Soil example

Marie-Anne deGraff

2/9/2020

Estimating carbon "production"

For an explanation of the statistical model, see the Estimation Model section below.

Simulating in R

Chemical model

Now we need to simulate some random chemical concentrations with a little bit of variability. The parameters $\beta^{(c)}$ will not have any biological meaning, but the resulting proportions will.

Start by defining the dimensions of data and the multinomial logit function that generates the chemical proportions,

```
nc <- 4 #number of chemicals observed
ni <- 100 #number of experiments
proc_sig2_c <- 0.5 #variability in the samples
multi_logit <- function(coef){
  #coef: list of multinomial coefficients
  #predictor variables
  ml_i <- c(0,coef)
  z <- exp(ml_i)/(1+sum(exp(ml_i[2:length(ml_i)])))
  return(z)
}
```

Now, generate the parameters, $\beta^{(c)}$, that produce the chemical proportions, but assume a little random variability in those parameters. I will use a mutlivariate normal distribution for reasons that I can explain later. For identifiability, you $n_c - 1$ parameters in the model, where n_c is the number of chemical you are observing. In this sense, we set $\beta^{(c=1)} = 0$, where $c = 1$ is the first chemical in your vector of chemicals that you are measuring.

```
library(mvtnorm)
beta_c <- c(-0.5,1,1.5) #true parameters of the multinomial logit
Sigma <- matrix(0,nc-1,nc-1) #Covariance of the multinomial logit parameters
diag(Sigma) <- proc_sig2_c #Variance
c_i <- rmvnorm(ni,beta_c,Sigma)  #parameter generating process
```

Now, generate the observed chemical proportions. These the random chemical *observations*!

```
p_c <- matrix(NA,ni,nc)
for(i in 1:ni){
  p_c[i,] <- multi_logit(c_i[i,])
}
```

Bacteria model

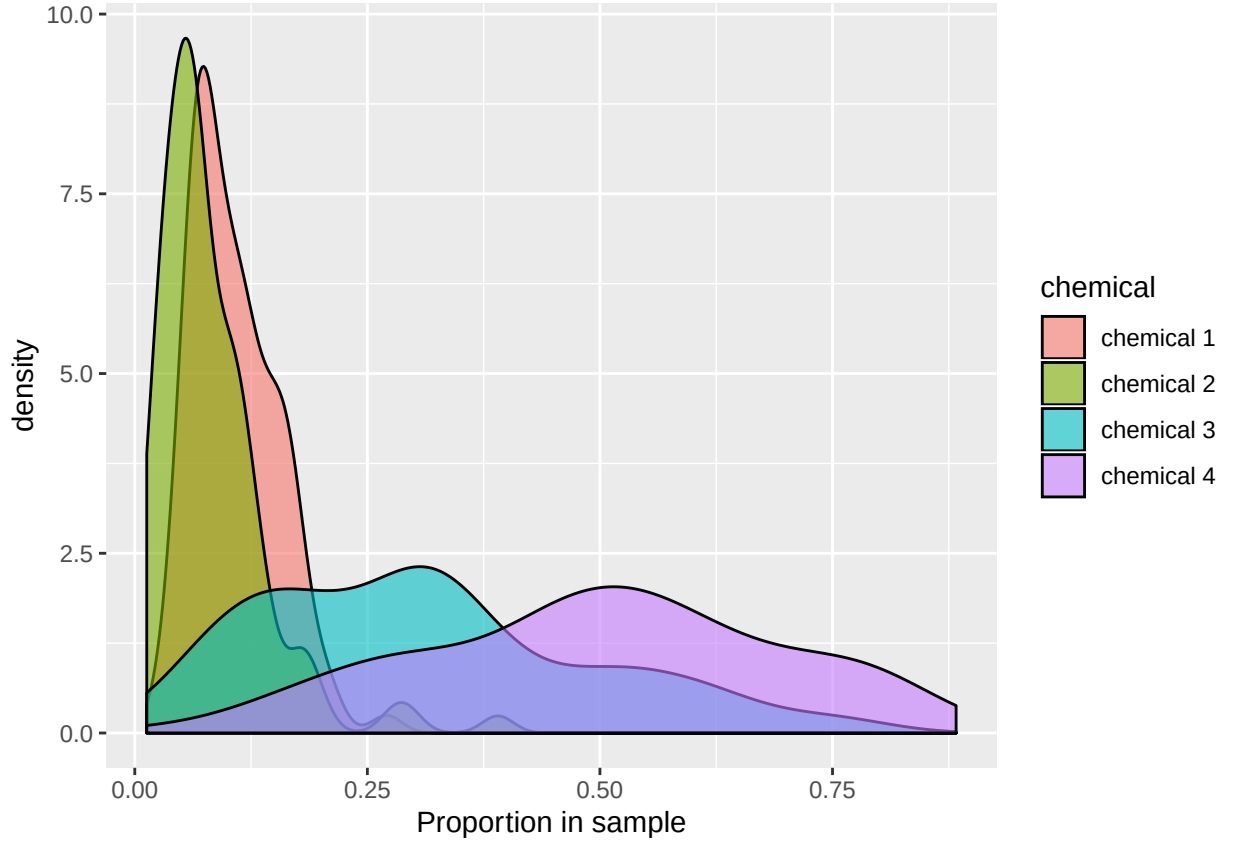
This is what the data look like for 100 samples for four chemicals based on a little random variability in the samples,

