

# MCZ Trait Measurement Concept

*Clement Garcia*

*9 May 2019*

## Rationale

### Monitoring

Seabed supplies many functions and services that anthropogenic activities are threatening. UK marine monitoring strategy endeavours to keep ‘clean, healthy, safe, productive and biologically diverse oceans and seas’.

Current monitoring strategies do consider the conceptual importance of structure and function of the seabed, but the measured parameters tend to focus ‘only’ on defining the sediment type or describing the structure of macrobenthic communities (Shannon, Evenness, ID and taxa-specific abundance). To understand better the functioning of the benthos, trait-based approaches are routinely used as a proxy between community structure and the processes they mediate.

However trait definition are mostly derived from literature generalisation with their corresponding shortcomings (space and time mismatch, taxo change, and taxa-specific value with no consideration for intra-specific variation)

### Traits measurement

It is well acknowledged that individual measurement of traits would greatly improve the accuracy of trait-based approach for the assessment of seabed functioning (being time and space-specific and allowing for the detection of minute change not achievable at higher resolution). Additionally, recent studies have found that the importance of intra-specific variation in trait-expression may be higher than cross-specific variation, ignoring it might therefore lead to improper conclusion regarding the trait-based derivation of ecosystem processes mediated by the fauna.

We argue that, with a new approach combining current method of sample processing together with little extra focalised efforts on measurement of functional trait-related parameters collection (e.g. zooscan) and modelling (trait-based methods) approach may considerably increase the quantity and quality of information available for data analysis with little extra cost, thereby improve the value for money of each benthic samples routinely taken during monitoring programme. Though we do acknowledge that measurement of physiology or behaviour related traits information demand medium to long term mesocosms experiment to be derived properly, most of the morphology-based parameters and some aspect of feeding and life-history may be recorded by direct measurement on the benthic individual collected.

### Objective

The success of such an approach require the knowledge of the amount of effort needed in order to acquire the most amount of information out of benthic community collection and subsequent sample processing, illustrated by individual biomass trait measurement.

The following analyses uses real data in order to identifying possible way of measuring the trade-off between the efforts put into sample processing and the amount of information obtain from benthic samples. In other words, we attempt to calculate how much information is obtain per unit of effort when processing biological sample. Subsequently, a few examples are given to show what can we do with the extra information given by such approach. The present report aims at:

- 1 - Present a method to adapt the number of replicates to the community complexity;
- 2 - Show how decision can be made as to which taxa should be further analysed;

- 3 - Define when the number of individuals should be subset and how many records are necessary;
- 4 - Present an application of possible use of individual biomass data;
- 5 - Suggest future direction for obtaining more information from sample.

Should the proposal be successful, we will then aim at applying this method from start to finish to a coherent set of data in order to establish a workable protocol directly applicable to current monitoring programme.

## Community data

### Real sampling data

This part presents a method to inform about the number of replicates ‘needed’ to characterise properly the community in place.

For this we use real community data as they would be gathered from regular benthic sample collection: taxa identity, abundance and biomass.

First we format the data.

```
## Abundance
ABNdf<-ABN %>%
  gather(Site, A, WSL.H_A_1_O:HP35_B_3_N) %>%
  separate(Site, c("Station", "Rep", "RepNb", "Design"), "_") %>%
  unite("Replicate", c("Rep", "RepNb")) %>%
  filter(Design == "N")

## Biomass
BIOdf<-BIO %>%
  gather(Site, B, WSL.H_A_1_O:HP35_B_3_N) %>%
  separate(Site, c("Station", "Rep", "RepNb", "Design"), "_") %>%
  unite("Replicate", c("Rep", "RepNb")) %>%
  filter(Design == "N")

## Combining
COMdf<-cbind(ABNdf, BIOdf[, 'B'])
colnames(COMdf)[6]<- 'B'

COMdf[COMdf$Replicate %in% "A_1", 'Replicate']<- 'R1'
COMdf[COMdf$Replicate %in% "A_2", 'Replicate']<- 'R2'
COMdf[COMdf$Replicate %in% "A_3", 'Replicate']<- 'R3'
COMdf[COMdf$Replicate %in% "B_1", 'Replicate']<- 'R4'
COMdf[COMdf$Replicate %in% "B_2", 'Replicate']<- 'R5'
COMdf[COMdf$Replicate %in% "B_3", 'Replicate']<- 'R6'
```

We’re now looking at how much information is brought with each replicate using the real data

```
## Average decrease of uncertainty that the replicate 1 provides
## Selection of replicate 1 - Abn
Arepl1<-COMdf %>%
  select(Taxon, Station, Replicate, A) %>%
  filter(Replicate %in% 'R1') %>%
  spread(Taxon, A)

## Selection of replicate 1 - Bio
Brep1<-COMdf %>%
  select(Taxon, Station, Replicate, B) %>%
```

```

filter(Replicate %in% 'R1') %>%
spread(Taxon, B)

## formatting
rownames(Arep1)<-Arep1$Station
rownames(Brep1)<-Brep1$Station
Arep1[, c('Station', 'Replicate')]<-NULL
Brep1[, c('Station', 'Replicate')]<-NULL

## Evolution of uncertainty after each replicate
ABNBIT<-expand.grid(Station = unique(COMdf$Station), Parameter = 'ABN',
                      Replicate = paste0('R', 0:6), NbUnit= NA, NbInd=NA,
                      H = NA, Bits = NA, BitsDatum = NA, BitsDatum_Ind=NA)
BIOBIT<-expand.grid(Station = unique(COMdf$Station), Parameter = 'BIO',
                      Replicate = paste0('R', 0:6), NbUnit= NA, NbInd=NA,
                      H = NA, Bits = NA, BitsDatum = NA, BitsDatum_Ind=NA)

## Matching order with the output
ABNBIT<-ABNBIT[with(ABNBIT, order(as.character(Replicate),
                                     as.character(Station))), , drop = FALSE ]
BIOBIT<-BIOBIT[with(BIOBIT, order(as.character(Replicate),
                                     as.character(Station))), , drop = FALSE ]

## Loop preparation
## Empty replicate
Arep<-Arep1
Brep<-Brep1
Arep[,]<-0
Brep[,]<-0

## Loop chain
REP<-paste0('R', 1:6)

for (i in REP){
  ## Abundance probability
  ## Drawing each replicate one after the other (from R1 to R6)
  ABNrepTemp<-COMdf %>%
    select(Taxon, Station, Replicate, A) %>%
    filter(Replicate %in% i) %>%
    spread(Taxon, A)

  ## Formatting
  rownames(ABNrepTemp)<-ABNrepTemp$Station
  ABNrepTemp[, c('Station', 'Replicate')]<-NULL

  ## Adding each replicate information (number of individuals) to the previous
  Arep<-Arep+ABNrepTemp

  ## Calculating Shannon entropy after each replicate addition
  H <- diversity(Arep, base=2)

  ## Storing data
  ## How many units of information have been added (individual)
}

```

```

ABNBIT[ABNBIT$Replicate == i, 'NbUnit']<- apply(Arep, 1, sum)
## How many individuals per unit of information
ABNBIT[ABNBIT$Replicate == i, 'NbInd']<- apply(Arep, 1, sum)
## Shannon entropy after each replicate
ABNBIT[ABNBIT$Replicate == i, 'H']<-H
## How much the uncertainty has been decreased by thanks to each replicate
ABNBIT[ABNBIT$Replicate == i,
       'Bits']=log2(length(unique(COMdf$Taxon))) - H
## Uncertainty divided by the number of individuals
ABNBIT[ABNBIT$Replicate == i,
       'BitsDatum']=log2(length(unique(COMdf$Taxon))) / apply(Arep, 1, sum)
## Uncertainty divided by the number individuals
BIOBIT[BIOBIT$Replicate == i,
       'BitsDatum_Ind']=log2(length(unique(COMdf$Taxon))) / apply(Arep, 1, sum)

## Biomass probability
## Drawing each replicate one after the other (from R1 to R6)
BIOrepTemp<-COMdf %>%
  select(Taxon, Station, Replicate, B) %>%
  filter(Replicate %in% i) %>%
  spread(Taxon, B)
## Formatting
rownames(BIOrepTemp)<-BIOrepTemp$Station
BIOrepTemp[, c('Station', 'Replicate')]<-NULL

## Adding each replicate information (biomass) to the previous
Brep<-Brep+BIOrepTemp
## Calculating Shannon entropy after each replicate addition
H <- diversity(Brep, base=2)

## Storing data
## How many units of information have been added (biomass)
BIOBIT[BIOBIT$Replicate == i, 'NbUnit']<- apply(Brep, 1, sum)
## How many individuals per unit of information
BIOBIT[BIOBIT$Replicate == i, 'NbInd']<- apply(Arep, 1, sum)
## Shannon entropy after each replicate
BIOBIT[BIOBIT$Replicate == i, 'H']<-H
## How much the uncertainty has been decreased by thanks to each replicate
BIOBIT[BIOBIT$Replicate == i,
       'Bits']=log2(length(unique(COMdf$Taxon))) - H
## Uncertainty divided by the number of units of information
BIOBIT[BIOBIT$Replicate == i,
       'BitsDatum']=log2(length(unique(COMdf$Taxon))) / apply(Brep, 1, sum)
## Uncertainty divided by the number individuals
BIOBIT[BIOBIT$Replicate == i,
       'BitsDatum_Ind']=log2(length(unique(COMdf$Taxon))) / apply(Arep, 1, sum)
}

## Defining a starting point
ABNBIT[ABNBIT$Replicate == "R0", 'NbInd_Bind']=0
ABNBIT[ABNBIT$Replicate == "R0", 'H']=0
ABNBIT[ABNBIT$Replicate == "R0", 'Bits']=log2(length(unique(COMdf$Taxon)))

