y=i)
}
axis(2, outer=F, at=1:8, labels = comp_figSK$Methods, las=1)
box()
axis(1, outer=T, at=c(100,1000,10000,100000)*1000, labels = paste(c("100","1","10","100"), c("K", rep("
```

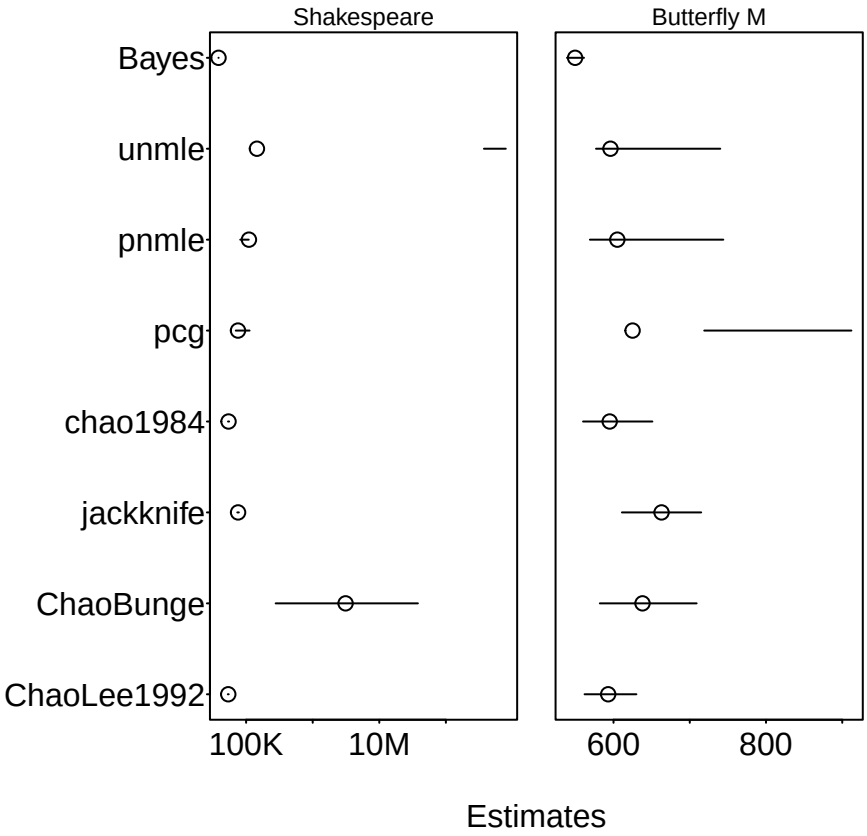
```
mtext("Shakespeare", cex=0.75)
plot(c(1,10), c(1,10), xlim=range(comp_figBF[,c(2,3)]), ##log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
polygon(x=c(range(comp_figBF[,c(2,3)]), rev(range(comp_figBF[,c(2,3)]))), y=c(7.8,7.8,8.2,8.2), col="gra
for (i in 1:8){
  segments(x0=comp_figBF[i,2], x1=comp_figBF[i,3], y0=i, y1=i)
  points(x=comp_figBF[i,1], y=i)
}
##axis(2, at=1:8, labels = comp_figBF$Methods, las=1)
axis(1, outer = T)
box()
mtext("Butterfly M", cex=0.75)

plot(c(1,10), c(1,10), xlim=c(200,600),#range(comp_figBF2[,c(2,3)]), ##log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
polygon(x=c(200,600,600,200), y=c(7.8,7.8,8.2,8.2), col="gray",border = "gray")
for (i in 1:8){
  segments(x0=comp_figBF2[i,2], x1=comp_figBF2[i,3], y0=i, y1=i)
  points(x=comp_figBF2[i,1], y=i)
}
##axis(2, at=1:8, labels = comp_figBF2$Methods, las=1)
box()
axis(1, outer = T)
mtext("Butterfly R", cex=0.75)
mtext("Estimates", side=1, outer=T, line=2)
```




```
##dev.off()
```

```
## SK 2 and BF_C
##tikz(file=paste(workDir, "Figs","Comp_fig2.tex", sep="/"), height=3, ##width=5, standAlone=F)
par(mfrow=c(1,2),oma=c(4, 6, 3, 1), mar=c(0,0.5,0.1,0.5), mgp=c(1.5,0.125,0), las=1, tck=-0.01)
plot(c(1,10), c(1,10), xlim=range(comp_figSK[,c(2,3)]), log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
for (i in 1:8){
  segments(x0=comp_figSK2[i,2], x1=comp_figSK2[i,3], y0=i, y1=i)
  points(x=comp_figSK2[i,1], y=i)
}
axis(2, outer=F, at=1:8, labels = comp_figSK2$Methods, las=1)
```