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.1
```

```
df_c <- data.frame(chemical=rep(paste("chemical",1:4), each=ni),
                   p_c=c(p_c))

pl <- ggplot(df_c, aes(x=p_c, fill=chemical)) +
  geom_density(alpha=0.6)+
  xlab("Proportion in sample")
print(pl)
```



Now, we'll use the same data generating process for producing samples of bacterial proportions based on the chemical concentrations in each sample. To do the I've updated the multinomial logit to account for the fact that each bacteria is a function of the vector of chemical concentrations in each sample.

Let's start by assume there are five bacteria,

```
#Estimate the bacteria mixture from chemical mixture
nb <- 5 #number of bacteria

#Matrix of coefficients relating chemical concentrations to bacteria
b_c <- rbind(c(0.25,0.5,1.,1.5)*0.1, #mean bacteria effects
             matrix(runif((nb-1)*nc),nb-1,nc)) #chemical interaction effects)


p_b <- matrix(NA,ni,nb) #matrix of bacteria ratios
```

This is the slightly modified multinomial logit function,

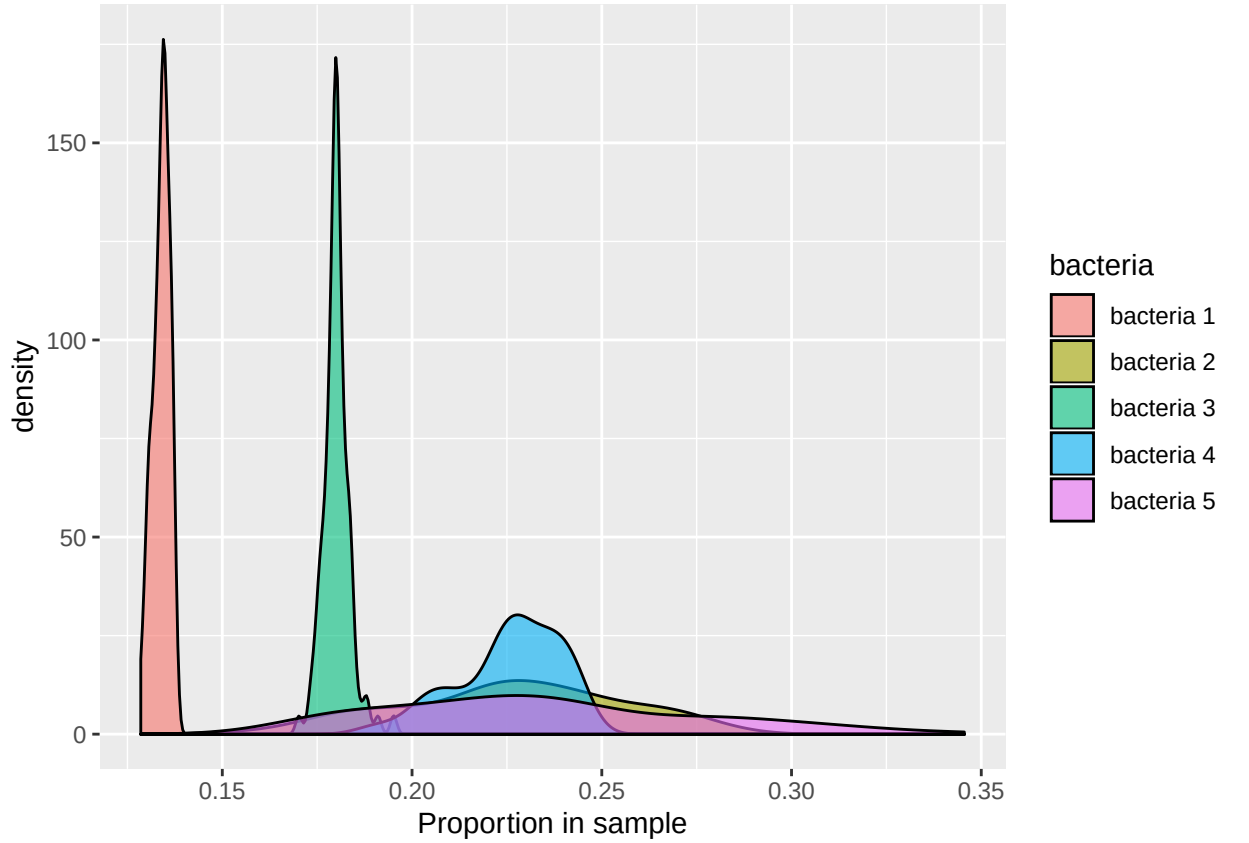
```
multi_logit2 <- function(coef,x){
  #coef: list of multinomial coefficients
  #predictor variables
