Trait-based products of EMODnet benthic biology

Karline Soetaert, Sarah O' Flynn, Olivier Beauchard, Peter Herman

9 October 2018

Abstract

Aim of this analysis.

Through their activity, benthic animals play an important role in marine ecosystem functioning. More specifically, they mix the sediment by their movement and feeding, a process called "bioturbation". They also create water movements, enhancing the exchange of dissolved constituents such as oxygen, and dissolved inorganic nutrients, a process called "bio-irrigation".

Both these activities have a large impact on the biogeochemical cycles in the environment, and they are commonly parameterised as single parameters in biogeochemical models.

Biologists have tried to categorise the bioturbation or bio-irrigation activity based on the identity of the organims. They do not derive the rate parameters as used in the biogeochemical models but rather derive a potential of the organisms to perform these tasks.

Here we use Benthic abundance data from EMODnet to estimate the bioturbation potential and the bioirrigation potential index.

To derive these potentials we need information on:

- The weights of the species and their total biomass
- The species mobility and sediment reworking mode

Reading the data

Species taxonomic tree

The names of all species encountered in the dataset were checked against the WORMS database, and their taxonomic tree added (this was done via http://www.marinespecies.org/, menu item *tools/Match taxa*). These data are read first. The dimensions of the data set and the first two entries are printed.

```
Taxo     <- read.csv(file= "taxo.csv")
cat("\ndimension and first part of the data set : \n")
```

```
##
## dimension and first part of the data set :
```

```
dim(Taxo)
```

```
## [1] 7650    8
```

```
head(Taxo, n = 2)
```

```
##     id          phy          clas                         ord           fam
## 1 7500 Hemichordata Enteropneusta [unassigned] Enteropneusta
## 2    1     Cnidaria      Hydrozoa               Leptothecata Sertulariidae
##            gen                          tx                          txa
```

```
## 1 [unassigned] [unassigned] Enteropneusta [unassigned] Enteropneusta
## 2  Abietinaria                Abietinaria                Abietinaria
```

the MWTL data

The MWTL data is the only data set that contains both species *biomass* and *densities* (the other data sets comprise only densities or just presence/absence). This data is used to estimate mean individual weights of the various species. The file *occ.csv* has been prepared by Olivier Beauchard as part of the EMODNET product on trait types. It contains average density and biomass per taxon and station. Averaging was over time. Some taxa lumping has been done in the preparation of the file, as well as taxonomic checks against WORMS.

Biomass is in *g AFDW/m2*, density in *number/m2*

```
MWTLdata <- read.csv("occ.csv", header=TRUE, stringsAsFactors = FALSE)

# calculate average weight per species and sample
MWTLdata$mw <- MWTLdata$biom/MWTLdata$ind
MWTLdata$lmw <- log(MWTLdata$mw)

# calculate arithmetic and geometric mean per species over all samples where the species occurs
spmweight <- aggregate(mw~tax,MWTLdata,FUN=mean)
spgmweight <- aggregate(lmw~tax,MWTLdata,FUN=mean)
spgmweight$gmw <- exp(spgmweight$lmw)
n <- aggregate(mw~tax, MWTLdata, FUN=function(x) length(x))
names(n) <- c("tax","n")

mweights <- merge(spmweight, spgmweight, by="tax")
mweights <- merge(mweights, n, by="tax")
mweights <- mweights[,-3]
write.csv(mweights, file="mweights.csv")
```

```
mws.mwtl <- read.csv("mweights.csv")
cat("\ndimension and first part of the data set : \n")
```

```
##
## dimension and first part of the data set :
```

```
dim(mws.mwtl)
```

```
## [1] 394   5
```

```
head(mws.mwtl, n = 2)
```

```
##   X                   tax           mw          gmw   n
## 1 1 Abludomelita obtusata 0.0003002036 0.0003002011  12
## 2 2             Abra alba 0.0028563437 0.0004744354 425
```

```
colnames(mws.mwtl)[2:3]   <- c("tx", "avweight")
mws.all <- merge(mws.mwtl, Taxo, by = "tx")
```