```

```

ABNBIT[ABNBIT$Replicate == "R0", 'BitsDatum']<-0
BIOBIT[BIOBIT$Replicate == "R0", 'NbInd_Bind']<-0
BIOBIT[BIOBIT$Replicate == "R0", 'H']<-0
BIOBIT[BIOBIT$Replicate == "R0", 'Bits']<-log2(length(unique(COMdf$Taxon)))
BIOBIT[BIOBIT$Replicate == "R0", 'BitsDatum']<-0

COMBIT<-rbind(ABNBIT, BIOBIT)

COMBITdf<-COMBIT %>%
  gather(Uncertainty, Value, H:BitsDatum)

```

Plot

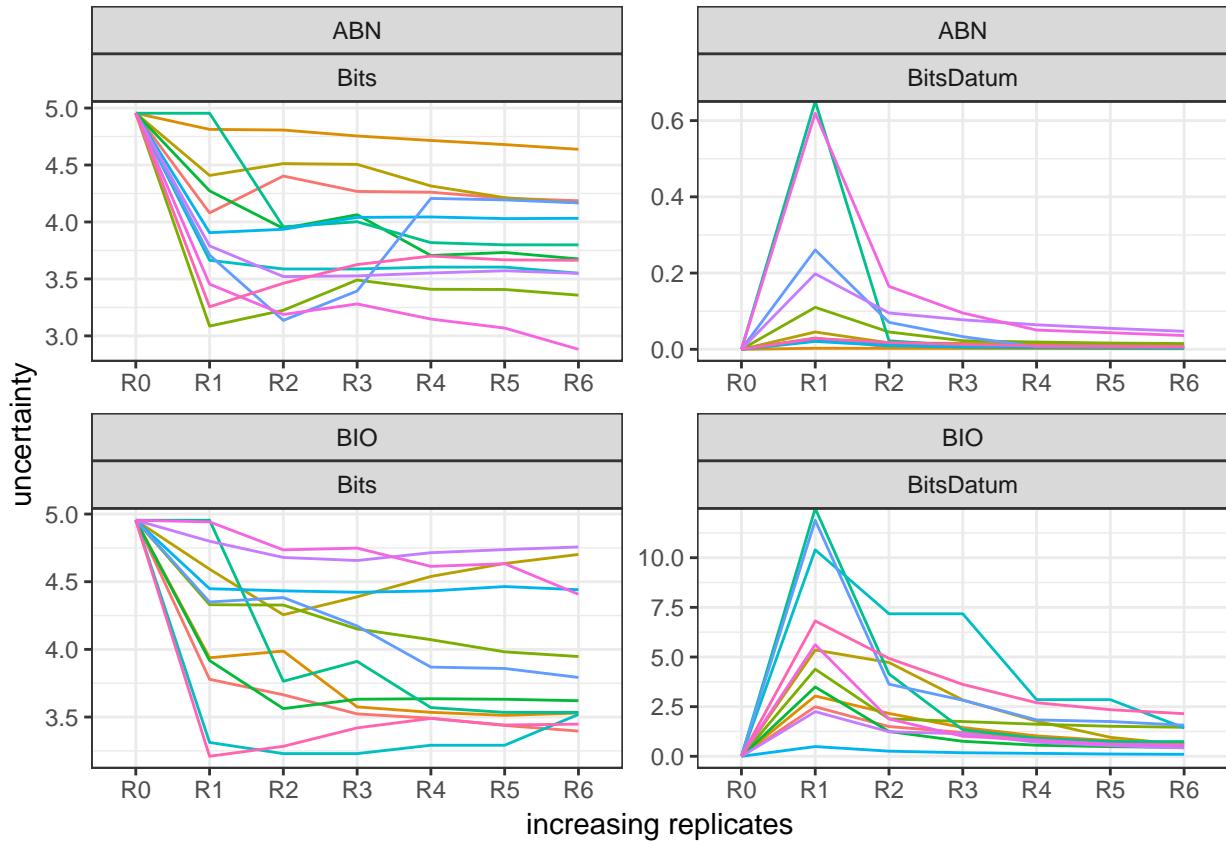


Figure 1: Uncertainty change with real data replication

## Simulated data

We then used the real data to simulate hypothetical replicate for each station. This approach allow to incorporate some probability in the different outcome of what information contain within each replicate *could* have been rather than what it was based on the observed real outcome.

This stochasticity can be used to refine the amount of certainty measured after each of the replicate and can be used on any type of data to be, for example, extrapolated at the scale of a geographical area or sediment type.

## Preparation of the simulations

```
## Step 1 - Allocating probabilities

## Probability of sampling for each of the taxa per station
## We allocate a station & taxa-specific probability of being sampled
## Probability is based on the known number of species, abundance and biomass
## (the 'true' picture)
## The 'truth' is the sum of all replicate
Pcom<- COMdf %>%
  group_by(Station) %>%
  summarise(SumA = sum(A),
            SumB = sum(B)) %>%
  left_join(COMdf, by='Station') %>%
  group_by(Station, Taxon, SumA, SumB) %>%
  summarise(SumTaxaA=sum(A),
            SumTaxaB=sum(B)) %>%
  mutate(Pabn=SumTaxaA/SumA,
        Pbio=SumTaxaB/SumB)

Pcom<-as.data.frame(Pcom)

## Step 2 - Quantity of biological material that a replicate can collect

## Set as the range of number of individual/biomass actually collected per station
## E.g. at HP07 between 12 and 27 individuals (or 0.35 and 2.63 g)
REP<- COMdf %>%
  group_by(Station, Replicate) %>%
  summarise(sumRepA= sum(A),
            sumRepB= sum(B))

## Step 3 - Abundance change is now simulated across a range of increasing replication

## Simulation of Abundance data for each station
## Choice of the station
for (k in unique(REP$Station)){

  ## Output dataframe
  StationSim<-expand.grid(Simulation = 1: 100, Taxon =unique(COMdf$Taxon),
                           Replicate = paste0('R',1:6), simA = 0, simB = 0 )

  ## Number of simulation
  for (l in 1:100){
    cat("sim_count: ", l, "\n")
  }
  ## Selecting the station-specific probability of sampling taxa
  PROBA<-Pcom[Pcom$Station %in% k,]

  ## Selecting the range of number of individuals that each replicate can collect
  PREP<-data.frame(R = paste0('R', 1:6),
                     REPValue = round(runif(6,
                                            min=min(REP[REP$Station%in% k,
                                                       'sumRepA']),
                                            max=REP[REP$Station%in% k, 'sumRepA']))),
  REPBvalue = round(runif(6,
```

```

        min=min(REP$Station%in% k,
                  'sumRepB']),
        max(REP[REP$Station%in% k, 'sumRepB'])),3))

## For each replicate
for (i in PREP$R){
## And the defined number of individuals sampled by replicate
  for (j in 1:PREP[PREP$R %in% i, 'REPAvalue']){

## Draw the species according to their probability
  SELECT<-which(cumsum(Pcom[Pcom$Station %in% k, 'Pabn'])>=runif(1))[1]
  Species<-Pcom[SELECT, 'Taxon']

## Individuals are added in the new simulated data
  StationSim[StationSim$Simulation %in% l &
              StationSim$Taxon %in% Species &
              StationSim$Replicate %in% i,
              'simA']<- StationSim[StationSim$Simulation %in% l &

}
## We also do it for the biomass sampled by replicate (need more thought)
}
}

## That would be too big of a file so we're exporting each simulation
write.csv(StationSim,
          paste0('C:/Users/cg05/Documents/Science/Project - Seedcorn/Proposal/MCZ Trait
               measurement/Concept/Example data/output/SIM',k,'.csv'))
}

```

### Uncertainty change with simulated data

Now that the evolution of the change in abundance have been simulated across the replication using information from real data. We can now calculate the amount of information that each added replication can provide with some probabilities associated to it.

