Model Development

Alyssa Willson

2023-04-26

Load data

We will use a very small portion of our data for the last time period in our fossil pollen record. We'll load the data, subset it, and then remove the rest.

```
### Load data ###

load('FossilPollen/Data/full_melt_UMW.RData')
load('FossilPollen/Data/sd_UMW.RData')

# For now let's join these to make things easier
pollen_data <- full_melt |>
  full_join(full_sd, by = c('time', 'long', 'lat', 'loc_time'))

load('Climate/processed_climate.RData')

### Start processing climate drivers ###

clim <- clim1 |>
  select(-Time) |>
  mutate(Loc = paste0(Latitude,'_',Longitude),
         Loc_Year = paste0(Loc,'_',Year))

loc_years <- unique(clim$Loc_Year)

climate <- matrix(, nrow = length(loc_years), ncol = ncol(clim))
colnames(climate) <- colnames(clim)

for(i in 1:length(loc_years)){
  temp <- clim |>
    filter(Loc_Year == loc_years[i])
  if(length(unique(temp$Longitude)) > 1){print('ERROR')}
  if(length(unique(temp$Latitude)) > 1){print('ERROR')}
  temperature <- mean(temp$Temperature)
  precipitation <- sum(temp$Precipitation)
  if(length(unique(temp$Year)) > 1){print('ERROR')}
  if(length(unique(temp$Month)) != 12){print('ERROR')}
  if(length(unique(temp$Loc)) > 1){print('ERROR')}
  if(length(unique(temp$Loc_Year)) > 1){print('ERROR')}

  climate[i,] <- c(temp$Longitude[1], temp$Latitude[1], temperature, precipitation,
                   temp$Month[1], temp$Year[1], temp$Loc[1], temp$Loc_Year[1])
}
```

```
climate <- as.data.frame(climate)
climate$Longitude <- as.numeric(climate$Longitude)
climate$Latitude <- as.numeric(climate$Latitude)
climate$Temperature <- as.numeric(climate$Temperature)
climate$Precipitation <- as.numeric(climate$Precipitation)
climate$Year <- as.numeric(climate$Year)
climate$Month <- as.numeric(climate$Month)

### Match climate and fossil pollen data ####
spat_climate <- climate |>
  filter(Year == 1900) |>
  filter(Month == 1)

coordinates(spat_climate) <- ~Longitude+Latitude

spat_pollen <- pollen_data
coordinates(spat_pollen) <- ~long+lat

d <- gDistance(spgeom1 = spat_pollen, spgeom2 = spat_climate, byid = T)
mins <- apply(d, 2, which.min)

mins <- as.data.frame(mins)
mins <- cbind(pollen_data, mins)

mins$Longitude <- NA
mins$Latitude <- NA
mins$Temperature <- NA
mins$Precipitation <- NA

for(i in 1:nrow(mins)){
  ind <- mins$mins[i]
  mins$Longitude[i] <- climate$Longitude[ind]
  mins$Latitude[i] <- climate$Latitude[ind]
}

for(i in 1:nrow(mins)){
  ind <- mins$mins[i]
  longitude <- mins$Longitude[i]
  latitude <- mins$Latitude[i]
  year <- mins$time[i]

  ii <- which(climate$Longitude == longitude &
                climate$Latitude == latitude &
                climate$Year == year)

  mins$Temperature[i] <- climate$Temperature[ii]
  mins$Precipitation[i] <- climate$Precipitation[ii]
}
```

```
cols <- colnames(mins)
ydata_columns <- which(grepl('.x', cols, fixed = T))
xdata_columns <- which(cols == 'Temperature' | cols == 'Precipitation')
```

```
ydata <- mins |>
  select(all_of(ydata_columns))
xdata <- mins |>
  select(all_of(xdata_columns))

clim_sub <- xdata |>
  slice_head(prop = 0.1)
data_sub <- ydata |>
  slice_head(prop = 0.1)

rm(clim, clim1, climate, d, full_melt, full_sd, mins, pollen_data, spat_climate, spat_pollen, temp, xda
```

Model 1: Observations drawn from a Gaussian distribution

Now, let's write a simple model to maek sure we know what we're doing in JAGS. We'll assume that our simulated data is normally distributed with mean μ and precision τ:

$$y_{i,j} \sim \mathcal{N}(\mu, \tau)$$

$$\mu \sim \mathcal{N}(0, 0.001)$$

$$\tau \sim \text{Gamma}(0.001, 0.001)$$

```
gauss_model <- "
model{
for(i in 1:nobs){
for(j in 1:ntaxa){
y[i,j] ~ dnorm(mu, tau)
}
}
mu ~ dnorm(0, 0.001)
tau ~ dgamma(0.001, 0.001)
}
"
```