Expanding the weight dataset

Next we estimate mean weights on species level, on genus level, on the family level, and on the level of order and bind all in one data.frame(*mws*).

```
mws.species <- data.frame(meanW = tapply(mws.all$avweight, INDEX = mws.all$tx, FUN = mean, na.rm = TRUE)
                          sdW   = tapply(mws.all$avweight, INDEX = mws.all$tx, FUN = sd,   na.rm = TRUE)
                          n     = tapply(mws.all$avweight, INDEX = mws.all$tx, FUN = length),
                          level = "species")

mws.genus   <- data.frame(meanW = tapply(mws.all$avweight, INDEX = mws.all$gen, FUN = mean, na.rm = TRU
                          sdW   = tapply(mws.all$avweight, INDEX = mws.all$gen, FUN = sd,   na.rm = TRU
                          n     = tapply(mws.all$avweight, INDEX = mws.all$gen, FUN = length),
                          level = "genus")
mws.genus   <- mws.genus[-which(rownames(mws.genus)==""),]
mws.genus   <- subset(mws.genus, !is.na(meanW))


mws.family  <- data.frame(meanW = tapply(mws.all$avweight, INDEX = mws.all$fam, FUN = mean, na.rm = TRU
                          sdW   = tapply(mws.all$avweight, INDEX = mws.all$fam, FUN = sd,   na.rm = TRU
                          n     = tapply(mws.all$avweight, INDEX = mws.all$fam, FUN = length),
                          level = "family")
mws.family  <- mws.family[-which(rownames(mws.family)==""),]
mws.family  <- subset(mws.family, !is.na(meanW))

mws.order   <- data.frame(meanW = tapply(mws.all$avweight, INDEX = mws.all$ord, FUN = mean, na.rm = TRU
                          sdW   = tapply(mws.all$avweight, INDEX = mws.all$ord, FUN = sd,   na.rm = TRU
                          n     = tapply(mws.all$avweight, INDEX = mws.all$ord, FUN = length),
                          level = "order")
mws.order   <- mws.order[-which(rownames(mws.order)==""),]
mws.order   <- subset(mws.order, !is.na(meanW))

mws <- rbind(mws.species, mws.genus, mws.family, mws.order)
mws$taxon <- rownames(mws)
```

The total number of weight values thus obtained is 873, of which 394 are estimated at species level, 280 at genus level, 156 at family level, and and 43 at level or order.

FeedingTypes

Feedingtypes are known from a subset of the species. The following types are distinguished:

- "CaSc" = carnivore/scavenger
- "De" = depositfeeder
- "He" = herbivore
- "Om" = omnivore
- "Pa" = parasite
- "Su" = suspension feeder
- "SuDe" = suspension/deposit feeder

```
FeedingType <- read.csv("feedingtype_matched.txt")
cat("\ndimension and first part of the data set : \n")
```

```
##
## dimension and first part of the data set :
```

```
head(FeedingType[,1:2], n = 2)
```

```
##                   n.acc Trophy
## 1 Abludomelita obtusata     De
```

```
## 2             Abra alba   SuDe
```

```
table(FeedingType$Trophy)
```

```
##
## CaSc   De   He   Om   Pa   Su SuDe
##   78  100   20   56    3   79   37
```

Feeding types on higher taxonomic levels

We now assign feeding types to the genera and families, for which we take the most commonly encountered feedingtype at the lower level.

```
ft.species <- data.frame(FeedingType[,1:2], level = "species")
colnames(ft.species)[1:2] <- c("taxon", "ft")
ft.species$n <- 1

ft.genus <- data.frame(ft    = tapply(FeedingType$Trophy, INDEX = FeedingType$Genus,
                                      FUN = function(x) names(sort(table(x), decreasing = TRUE)[1])),
                       n     = tapply(FeedingType$Trophy, INDEX = FeedingType$Genus, FUN = length))
ft.genus  <- data.frame(taxon = rownames(ft.genus), ft.genus, level = "genus")

ft.family <- data.frame(ft = tapply(FeedingType$Trophy, INDEX = FeedingType$Family,
                                    FUN = function(x) names(sort(table(x), decreasing = TRUE)[1])),
                        n   = tapply(FeedingType$Trophy, INDEX = FeedingType$Family, FUN = length))
ft.family <- data.frame(taxon = rownames(ft.family), ft.family, level = "family")

ft.order <- data.frame(ft = tapply(FeedingType$Trophy, INDEX = FeedingType$Order,
                                   FUN = function(x) names(sort(table(x), decreasing = TRUE)[1])),
                       n   = tapply(FeedingType$Trophy, INDEX = FeedingType$Order, FUN = length))
ft.order <- data.frame(taxon = rownames(ft.order), ft.order, level = "order")


ft <- rbind(ft.species, ft.genus, ft.family, ft.order)
head(ft, n = 2)
```

```
##                   taxon   ft   level n
## 1 Abludomelita obtusata   De species 1
## 2             Abra alba SuDe species 1
```