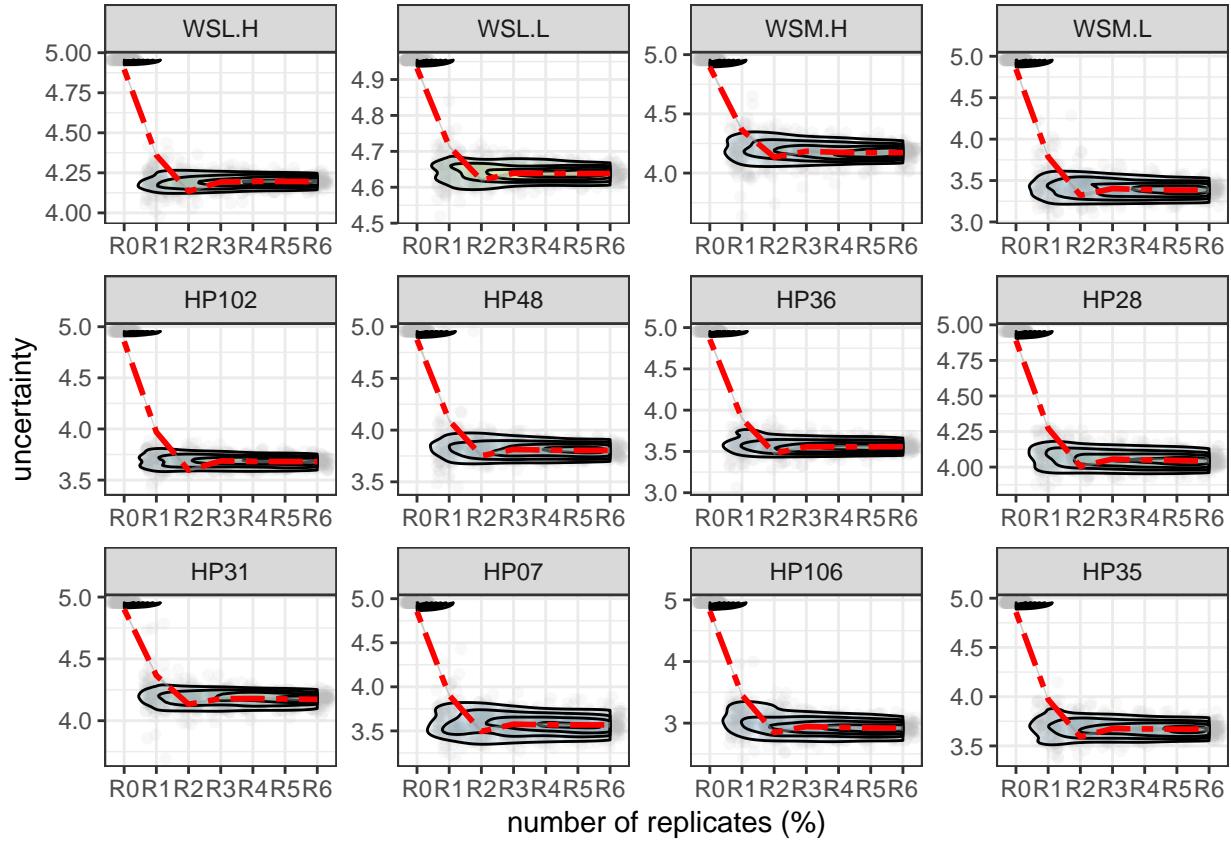


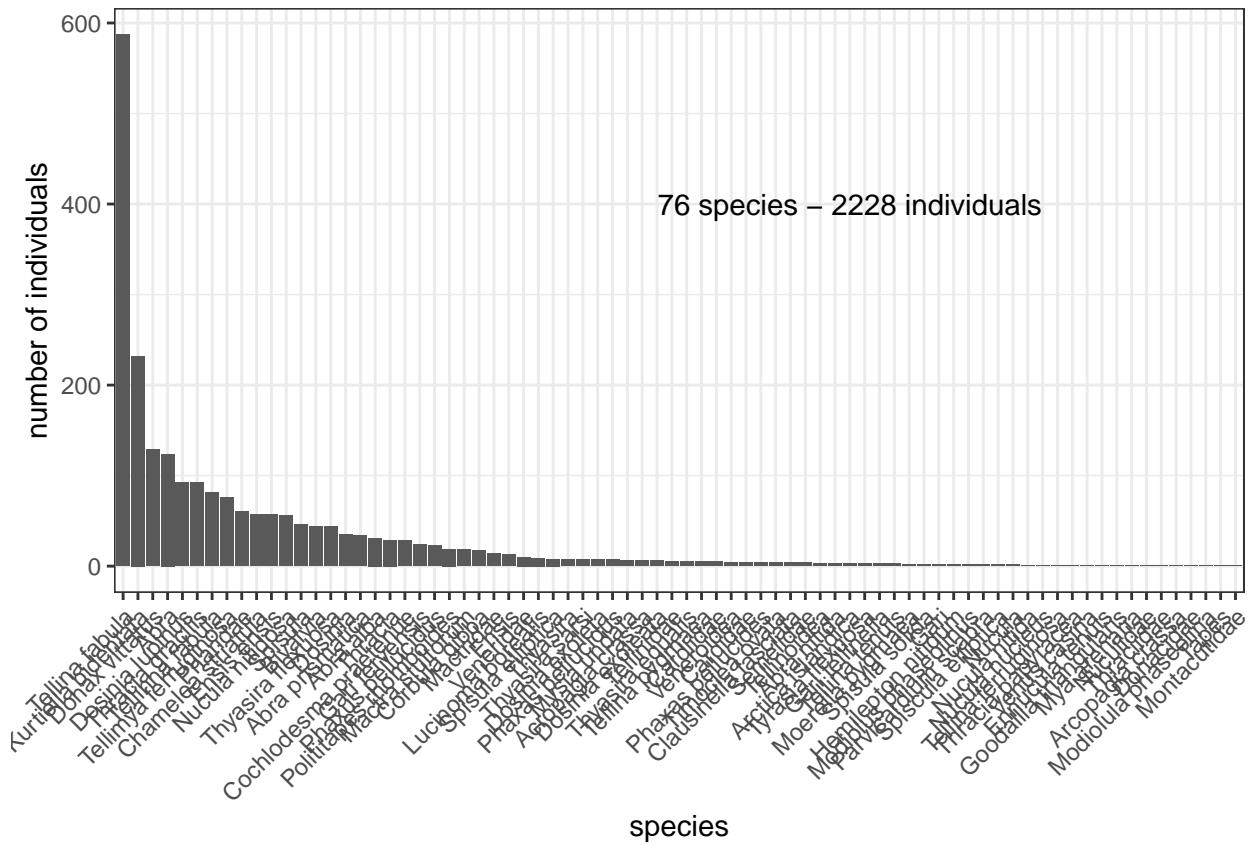
Figure 2: Uncertainty change with simulated data replication

## Individual biomass

Bivalve data from the BACI database (*speak to Paul Mc for more details*).

All species were individually measured including, length, width, depth and weight. We looked at which species were the most abundance overall to develop our approach.

## Data preparation and formatting



With 587 individuals, *Tellina fabula* is the most abundance species overall from the database. As an example, we're going to be using it but the same can be done for all species we'd have individual information for. Size-class is largely skewed toward the small individuals.

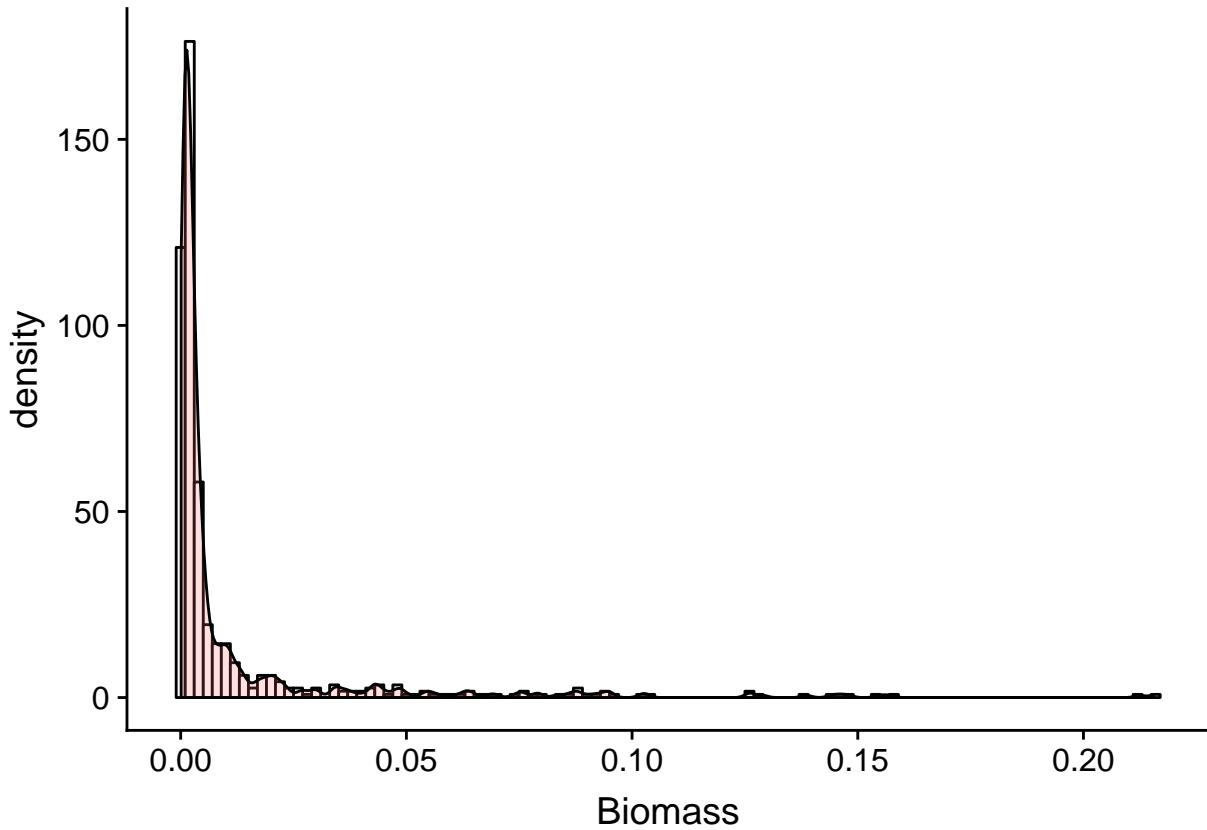
```
INTACT<-BACI %>%
  filter(State == 'Intact' & !is.na(Species))

Tfab<-INTACT %>%
  filter(Species == 'Tellina fabula' & !is.na(Biomass))

Tfab[is.na(Tfab$Biomass)]

## data frame with 0 columns and 587 rows

ggplot(Tfab, aes(x=Biomass)) +
  geom_histogram(aes(y=..density..),
                binwidth=.002,
                colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666")
```



```

set.seed(2)

## Output tibble
BIT<-expand.grid(PERC = c(1:100), Simulation = 1:500, NbInd=NA, sumHi = NA,
                  Bits = NA, BitsDatum = NA)

## Range of biomass value (assuming maximum information)
## Not to be changed throughout the simulation (105 bins)
RANGE<-Tfab %>%
  sample_n(size= round((100*nrow(Tfab))/100)) %>%
  mutate(gr=cut(Biomass, breaks= seq(0, max(Tfab$Biomass)+0.0004, by = 0.0005))) %>%
  group_by(gr) %>%
  summarise(n=n())

## Loop from 100 down to 1% of the total amount of data
for (j in 1:500){
  cat("sim_count: ", j, "\n")

  for (i in 1:100){
    Tfab %>%
      # Sample the % of data
      sample_n(size= round((i*nrow(Tfab))/100), replace=T) %>%

      # Allocate the correct bins of the subset
      mutate(gr=cut(Biomass,
                    breaks= seq(0, max(Tfab$Biomass)+0.0004,