  z_tmp <- c(1,rep(0,ncol(coef)))
  for(ii in 1:ncol(coef)){ #move across bacteria
    z_tmp[ii+1] <- exp(sum(coef[,ii]*x))
  }
  z <- z_tmp/sum(z_tmp)
  return(z)
}
```

Now the fun part, let's generate some bacteria concentrations,

```
p_x <- cbind(rep(1,ni),p_c) #get the chemical concentrations
#Use the multinomial to take the chemical concentrations and predict biological concentrations
for(i in 1:ni){
  p_b[i,] <- multi_logit2(b_c,p_x[i,])
}

library(ggplot2)
df_b <- data.frame(bacteria=rep(paste("bacteria",1:5), each=ni),
                   p_b=c(p_b))

pl <- ggplot(df_b, aes(x=p_b, fill=bacteria)) +
  geom_density(alpha=0.6)+
  xlab("Proportion in sample")
print(pl)
```



The output is similar to the chemical proportions, but now we are simulating a bacterial concentration from chemical concentration,

```
p_x <- cbind(rep(1,ni),p_c) #get the chemical concentrations
#Use the multinomial to take the chemical concentrations and predict biological concentrations
for(i in 1:ni){
  p_b[i,] <- multi_logit2(b_c,p_x[i,])
}
```

Chemical flux

Now model the carbon flux. I don't what these equations are, but for right now they are just linear. Start by creating the parameters for a linear relationship.