The total number of feeding types thus obtained is 857, of which 373 are known at species level, 270 at genus level, 167 at family level and 47 at order level.

life history traits

Life history traits are also assigned on genus and family level, where we take the most commonly encountered trait at the lower level.

```
Traits.all <- read.csv("TraitsBiotur.csv")
cat("\ndimension and first part of the data set : \n")
```

```
##
## dimension and first part of the data set :
```

```
dim(Traits.all)
```

```
## [1] 273  21
```

```
head(Traits.all, n = 2)
```

```
##        phy        cla       ord           fam           gen
## 1 Annelida Polychaeta Sabellida    Serpulidae Spirobranchus
## 2 Annelida Polychaeta Sabellida Sabellariidae    Sabellaria
##                     taxon   Motility Body.size Burrowing.depth  Morphology
## 1 Spirobranchus triqueter Tubicolous     1-3cm             0cm Cylindrical
## 2    Sabellaria spinulosa Tubicolous    3-10cm             0cm Cylindrical
##   Mobility    Mixing.type Mixing.rate Mi Mi2 Ri Morphology.2
## 1 Very low Surface mixing    Very low  1   1  2            1
## 2 Very low Surface mixing    Very low  1   1  1            1
##   Morphology.2.1 Body.size.2 Burrowing.depth.2 Mixing.rate.2
## 1              1           2                 1             2
## 2              1           3                 1             3
```

```
traits.species <- data.frame(Traits.all[, c("taxon", "Motility", "Body.size", "Burrowing.depth", "Morpho
   n = 1, level = "species")
```

```
traitfun <- function(taxon = "gen", what = "Motility"){
  X <- data.frame(what = tapply(Traits.all[,what], INDEX = Traits.all[,taxon],
                       FUN = function(x) names(sort(table(x), decreasing = TRUE)[1])),
             n =  tapply(Traits.all[,what], INDEX = Traits.all[,taxon], FUN = length))
  colnames(X)[1] <- what
  X
}

traitfun.all <- function(taxon){
  T <- traitfun(taxon, what = "Motility")
  n <- T$n
  X <- T[,1]
  What <- c("Motility", "Body.size", "Burrowing.depth", "Morphology", "Mobility", "Mixing.type",
"Mixing.rate", "Mi", "Mi2", "Ri", "Morphology.2", "Morphology.2.1", "Body.size.2", "Burrowing.depth.2",

  for (w in What[-1])
    X <- data.frame(X,  traitfun(taxon, what = w)[,1])
  names(X) <- What
  X <- data.frame(taxon = rownames(T), X, n = n)
}

traits.genus  <- traitfun.all("gen")
traits.genus$level <- "genus"
traits.family <- traitfun.all("fam")
traits.family$level <- "family"
traits.order <- traitfun.all("ord")
traits.order$level <- "order"
traits <- rbind(traits.species, traits.genus [!traits.genus$taxon %in% traits.species$taxon,])
traits <- rbind(traits,         traits.family[!traits.family$taxon %in% traits$taxon,])
traits <- rbind(traits,         traits.order [!traits.order$taxon %in% traits$taxon,])
```

The total number of traits thus obtained is 523, of which 273 are estimated at species level, 131 at genus level, 89 at family level and 30 at order level.

We tabulate the number of occurrences of each trait in the resulting data.frame.

```
## motility:
```

```
##
##        Attached         Crawler Crawler-Swimmer      Tubicolous
##              15             288             132              88
```

```
## Body size:
```

```
##
##    <1cm   >20cm   1-3cm 10-20cm  3-10cm
##     101      36     106      55     225
```

```
## Burrowing depth:
```

```
##
##   >30cm   0-5cm     0cm 10-15cm 15-30cm  5-10cm
##      28     216      78      61      83      57
```

```
## Morphology:
```

```
##
## Articulated    Bivalved Cylindrical        Flat    Globular     Stellar
##         118         108         232           1          38          26
```

```
## Mobility:
```

```
##
##         High Intermediate          Low     Very low
##           40          246          153           84
```

```
## Mixing type:
```

```
##
##      D/U conveying          Diffusion Downward conveying
##                 34                123                 23
##       Regeneration     Surface mixing   Upward conveying
##                 10                301                 32
```

```
## Mixing rate:
```

```
##
##         High Intermediate          Low    Very high     Very low
##           34           62          143           44          240
```

```
## Mi:
```

```
##
##   1   2   3   4
##  84 153 246  40
```

```
## Ri:
```

```
##
##   1   2   3   4   5
##  28 275  86 124  10
```