```

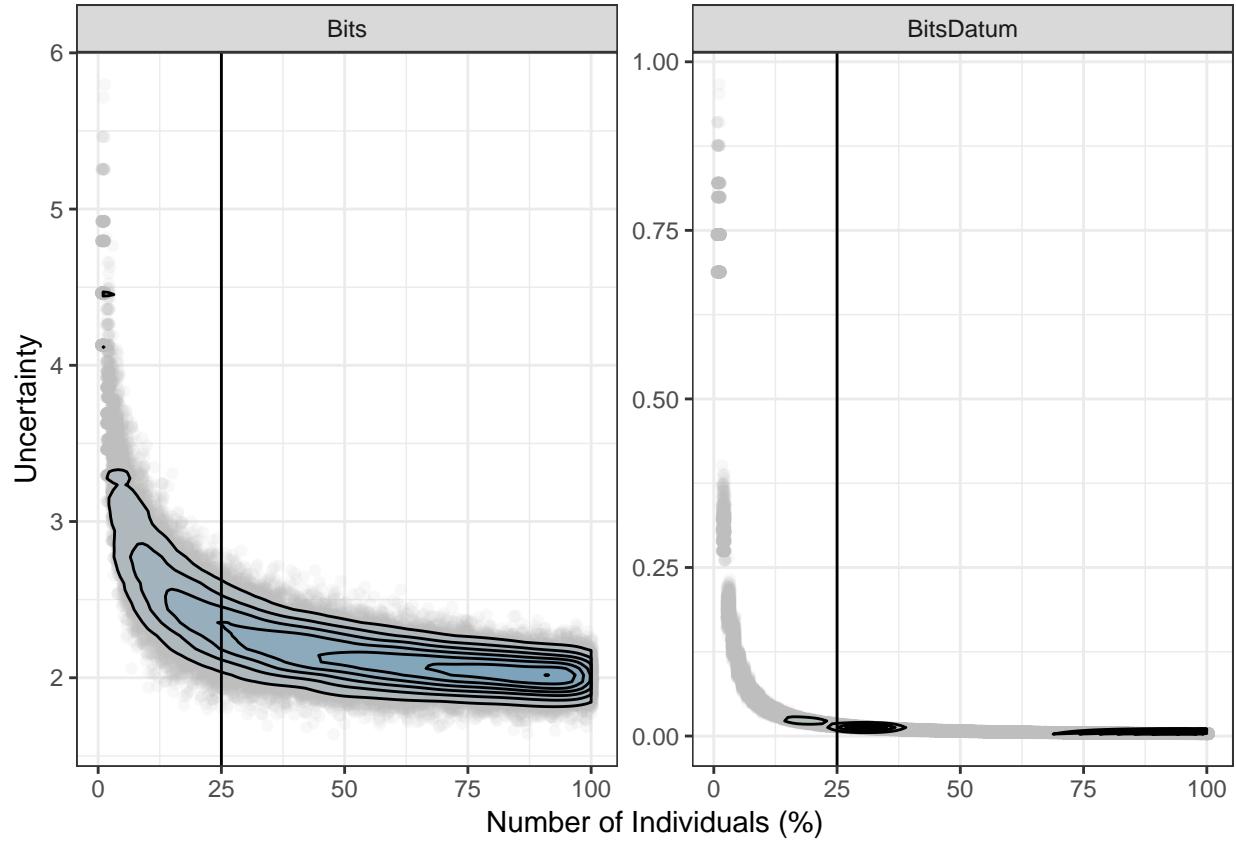
```

        by = 0.0005))) -> TEMP

# Relate the fixed number of bins
TEMP<-full_join(RANGE[, 'gr'], TEMP, by='gr')
TEMP %>%
  # Compute number of rows per bins (if individuals -> 1, if NA -> 0)
  mutate(n = ifelse(is.na(Biomass) == T, 0, 1)) %>%
  group_by(gr) %>%
  summarise(N=sum(n)) %>%
  # Compute pi (probability of outcome) value for each bin
  mutate(pi = N/round((i*nrow(Tfab))/100)) %>%
  # Compute Hi (degree of uncertainty) value for each bin
  mutate(Hi = ifelse(pi>0, -pi * log2(pi), 0))->TEMP

RANGE[,paste0(i, 'Perc')]<- TEMP$pi
BIT[BIT$PERC == i & BIT$Simulation == j,
  'NbInd']<- round((i*nrow(Tfab))/100)
BIT[BIT$PERC == i & BIT$Simulation == j,
  'sumHi']<-sum(TEMP$Hi)
BIT[BIT$PERC == i & BIT$Simulation == j,
  'Bits']<-(log2(105)) - sum(TEMP$Hi)
BIT[BIT$PERC == i & BIT$Simulation == j,
  'BitsDatum']<-((log2(105)) - sum(TEMP$Hi))/sum(TEMP$N)
}
}
write.csv(BIT, 'C:/Users/cg05/Documents/Science/Project - Seedcorn/Proposal/MCZ Trait
measurement/Concept/Example data/output/BIT_biomass.csv')