Now, we'll compile and run our JAGS model.

```
data <- list(nobs = nrow(data_sub),
             ntaxa = ncol(data_sub),
             y = data_sub)
jags.gauss_model <- jags.model(file = textConnection(gauss_model),
                               data = data,
                               n.chains = 3)
out.gauss_model <- coda.samples(model = jags.gauss_model,
                                variable.names = c('mu', 'tau'),
                                n.iter = 1000)
```

Now, we can do a quick visualization of this output. We expect to have converged, but we also expect that this won't be particularly informative since a Normal distribution is inappropriate for this data and each observation and taxon is fit separately.

```
plot(out.gauss_model)
gelman.plot(out.gauss_model)
```

Model 2: Observations drawn from a Dirichlet distribution

We can repeat this with a vrsion of the model in which the fractional composition observations are drawn from a Dirichlet distribution. THis is appropriate because fractional composition is limited to (0, 1) and is constrained to sum to 1. Using the same notation as before, we have

$$\boldsymbol{y}_i \sim \text{Dirichlet}(\boldsymbol{\alpha})$$

$$\alpha_j \sim \text{Gamma}(0.001, 0.001)$$

In this case, the taxa at each location are modeled jointly, as indicated by the vector $\boldsymbol{y}_i$. Each taxon has a different paramter describing the Dirichelt distribution, $\boldsymbol{\alpha}$, each with an uninformative Gamma prior, with support $(0, \infty)$.

```
dirch_model <- "
model{
for(i in 1:nobs){
y[i,1:ntaxa] ~ ddirch(alpha[1:ntaxa])
}
for(j in 1:ntaxa){
alpha[j] ~ dgamma(0.001, 0.001)
}
}
"
```

```
data <- list(nobs = nrow(data_sub),
             ntaxa = ncol(data_sub),
             y = data_sub)
jags.dirch_model <- jags.model(file = textConnection(dirch_model),
                               data = data,
                               n.chains = 3)
out.dirch_model <- coda.samples(model = jags.dirch_model,
                                variable.names = 'alpha',
                                n.iter = 1000)
```

```
plot(out.dirch_model)
gelman.plot(out.dirch_model)
```

Convergence looks great. We will use this model as the basis for a model with climate drivers.

Model 3: Including a hyperprior

Before modeling fractional composition with climate drivers, let's add a hyper prior to our model. When we add climate drivers to our model, we'll be incorporating a series of linear models with climate drivers and the α parameters of the Dirichlet distribution as the response. Therefore, we want to be able to manipulate the α parameters instead of specifying a vague prior for them.

We'll start with the following.

$$\boldsymbol{y}_i \sim \text{Dirichlet}(\boldsymbol{\alpha})$$

$$\alpha_j \sim \text{Gamma}(\alpha_\alpha, \beta_\alpha)$$

$$\alpha_\alpha \sim \text{Gamma}(0.001, 0.001)$$

$$\beta_\alpha \sim \text{Gamma}(0.001, 0.001)$$

```
dirch_hyper_model <- "
model{
for(i in 1:nobs){
y[i,1:ntaxa] ~ ddirch(alpha[1:ntaxa])
}
for(j in 1:ntaxa){
alpha[j] ~ dgamma(alpha_alpha, beta_alpha)
}
alpha_alpha ~ dgamma(0.001, 0.001)
beta_alpha ~ dgamma(0.001, 0.001)
}
"
```

```
data <- list(nobs = nrow(data_sub),
             ntaxa = ncol(data_sub),
             y = data_sub)
jags.dirch_hyper_model <- jags.model(file = textConnection(dirch_hyper_model),
                                     data = data,
                                     n.chains = 3)
out.dirch_hyper_model <- coda.samples(model = jags.dirch_hyper_model,
                                      variable.names = c('alpha_alpha', 'beta_alpha'),
                                      n.iter = 1000)
```

```
plot(out.dirch_hyper_model)
gelman.plot(out.dirch_hyper_model)
```

Bad convergence. Let's think about what we just did. We said that the proportion of each taxon at each location is drawn from the concentration parameters of each taxon at all locations. So, we are assuming that some taxa are more abundant than others across all locations, which may not be appropriate.