The density data

Density data are read.

```
density <- read.csv("df_ab.csv")

cat("\ndimension and first part of the data set : \n")
```

```
##
## dimension and first part of the data set :
```

```
dim(density)
```

```
## [1] 1128549       6
```

```
head(density, n = 2)
```

```
##     data     sta        x        y          tx     dens
## 1 HELCOM HELCOM1 8.267167 56.72267   Abra alba 139.8601
## 2 HELCOM HELCOM1 8.267167 56.72267 Abra nitida 139.8601
```

```
cat("\ntotal number of data points per provider : \n")
```

```
##
## total number of data points per provider :
```

```
table(density$data)
```

```
##
##   HELCOM MACROBEL  MAREANO     MWTL     NSBS     ODAM      PMP     PORT
##    10399    12701    21591     8475    14622    94544      333     6796
##   REBENT     RSMP    SHARK     SMHI
##    24516   875244    56268     3060
```

Some data providers did not record density, but just presence/absence; these data are removed.

```
density      <- density[!is.na(density$dens), ]
density$data <- droplevels(density$data)

density$ID <- paste(density$sta, density$tx, sep = "")
```

The total number of species in this data set is 5433

A look at the data

Species densities are summed per station to give the total densities

```
TotalDensity <- aggregate(density$dens, by = list(data = density$data, sta = density$sta, x = density$x
names(TotalDensity)[5] <- "TotalDens"
cat("total number of stations per provider : \n")
```

```
## total number of stations per provider :
```

```
table(TotalDensity$data)
```

```
##
##   HELCOM MACROBEL  MAREANO     MWTL     NSBS     ODAM      PMP   REBENT
##      914      768      370      103      235     3057       23      216
##     RSMP    SHARK     SMHI
##    23915     6925      587
```

The total number of stations in this data set is 37113

The positions of the stations are plotted, colored according to the data provider

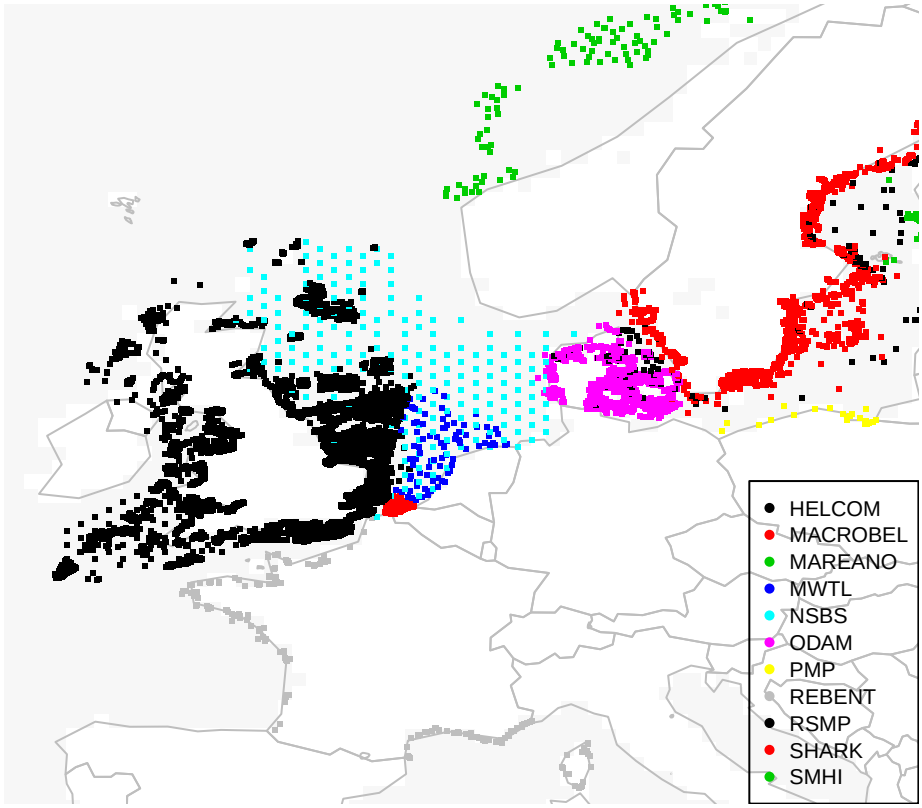
```
require(rworldmap)
```

```
## Loading required package: rworldmap
```