```

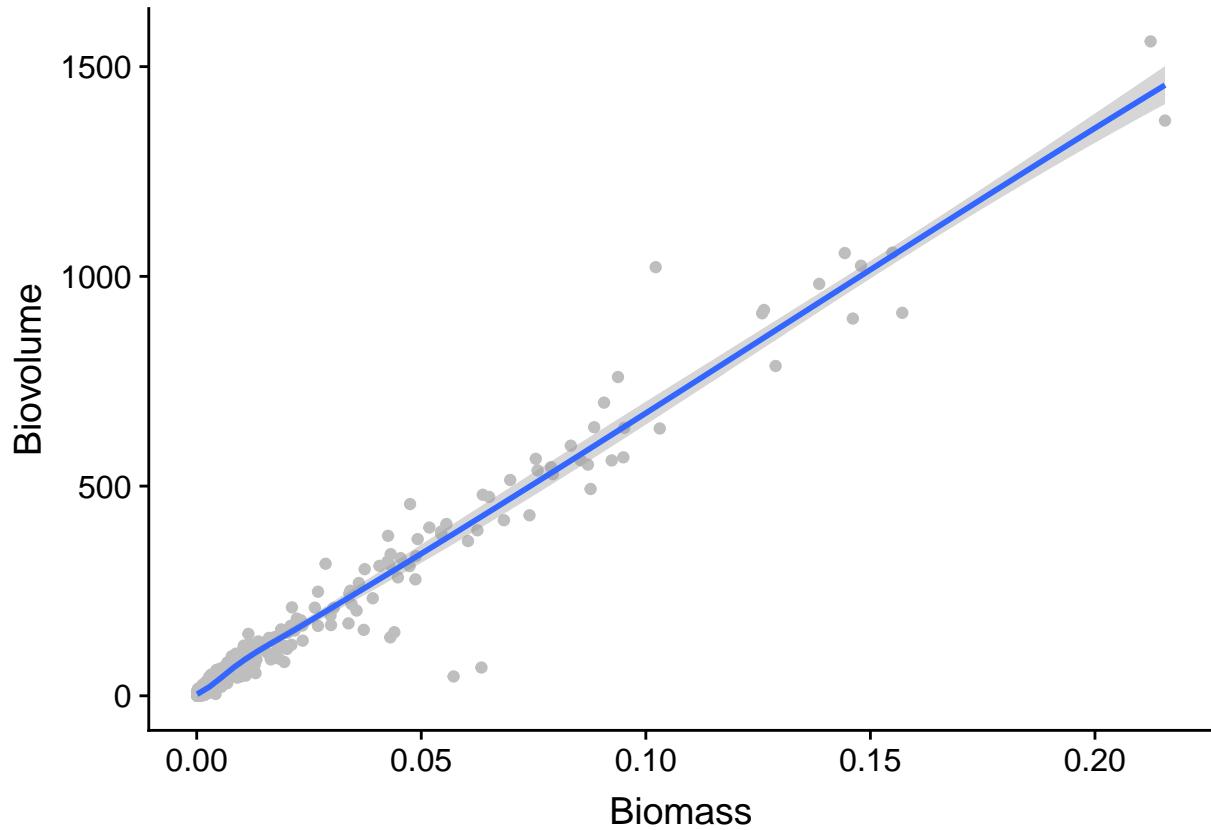


Relationship between individual biomass and volume. Assuming that the bivalve is a cylinder with an elliptic shape rather than a circle, the formula to get the volume is: ellipse area x length:  $(\pi \times \text{depth} \times \text{width}) \times \text{length}$

```
## Biovolume calculation
Tfab$Biovolume<- (Tfab$Width*Tfab$Depth*pi)*Tfab$Length

## Correlation
ggplot(Tfab, aes(Biomass, Biovolume)) +
  geom_point(colour="grey")+
  stat_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
## Warning: Removed 1 rows containing missing values (geom_point).
```



```

bioVmod<-lm(Biomass~Biovolume, data=Tfab)

## Model parameters
summary(bioVmod)

##
## Call:
## lm(formula = Biomass ~ Biovolume, data = Tfab)
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -0.044315 -0.000949 -0.000316  0.000548  0.053825 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -9.894e-05  2.290e-04 -0.432   0.666    
## Biovolume    1.435e-04  1.126e-06 127.387  <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005053 on 584 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9653, Adjusted R-squared:  0.9652 
## F-statistic: 1.623e+04 on 1 and 584 DF,  p-value: < 2.2e-16 

BACIA<-read.csv(paste0('C:/Users/cg05/Documents/Science/Project - Seedcorn/Proposal',
                       '/MCZ Trait measurement/Concept/Example data/input/BACIA.csv'),
```

```

na.strings= c(' ', "", NA))

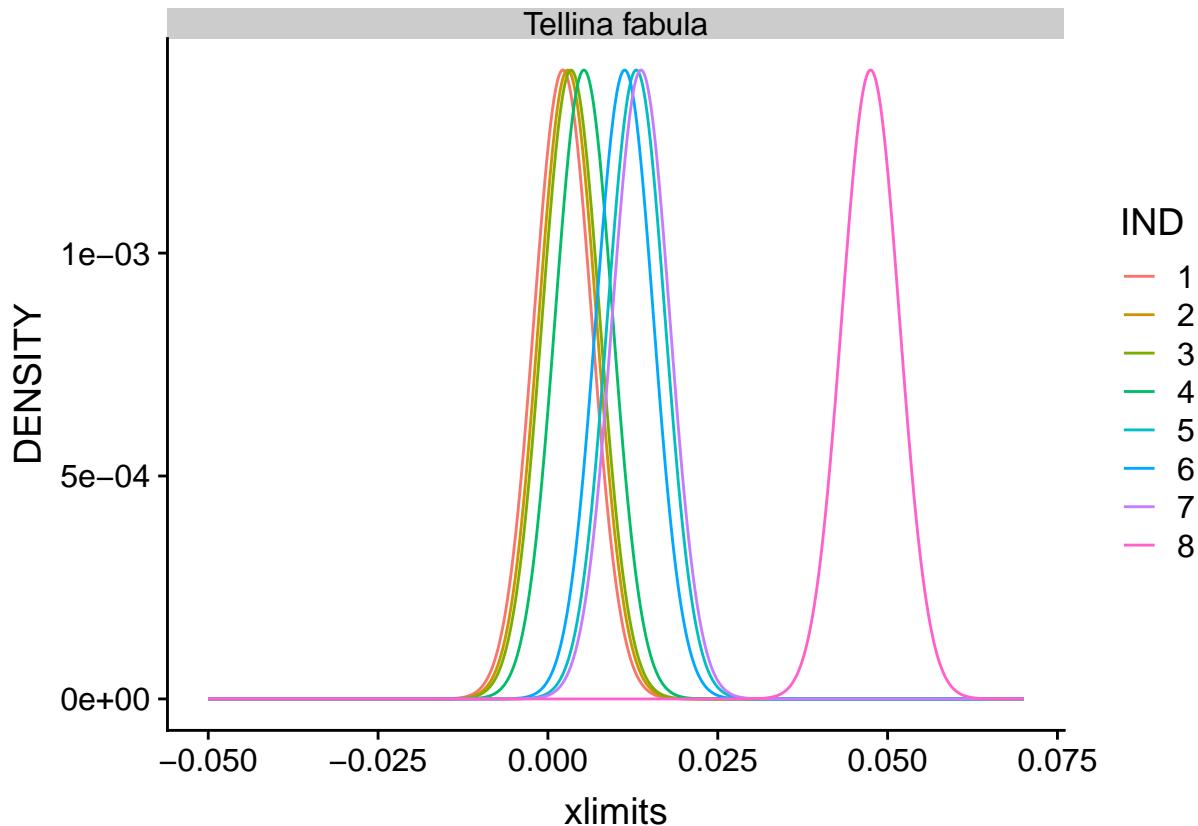
## Station A012 - Tellina fabula
A012<-BACIA[BACIA$StationName %in% "A012",]

xlimits=seq(from=-0.05, to=0.07, length=1000)
output<-expand.grid(Taxa = unique(A012$Species),
                     IND=factor(c(1:max(summary(A012$Species)))),
                     xlimits=xlimits, DENSITY=NA)
output <- output[with(output, order(Taxa, IND, xlimits)), , drop = FALSE ]

for(j in unique(A012$Species)){
  IND<-A012[A012$Species %in% j, "Biomass"]
  for (i in 1:length(IND)){
    output[output$Taxa %in% j & output$IND %in% i,
           "DENSITY"]<-dnorm(xlimits,
                               mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                       hpi(c(IND, IND+0.01))))/(sum(dnorm(xlimits,
                               mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                       hpi(c(IND, IND+0.01)))))*length(IND))
  }
}

ggplot(output[output$Taxa %in% 'Tellina fabula'], aes(x=xlimits, y=DENSITY, colour=IND)) +
  geom_line()+
  facet_wrap(~Taxa)

```

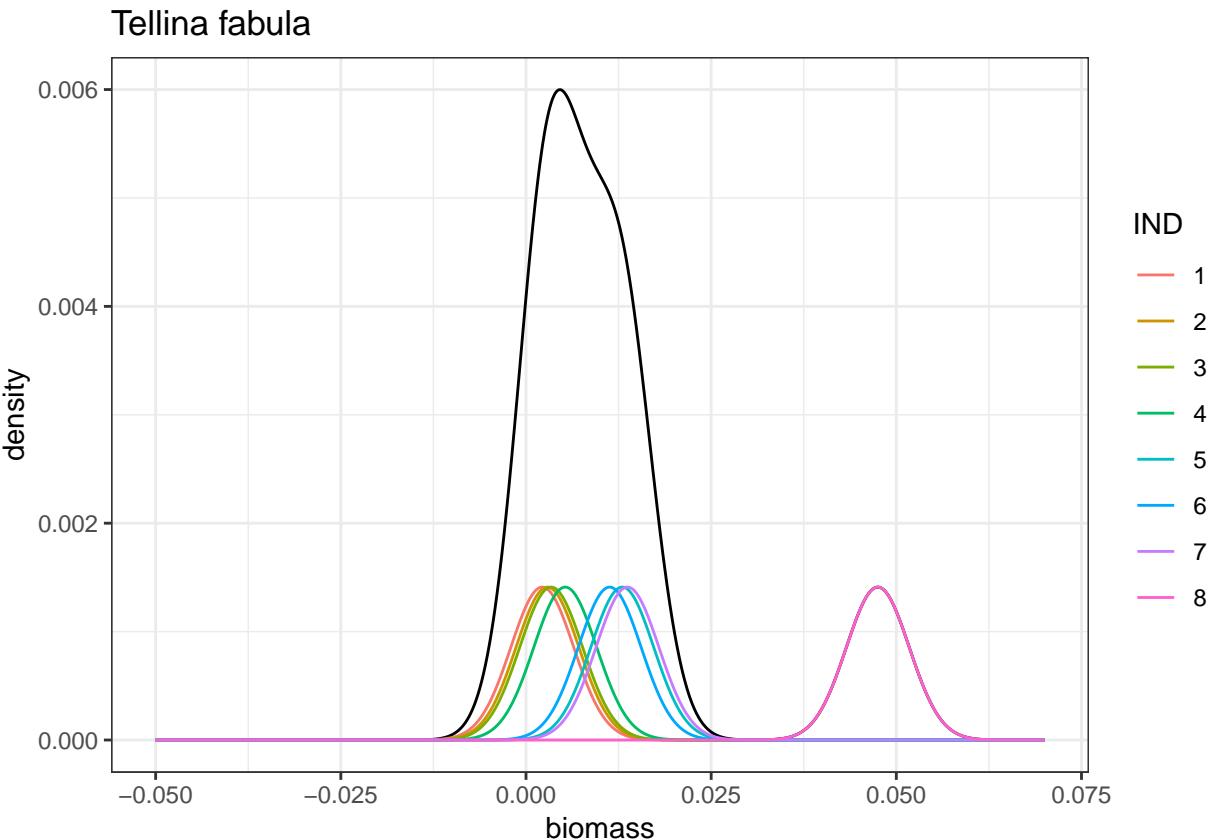


```

SPdens<-output[output$Taxa %in% 'Tellina fabula',] %>%
  group_by(xlimits,Taxa) %>%
  summarise(sumDENS=sum(DENSITY))

ggplot(SPdens, aes(x=xlimits, y=sumDENS)) +
  geom_line()+
  geom_line(data=output[output$Taxa %in% 'Tellina fabula',],
            aes(x=xlimits, y=DENSITY, colour=IND))+ 
  scale_y_continuous(name = 'density')+ 
  scale_x_continuous(name = 'biomass')+ 
  theme_bw()+
  ggtitle('Tellina fabula')