```
#Now estimate carbon flux from bacteria mixture as a function of temperature
mu_lam = 0
b_int <- runif(nb,min=1,max=10)
b_slope <- runif(nb,min=-1,max=1)
temps <- c(0,5,10,15)
t <- rep(temps,each=ni/length(temps))
nu <- rep(0,ni)
for(i in 1:ni){
  nu[i] <- sum(mu_lam + (b_int + b_slope * t[i])*p_b[i,])
}
```

Next we need to create the temperature related experiment. So just assume that the experiment is divided into four equal treatment group. Temperature does not need to be categorical, it could just as easily be

continuous.

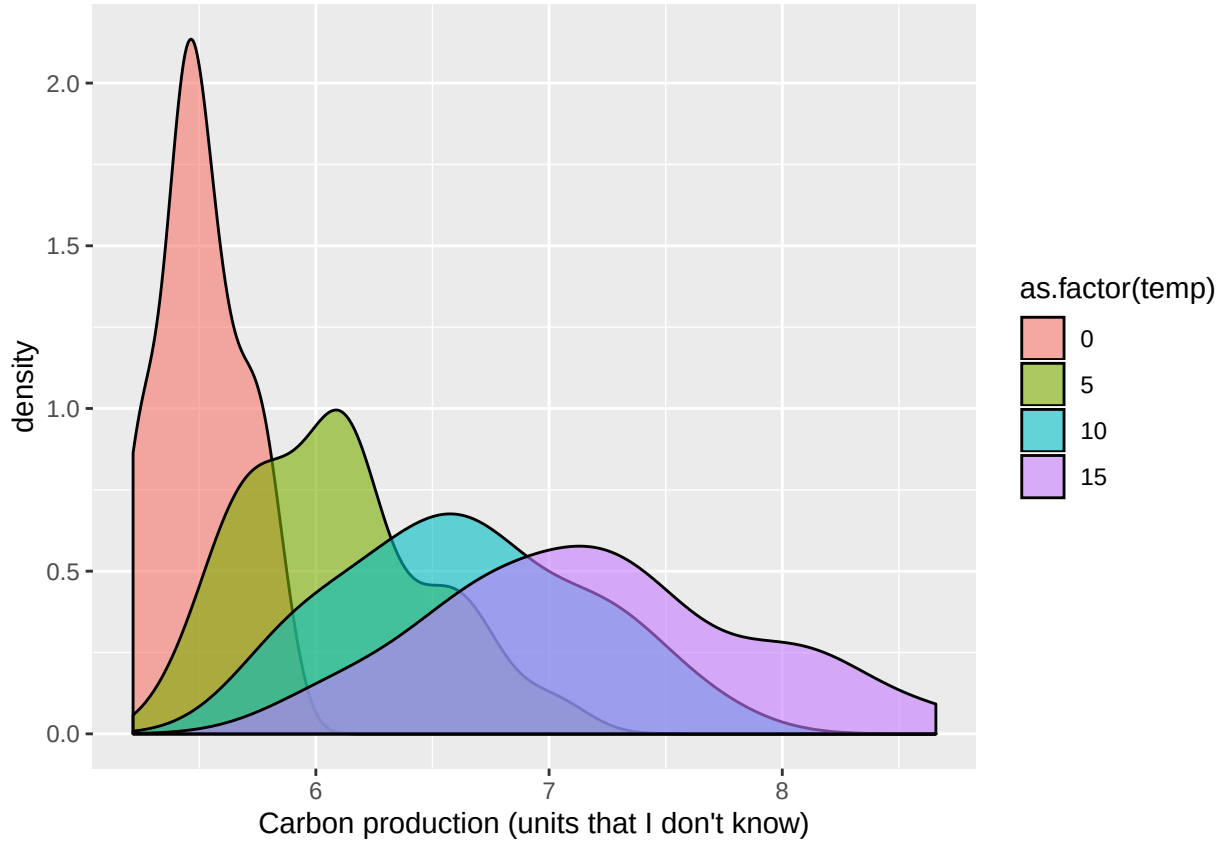
```
#Now create the temperature experiment
temps <- c(0,5,10,15)
t <- rep(temps,each=ni/length(temps))
```

Now create your carbon production as a function of the bacteria proportions, which are themselves a function of the chemical proportions.