```
box()
axis(1, outer=T, at=c(100,1000,10000,100000)*1000, labels = paste(c("100","1","10","100"), c("K", rep("
mtext("Shakespeare", cex=0.75)
plot(c(1,10), c(1,10), xlim=range(comp_figBF_c[,c(2,3)]), ##log="x",
     ylim=c(1,8), axes=F, xlab="Estimates", ylab="")
for (i in 1:8){
  segments(x0=comp_figBF_c[i,2], x1=comp_figBF_c[i,3], y0=i, y1=i)
  points(x=comp_figBF_c[i,1], y=i)
}
##axis(2, at=1:8, labels = comp_figBF$Methods, las=1)
axis(1, outer = T)
box()
mtext("Butterfly M", cex=0.75)

mtext("Estimates", side=1, outer=T, line=2)
```




```
##dev.off()
```

Model Evaluation

Using posterior simulation: model predicted number of species with 1, 2, 3, . . . , n specimens

1. The model: the number of observed specimens from 1, . . . , S species

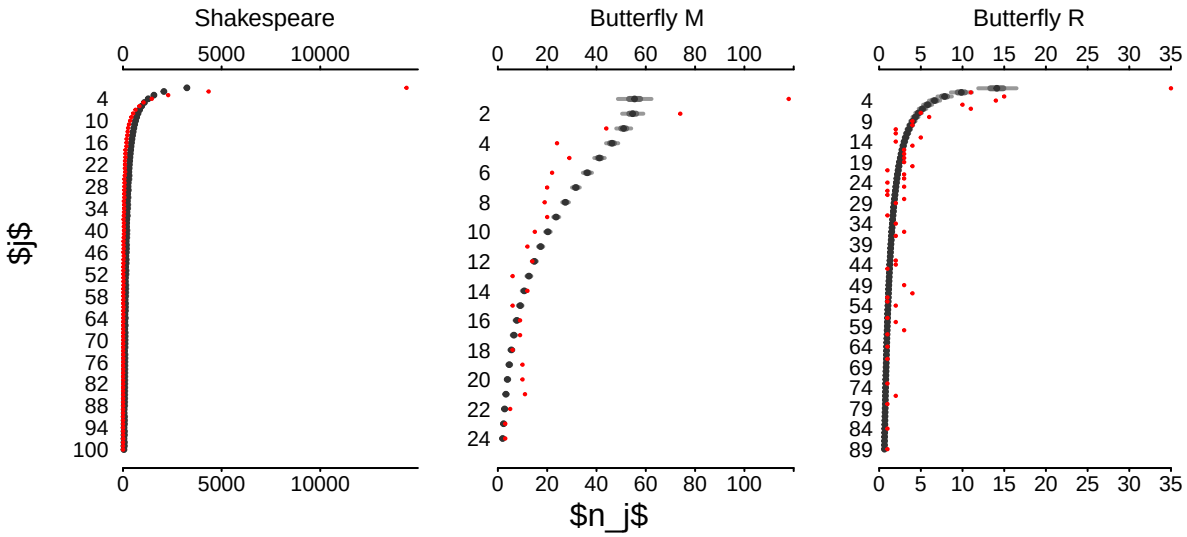
$$\Pr(n_i = k) = \frac{\Gamma(r+k)}{k!\Gamma(r)} \left(\frac{r}{\mu+r}\right)^r \left(\frac{\mu}{\mu+r}\right)^k$$

Number of *species* with k specimens is

$$S \times \Pr(n_i = k)$$

Negative binomial in R `dnbinom(x, size=r, mu)`

```
SKexpSP <- rv(0)
for (i in 1:100)
SKexpSP[i] <- rvsims(fitcoefSK$N * dnbinom(i, size=fitcoefSK$r, mu=fitcoefSK$mu))

BFexpSP.M <- rv(0)
for (i in 1:24)
BFexpSP.M[i] <- rvsims(fitcoefBF_C$N * dnbinom(i, size=fitcoefBF_C$r, mu=fitcoefBF_C$mu))

BFexpSP.R <- rv(0)
for (i in 1:dim(datBF_W)[1])
BFexpSP.R[i] <- rvsims(fitcoefBF$N * dnbinom(i, size=fitcoefBF$r, mu=fitcoefBF$mu))

##tikz(paste(workDir, "Figs","Comp_fig3.tex", sep="/"), height=3, ##width=6.5, standAlone=F)
par(mfrow=c(1,3), oma=c(3,3,2,1), mar=c(0,0,0,0), mgp=c(1.5,0.125,0), las=1, tck=-0.01)
Xlim=range(c(Shakespeare$n_j[1:100]))
mlplot(SKexpSP, cex=0.5, xlim=Xlim)
points(Shakespeare[1:100,]$n_j, Shakespeare[1:100,]$j, col="red", cex=0.25)
mtext("Shakespeare", side=3, cex=0.75, line=-0.5)
Xlim=range(datBF_C$n_j)
mlplot(BFexpSP.M, cex=0.5, xlim=Xlim)
points(datBF_C$n_j, datBF_C$j, col="red", cex=0.25)
mtext("Butterfly M", cex=0.75, line=-0.5)
Xlim=range(datBF_W$n_j)
mlplot(BFexpSP.R, cex=0.5, xlim=Xlim)
points(datBF_W$n_j, datBF_W$j, col="red", cex=0.25)
mtext("Butterfly R", cex=0.75, line=-0.5)

mtext("$n_j$", side=1, outer=T, line=1.5)
mtext("$j$", side=2, outer=T, las=0)
```




```
##dev.off()
## probability of observing j specimens
```