```



```

## Deriving biovolume, respiration and filtering rate
# biovolume = Biomass*1.435e-04 - -9.894e-05
# Filtering rate (in ml/h) F = a*W exp b (Winter 1973)
# Where at 12 deg c (a=2410 & b = 0.73)
# Respiration 0.78-0.88 mg O2/g/h (Landes et al. 2015)
outputTfab<-output[output$Taxa %in% 'Tellina fabula',]

head(outputTfab)

##           Taxa IND      xlims      DENSITY
## 4   Tellina fabula  1 -0.05000000 2.178467e-36
## 44  Tellina fabula  1 -0.04987988 3.083079e-36
## 84  Tellina fabula  1 -0.04975976 4.359841e-36
## 124 Tellina fabula  1 -0.04963964 6.160405e-36
## 164 Tellina fabula  1 -0.04951952 8.697619e-36
## 204 Tellina fabula  1 -0.04939940 1.226998e-35

# Can't have negative biomass - need to sort it later
# In the meantime, I'll just move the value across the range
outputTfab$biomass<-outputTfab$xlimits + 0.06
outputTfab$biovolume<-outputTfab$biomass*1.435e-04 - -9.894e-05
outputTfab$filtering<-2410*outputTfab$biomass*exp(0.73)
outputTfab$respiration<-0.83*outputTfab$biomass

outputTfabdf<-outputTfab %>%
  gather(physiology, value, biomass:respiration)

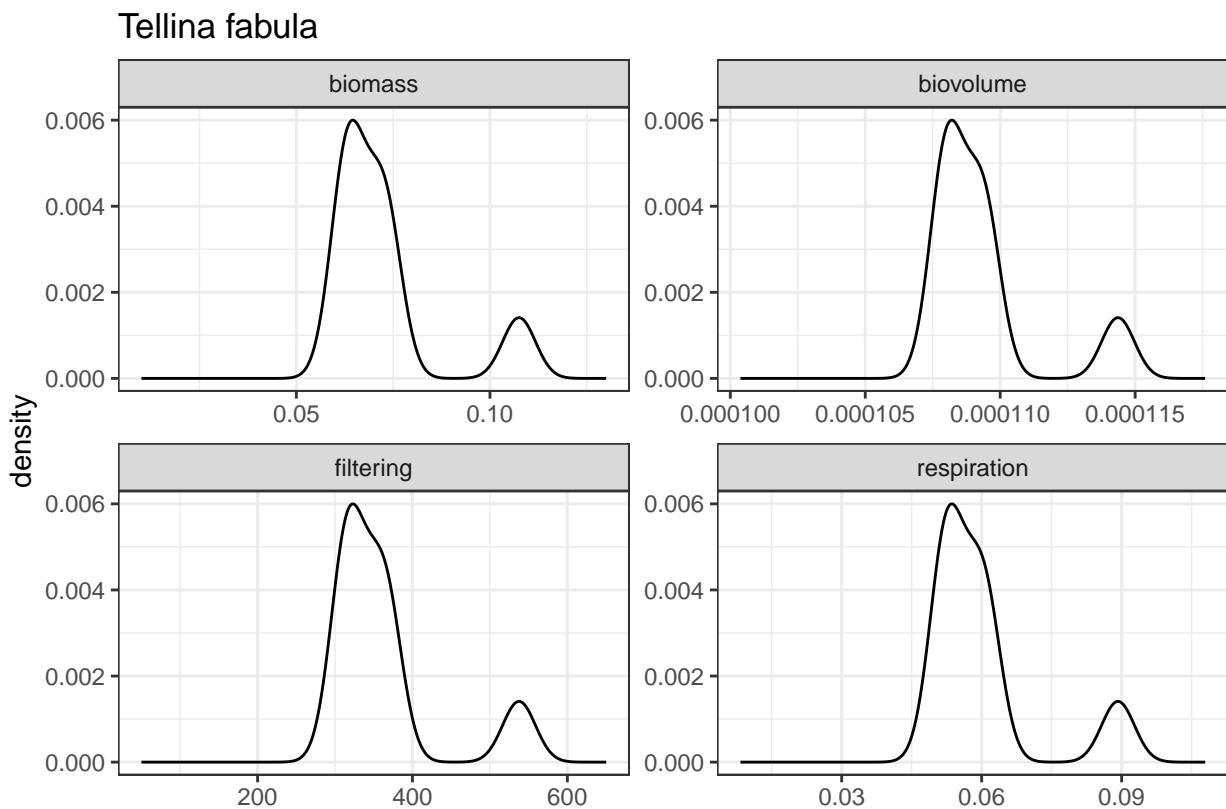
```

```

SPdens<-outputTfabdf %>%
  group_by(physiology, value, Taxa) %>%
  summarise(sumDENS=sum(DENSITY))

ggplot(SPDens, aes(x=value, y=sumDENS)) +
  geom_line()+
  scale_y_continuous(name = 'density')+
  scale_x_continuous(name = '')+
  theme_bw()+
  facet_wrap(~physiology, scale = 'free')+
  ggtitle('Tellina fabula')

```



```

A017<-BACIA[BACIA$StationName %in% "A017",]

xlimits=seq(from=-0.05, to=0.07, length=1000)
output<-expand.grid(Taxa = unique(A017$Species),
                      IND=factor(c(1:max(summary(A017$Species)))),
                      xlimits=xlimits, DENSITY=NA)
output <- output[with(output, order(Taxa, IND, xlimits)), , drop = FALSE ]

for(j in unique(A017$Species)){
  IND<-A017[A017$Species %in% j, "Biomass"]
  for (i in 1:length(IND)){
    output[output$Taxa %in% j & output$IND %in% i,

```

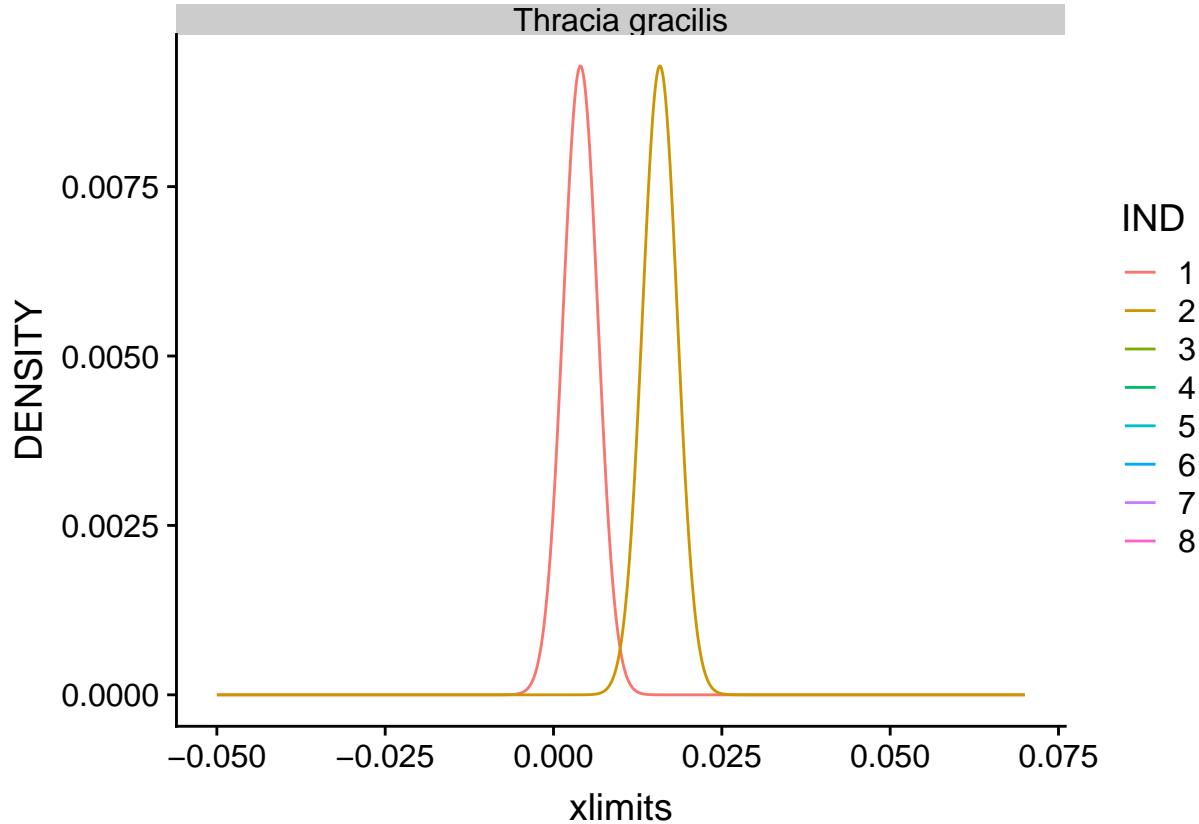
```

    "DENSITY"]<-dnorm(xlimits,
mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                           hpi(c(IND, IND+0.01)))/(sum(dnorm(xlimits,
mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                           hpi(c(IND, IND+0.01)))))*length(IND))
}
}

ggplot(output[output$Taxa %in% 'Thracia gracilis',], aes(x=xlimits, y=DENSITY, colour=IND)) +
  geom_line()+
  facet_wrap(~Taxa)