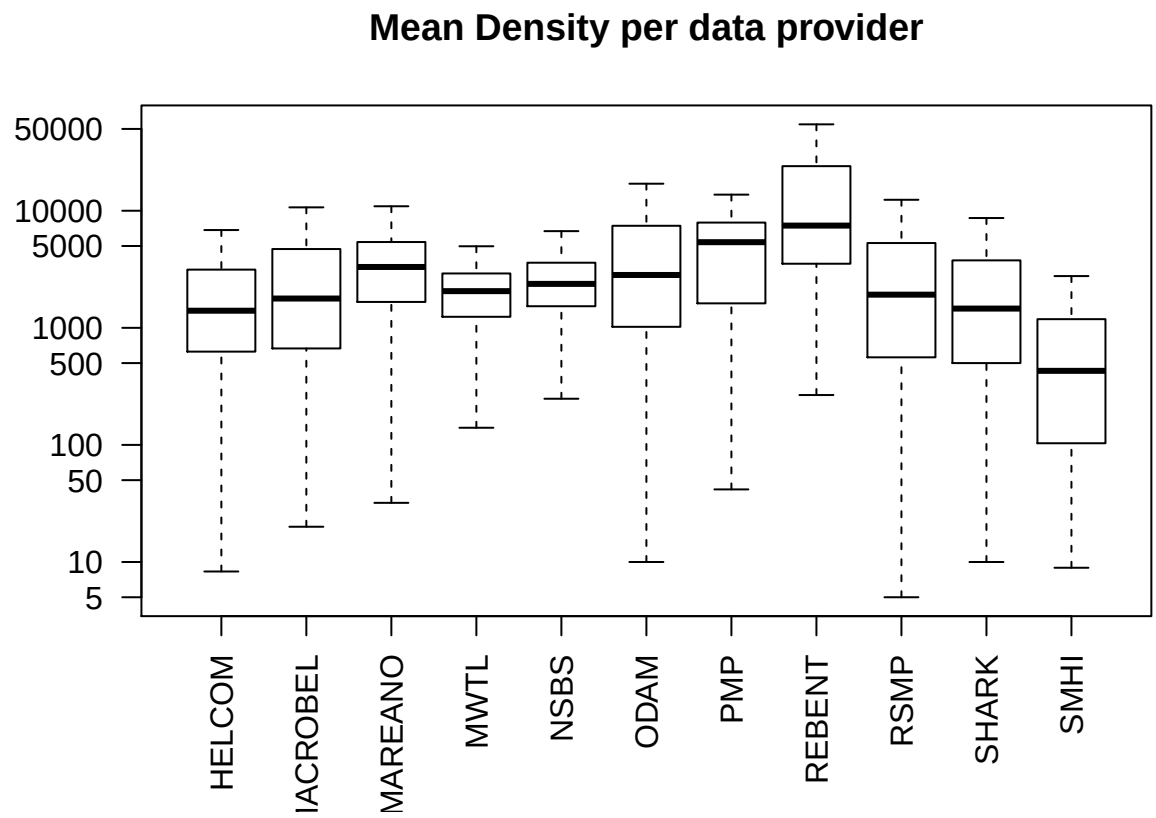
```
## Loading required package: sp
## ### Welcome to rworldmap ###
## For a short introduction type :   vignette('rworldmap')
```

```
mapGriddedData(colourPalette = rep("white", 5), oceanCol = grey(0.97), addLegend = FALSE,
               xlim = c(-10,20), ylim = c(45,65))
title( "data providers")
with(TotalDensity, points(x, y, pch = ".", cex = 3,  col = data))

legend("bottomright", pch = 16, cex = 0.7, col = 1:20, legend = levels(TotalDensity$data))
```




```
with (TotalDensity, boxplot((TotalDens)~data, outline = FALSE, las = 2, main = "Mean Density per data p
```



Adding individual weights and traits to density data

Weights and trait information is not available for all species, even after determining this information on genus and familiy level. We create data sets that remove the species that do not have all information. We then calculate total density on this reduced data set and see what fraction of total density we obtain.