Model 4: Site-specific parameters

While more complicated, it is probably worth making the α parameters site-specific:

$$\boldsymbol{y}_i \sim \text{Dirichlet}(\boldsymbol{\alpha}_i)$$

$$\alpha_{i,j} \sim \text{Gamma}(\alpha_{\alpha,j}, \beta_{\beta,j})$$

$$\alpha_{\alpha,j} \sim \text{Gamma}(0.001, 0.001)$$

$$\beta_{\alpha,j} \sim \text{Gamma}(0.001, 0.001)$$

Here, I have also chosen to allow the concentration of each taxon to be drawn independently from the uninformative prior. This was done to allow more flexibility in the model, so that the concentration hierarchically differs by site, then by taxon.

```
dirch_hyper_unpool_model <- "
model{
for(i in 1:nobs){
y[i,1:ntaxa] ~ ddirch(alpha[i,1:ntaxa])
for(j in 1:ntaxa){
alpha[i,j] ~ dgamma(alpha_alpha[j], beta_alpha[j])
}
}
for(j in 1:ntaxa){
alpha_alpha[j] ~ dgamma(0.001, 0.001)
beta_alpha[j] ~ dgamma(0.001, 0.001)
}
}
"
```

```
data <- list(nobs = nrow(data_sub),
             ntaxa = ncol(data_sub),
             y = data_sub)
jags.dirch_hyper_unpool_model <- jags.model(file = textConnection(dirch_hyper_unpool_model),
                                            data = data,
                                            n.chains = 3)
out.dirch_hyper_unpool_model <- coda.samples(model = jags.dirch_hyper_unpool_model,
                                             variable.names = c('alpha_alpha', 'beta_alpha'),
                                             n.iter = 1000)
```

```
#plot(out.dirch_hyper_unpool_model)
#gelman.plot(out.dirch_hyper_unpool_model)
```

Model 5: Incorporating climate drivers

Now, let's model the α parameters as a function of climate drivers:

$$\boldsymbol{y}_i \sim \text{Dirichlet}(\boldsymbol{\alpha}_i)$$

$$\alpha_{i,j} \sim \text{Gamma}(\alpha_{\alpha,i,j}, \beta_{\beta,i,j})$$

$$\alpha_{\alpha,i,j} = \frac{\mu_{i,j}^2}{\sigma_{i,j}^2}$$

$$\beta_{\alpha,i,j} = \frac{\mu_{i,j}}{\sigma_{i,j}^2}$$

$$\mu_{i,j} = \beta_0 + beta_{1,j} x_{i,1} + beta_{2,j} x_{i,2}$$

$$\sigma_{i,j}^2 = \frac{1}{\tau_{i,j}^2}$$

$$\tau_{i,j} \sim \text{Gamma}(0.001, 0.001)$$

$$\boldsymbol{\beta} \sim \text{MVN}(\mu_\beta, \tau_\beta)$$

```
dirch_driver_model <- "
model{
for(i in 1:nobs){
y[i,1:ntaxa] ~ ddirch(alpha[i,1:ntaxa])
for(j in 1:ntaxa){
alpha[i,j] ~ dgamma(alpha_alpha[i,j], beta_alpha[i,j])

alpha_alpha[i,j] <- mu[i,j]^2 / sigma.sq[j]
beta_alpha[i,j] <- mu[i,j] / sigma.sq[j]

logit(mu[i,j]) <- beta[1] + beta[match[1,j]] * x[i,1] + beta[match[2,j]] * x[i,2]
}
}
for(j in 1:ntaxa){
sigma.sq[j] <- 1 / tau[j]
tau[j] ~ dgamma(0.001, 0.001)
}
beta ~ dmnorm(mu_beta, tau_beta)
}
"
```

```
maxparam <- (2 * ncol(data_sub)) + 1
match <- seq(from = 2, to = maxparam, by = 1)
match <- matrix(match, nrow = 2, ncol = maxparam)

data <- list(nobs = nrow(data_sub),
             ntaxa = ncol(data_sub),
             y = data_sub,
             x = clim_sub,
             match = match,
             mu_beta = rep(x = 0, times = maxparam),
             tau_beta = diag(x = 0.001,
                             nrow = maxparam,
                             ncol = maxparam))
jags.dirch_driver_model <- jags.model(file = textConnection(dirch_driver_model),
                                      data = data,
                                      n.chains = 3)
out.dirch_driver_model <- coda.samples(model = jags.dirch_driver_model,
                                       variable.names = c('beta', 'tau'),
                                       n.iter = 1000)
```

```
plot(out.dirch_driver_model)
gelman.plot(out.dirch_driver_model)
```

While not converged, this is really promising given how few observations we are using right now. This is a good start, but we aren't just interested in the relationship between climate and vegetation. We're also interested in how the vegetation moderates that relationship, and how space and time factor in.

Model 6: Incorporating species covariance

The next step is adding species covariance into the model. This is accomplished by adding a random species effect $\boldsymbol{\epsilon}$ into the equation governing the relationship between fractional composition and the climate drivers:

$$\boldsymbol{y}_i \sim \text{Dirichlet}(\boldsymbol{\alpha}_i)$$