```

## Warning: Removed 6000 rows containing missing values (geom\_path).



```

SPdens2<-output[output$Taxa %in% 'Thracia gracilis',] %>%
  group_by(xlimits, Taxa) %>%
  summarise(sumDENS=sum(DENSITY, na.rm=T))

ggplot(SPDens2, aes(x=xlimits, y=sumDENS)) +
  geom_line(data=output[output$Taxa %in% 'Thracia gracilis',],
            aes(x=xlimits, y=DENSITY, colour=IND))+ 
  geom_line()+
  geom_line(data=output[output$Taxa %in% 'Thracia gracilis',],
            aes(x=xlimits, y=DENSITY, colour=IND))+ 
  scale_y_continuous(name = 'density')+ 
  scale_x_continuous(name = 'biomass')+

```

```

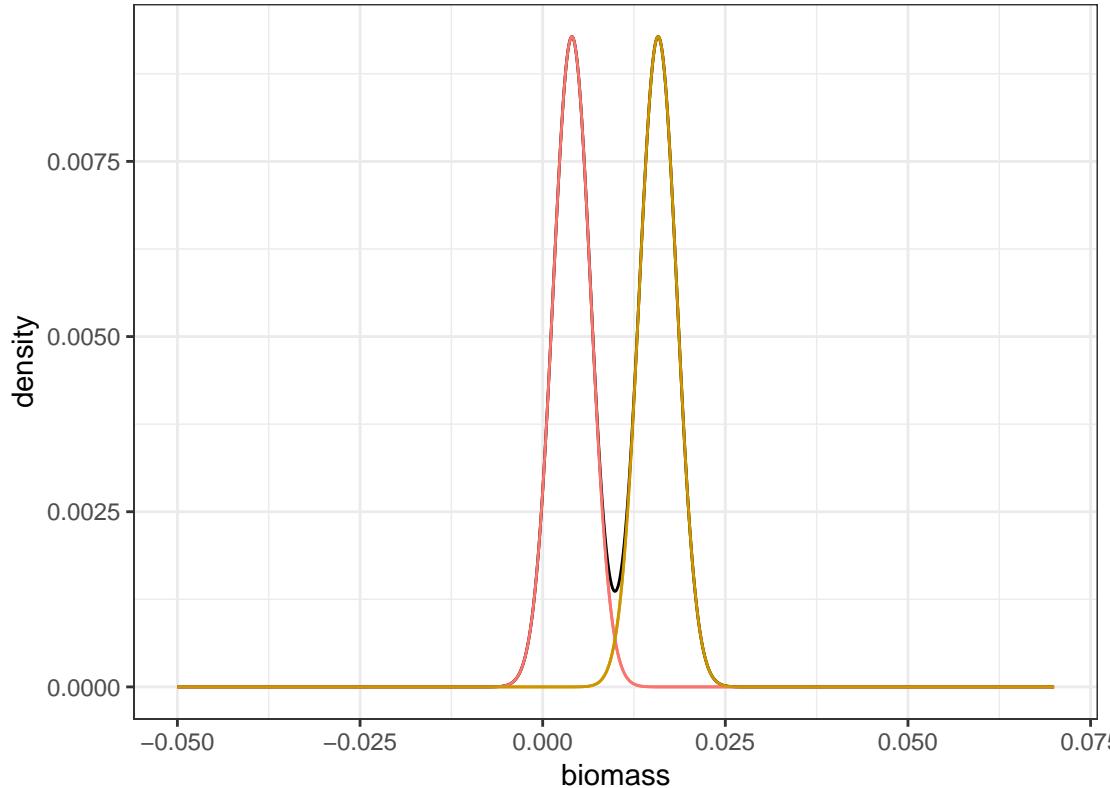
theme_bw()+
ggtitle('Thracia gracilis')

## Warning: Removed 6000 rows containing missing values (geom_path).

## Warning: Removed 6000 rows containing missing values (geom_path).

```

### Thracia gracilis



```

## Tellina fabula
xlimits=seq(from=-0.05, to=0.07, length=1000)
output<-expand.grid(Taxa = unique(A012$Species),
                     IND=factor(c(1:max(summary(A012$Species)))),
                     xlimits=xlimits, DENSITY=NA)
output <- output[with(output, order(Taxa, IND, xlimits)), , drop = FALSE ]

for(j in unique(A012$Species)){
  IND<-A012[A012$Species %in% j, "Biomass"]
  for (i in 1:length(IND)){
    output[output$Taxa %in% j & output$IND %in% i,
          "DENSITY"]<-dnorm(xlimits,
                             mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                     hpi(c(IND, IND+0.01))))/(sum(dnorm(xlimits,
                             mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                     hpi(c(IND, IND+0.01)))))*length(IND))
  }
}