```
#Now estimate carbon flux from bacteria mixture as a function of temperature
nu <- rep(0,ni)
for(i in 1:ni){
  nu[i] <- sum(mu_lam + (b_int + b_slope * t[i])*p_b[i,])
}

df_C <- data.frame(temp=t, C=nu)

pl <- ggplot(df_C, aes(x=nu, fill=as.factor(temp))) +
  geom_density(alpha=0.6)+
  xlab("Carbon production (units that I don't know)")
print(pl)
```



Review of the model

Hi, Marie-Anne. This is a simulation model with statistical properties so it is easy enough to now turn around and estimate the parameters of the model that produced these simluated results. We should chat though, about whether there are non-linear dynamics.

The model of the chemical mixture is,

$$\hat{p}^{(c)} = \frac{exp\Big(\beta^{(c)}\Big)}{1+\sum_{i=2}^{n_c}\Bigg(exp\Big(\beta^{(c)}\Big)\Bigg)}$$

where, n_c is the number of chemicals, $\beta^{(c)}$ is the coefficient describing the proportion of chemical c is the soil. To maintain identifiability of the problem $\beta^{(c=1)} = 0$, such that there are only $n_c - 1$ estimable parameters for the chemical mixture.

The likelihood of the vector of chemical concentrations for the i^{th} sample, $\mathbf{p_i^c}$, given the vector of predicted chemical concentrations, $\hat{\mathbf{p}}^c$, is multinomially distributed,

$L(\mathbf{p_i^c}|\boldsymbol{\beta}) \sim \mathbf{Multinomial}(\hat{\mathbf{p}}^\mathbf{c}, \mathbf{p^c})$

The biological model

The estimate of the bacterial mixture,$\hat{p}^{(b)}$, is a function of the chemical mixture, $\hat{p}^{(c)}$,

$$\hat{p}^{(b)} = \frac{exp\Big((\mathbf{A}^{(b)})^T\hat{\mathbf{p}}^c\Big)}{1+\sum_{i=2}^{n_b} exp\Big((\mathbf{A}^{(b)})^T\hat{\mathbf{p}}^c\Big)}$$

where, n_b is the number of measured bacteria in the sample, $\mathbf{A}^{(b)}$ is a vector of coefficients relating bacteria b to the vector of predicted chemical concentrations,$\hat{\mathbf{p}}^c$. Just as there are only $n_c - 1$ coefficients estimated for the chemical composition, there are only $n_b - 1$ vectors of bacteria coefficient for the bacterial mixture; thus, the dimensions of the

The model of bacterial mixture as a function of the chemical mixture is as follows, $\mathbf{A}$ are $(n_b - 1, n_c)$.

The likelihood of the vector of observed biological proportions for the i^{th} sample given the matrix of biological coefficients, $\mathbf{A}$ is also assumed to be multinomially distributed,

$L(\mathbf{p_j^b}|\mathbf{A}) \sim \mathbf{Multinomial}(\hat{\mathbf{p}}^\mathbf{b}, \mathbf{p_j^b})$

Model of carbon production.

I puling this out of thin air at this point. I don't know if these are linear or non-linear relationships, but let's start with linear for now to keep it simple.

Let's assume that the model of carbon production is an additive effect as a function of the biological proportions,

$\hat{c}_i = \sum_b \alpha^{(b)} \times \gamma^{(b)}\hat{p}_i^{(b)}$

Then the likelihood of the observed carbon production, c_i, given the parameter vector, $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, is log-normally distrbuted (that's a guess, we can talk late),

$\mathcal{L}(c_i|\boldsymbol{\alpha}, \boldsymbol{\gamma}, \sigma) \sim Normal(log(c_i), log(\hat{c}_i), \sigma)$

Optimization in TMB

Hi Marie-Anne, check out the C++ code. You'll need to download the TMB library for R. I've attached the C++ as well as the DLL for running the optimization. You don't need to recompile the code you just need to run the wrapper ("optimizeModel.r"). I don't have to estimate the random effects model right now, but we can work on this later if you're interested.