```

```

SPdens<-output[output$Taxa %in% 'Tellina fabula',] %>%
  group_by(xlimits,Taxa) %>%
  summarise(sumDENS=sum(DENSITY))

## Thracia gracilis
A017<-BACIA[BACIA$StationName %in% "A017",]
summary(A017)

##     Sector   StationName   StationNb      LabRef
##  BACIA:12    A017    :12   Min.   :69   Min.   :61991
##                 A002    : 0   1st Qu.:69   1st Qu.:61991
##                 A003    : 0   Median  :69   Median  :61991
##                 A004    : 0   Mean    :69   Mean    :61991
##                 A006    : 0   3rd Qu.:69   3rd Qu.:61991
##                 A007    : 0   Max.    :69   Max.    :61991
##                 (Other): 0
##           Species       Stage      State      Length
## Tellina fabula    :8   Adult    :12   Damaged: 0   Min.   : 2.100
## Kurtiella bidentata:2 Juvenile: 0   Intact  :12   1st Qu.: 2.700
## Thracia gracilis  :2
## Bivalvia          :0
## Chamelea striatula:0
## Corbula gibba    :0
## (Other)           :0
##       Width       Depth      Biomass      Level
## Min.   :1.100   Min.   :0.2000   Min.   :0.001200   Class   : 0
## 1st Qu.:1.600   1st Qu.:0.2750   1st Qu.:0.001650   Family  : 0
## Median :1.800   Median :0.4500   Median :0.002300   Genus   : 0
## Mean   :2.250   Mean   :0.6083   Mean   :0.008183   Species :12
## 3rd Qu.:2.125   3rd Qu.:0.6250   3rd Qu.:0.003400   Superfamily: 0
## Max.   :6.000   Max.   :1.9000   Max.   :0.060400
##
##limits=seq(from=-0.05, to=0.07, length=1000)
output<-expand.grid(Taxa = unique(A017$Species),
                     IND=factor(c(1:max(summary(A017$Species)))),
                     xlimits=xlimits, DENSITY=NA)
output <- output[with(output, order(Taxa, IND, xlimits)), , drop = FALSE ]

for(j in unique(A017$Species)){
  IND<-A017[A017$Species %in% j, "Biomass"]
  for (i in 1:length(IND)){
    output[output$Taxa %in% j & output$IND %in% i,
          "DENSITY"]<-dnorm(xlimits,
                             mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                       hpi(c(IND, IND+0.01))))/(sum(dnorm(xlimits,
                             mean=IND[i], sd=ifelse(length(IND)>1, hpi(IND),
                                                       hpi(c(IND, IND+0.01)))))*length(IND))
  }
}

SPdens2<-output[output$Taxa %in% 'Thracia gracilis',] %>%
  group_by(xlimits, Taxa) %>%
  summarise(sumDENS=sum(DENSITY, na.rm=T))

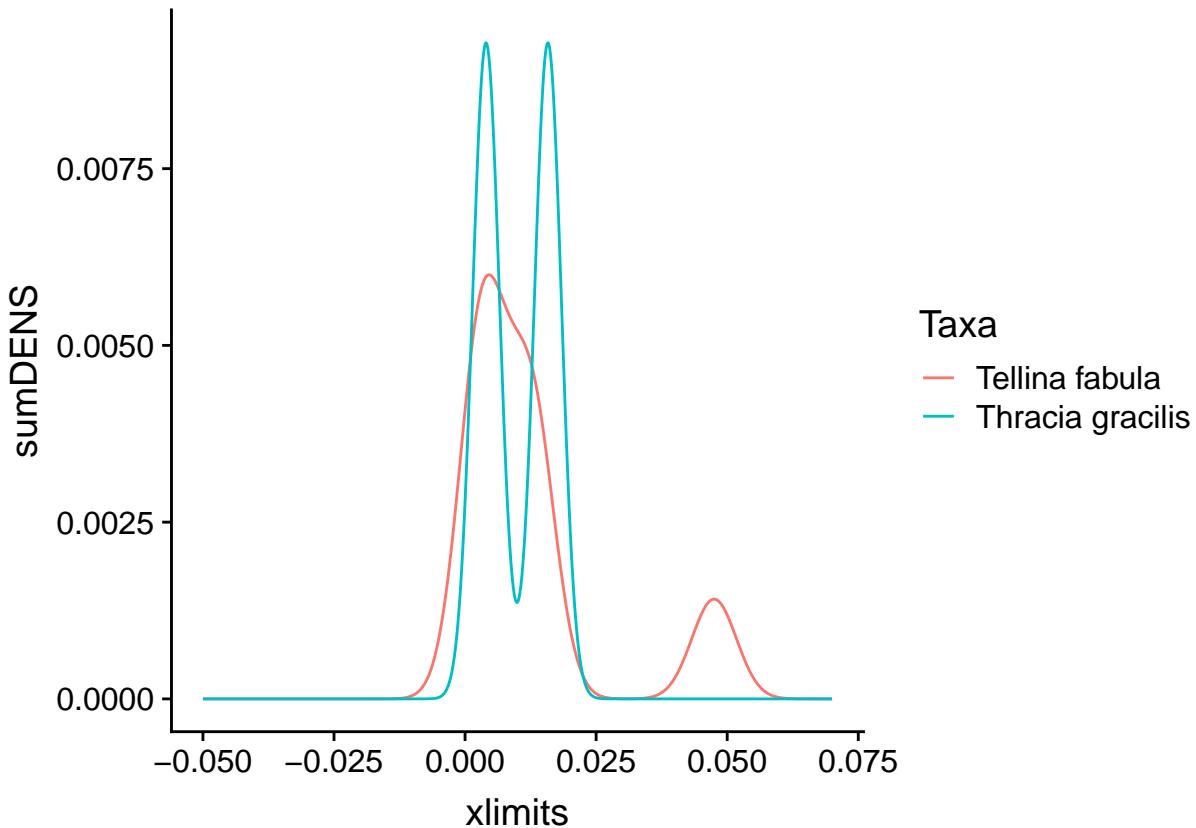
```

```

## Combining species
VIRTcom<-rbind(SPdens, SPdens2)

ggplot(VIRTcom, aes(x=xlimits, y=sumDENS, colour=Taxa)) +
  geom_line()

```



```

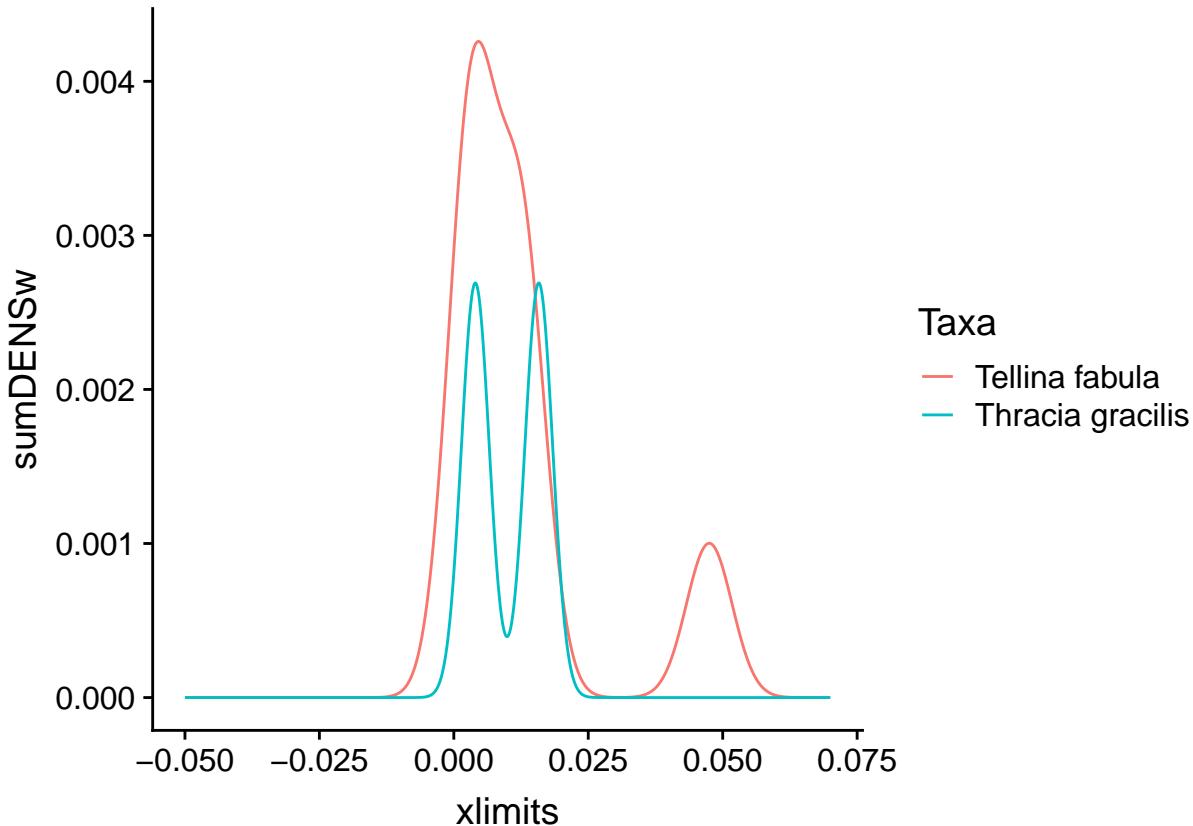
## Proportion of abundance
# 5 Tellina fabula (0.71) and 2 Thracia gracilis (0.29)
VIRTcom[VIRTcom$Taxa %in% "Tellina fabula",
         'sumDENSw']<-VIRTcom[VIRTcom$Taxa %in% "Tellina fabula", 'sumDENS']*0.71

VIRTcom[VIRTcom$Taxa %in% "Thracia gracilis",
         'sumDENSw']<-VIRTcom[VIRTcom$Taxa %in% "Thracia gracilis", 'sumDENS']*0.29

# sum of the weighted proportion
sumVIRTcom<-VIRTcom %>%
  group_by(xlimits) %>%
  summarise(DENSw = sum(sumDENSw))

ggplot(VIRTcom, aes(x=xlimits, y=sumDENSw, colour=Taxa, fill=Taxa)) +
  geom_line()

```



```
test<-cbind(sumVIRTcom,
            VIRTcom[VIRTcom$Taxa %in% "Tellina fabula", 'sumDENSw'],
            VIRTcom[VIRTcom$Taxa %in% "Thracia gracilis", 'sumDENSw'])
colnames(test)<-c('Biomass', 'DENSw', 'Tellina', 'Thracia')
head(test)
```

```
##      Biomass      DENSw     Tellina     Thracia
## 1 -0.0500000 1.794088e-36 1.794088e-36 2.585518e-98
## 2 -0.04987988 2.540944e-36 2.540944e-36 6.835968e-98
## 3 -0.04975976 3.595841e-36 3.595841e-36 1.803483e-97
## 4 -0.04963964 5.084637e-36 5.084637e-36 4.747707e-97
## 5 -0.04951952 7.184119e-36 7.184119e-36 1.247141e-96
## 6 -0.04939940 1.014241e-35 1.014241e-35 3.268937e-96
test$TellinaProp<-(test$Tellina/test$DENSw)*test$DENSw
test$ThraciaProp<-(test$Thracia/test$DENSw)*test$DENSw
```

```
ggplot(test, aes(x=Biomass, y=DENSw)) +
  geom_line()+
  scale_y_continuous(name = 'density')+
  scale_x_continuous(name = 'biomass')+
  geom_area(aes(x=Biomass, y=TellinaProp+ThraciaProp, fill='red'))+
  geom_area(aes(x=Biomass, y=TellinaProp, fill='blue'))+
  scale_fill_discrete(name="species",
                      breaks=c("red","blue"),
```

```
labels=c("Thracia", "Tellina"))
```