First the required taxonomic information is added to the density list:

```
which(!density$tx %in% Taxo$txa)
```

```
## integer(0)
```

```
TaxoSpec <- Taxo$txa %in% density$tx
TaxoUsed <- Taxo[Taxo$txa%in%density$tx ,]
density.full <- merge(density, TaxoUsed, by.x = "tx", by.y = "tx")
```

To merge density data with the other data sets, a function is created.

```
MergeData <- function(density = density.full, data2){

  DataWithBiomass    <-       merge(density,            data2, by.x = "tx", by.y = "taxon")    # merging

  DataWithoutBiomass <- density[!density$ID %in% DataWithBiomass$ID, ]
  DataWithBiomass    <- rbind(DataWithBiomass,
                              merge(DataWithoutBiomass, data2, by.x = "gen", by.y = "taxon"))
```

```
  DataWithoutBiomass <- density[!density$ID %in% DataWithBiomass$ID, ]
  DataWithBiomass    <- rbind(DataWithBiomass,
                              merge(DataWithoutBiomass, data2, by.x = "fam", by.y = "taxon"))

  DataWithoutBiomass <- density[!density$ID %in% DataWithBiomass$ID, ]

  DataWithBiomass    <- rbind(DataWithBiomass,
                              merge(DataWithoutBiomass, data2, by.x = "ord", by.y = "taxon"))

  DataWithoutBiomass <- density[!density$ID %in% DataWithBiomass$ID, ]

    totalDensity <- with (DataWithBiomass, aggregate(dens, by = list(data = data, sta = sta, x = x, y =
  colnames(totalDensity)[5] <- "EstDens"
  list(complete = DataWithBiomass, incomplete = DataWithoutBiomass, DENS = totalDensity)
}
```

Merging density and weights

```
DensityWeight <- MergeData(density = density.full, data2 = mws)
```

The fraction of data for which individual weight could be estimated = 0.8758442.

The type of organisms for which the information is lacking belong to the phyla:

```
TT <- table(DensityWeight$incomplete$phy)
sort(TT[TT>0], decreasing = TRUE)
```

```
##
##        Cnidaria        Mollusca        Chordata        Annelida
##           34134           26415           10237           10028
##   Echinodermata         Bryozoa      Arthropoda       Sipuncula
##            9953            9538            9340            7669
##        Porifera        Nemertea  Cephalorhyncha Platyhelminthes
##            5996            2787            2546            2437
##      Rhodophyta      Ciliophora      Entoprocta    Foraminifera
##            1568            1394            1347             483
##                   Hemichordata    Chaetognatha     Brachiopoda
##             347             182             136             134
##    Gastrotricha       Phoronida      Tardigrada     Chlorophyta
##             101              95              46              38
##    Nematomorpha      Ochrophyta Xenacoelomorpha    Tracheophyta
##              25              25              14               8
##        Nematoda  Acanthocephala         Myzozoa
##               6               4               1
```

Many of these phyla are small organisms (Tardigrada, Foraminifera,...), so the biomass that is not taken into account is probably limited.

Merging density and traits

```
DensityTrait <- MergeData(density = density.full, data2 = traits)
```

The fraction of data for which traits could be estimated = 0.8451735 The unclassified organisms belong to:

```
TT <- table(DensityTrait$incomplete$phy)
sort(TT[TT>0], decreasing = TRUE)
```

```
##
##         Bryozoa        Mollusca        Cnidaria      Arthropoda
##           66661           30504           25460           11490
##        Chordata       Sipuncula        Annelida   Echinodermata
##            8566            7000            6579            5512
##        Porifera  Cephalorhyncha      Ciliophora        Nemertea
##            3262            2505            1394             708
##    Foraminifera      Rhodophyta Platyhelminthes    Hemichordata
##             342             341             177             123
##     Brachiopoda    Chaetognatha     Chlorophyta      Ochrophyta
##             107              85              38              24
##        Nematoda    Tracheophyta         Myzozoa
##               5               2               1
```

Merging density and feeding types

```
DensityFT <- MergeData(density = density.full, data2 = ft)
```

The fraction of data for which feeding types could be estimated = 0.8743257 The unclassified organisms belong to:

```
TT <- table(DensityFT$incomplete$phy)
sort(TT[TT>0], decreasing = TRUE)
```

```
##
##         Bryozoa        Cnidaria        Mollusca        Chordata
##           66661           25460           11121            8566
##       Sipuncula        Annelida        Porifera      Arthropoda
##            7000            6579            3262            3086
##  Cephalorhyncha      Ciliophora   Echinodermata        Nemertea
##            2505            1394            1123             708
##    Foraminifera      Rhodophyta Platyhelminthes    Hemichordata
##             342             341             177             123
##     Brachiopoda    Chaetognatha     Chlorophyta      Ochrophyta
##             107              85              38              24
##        Nematoda    Tracheophyta         Myzozoa
##               5               2               1
```

Data for estimating bioturbation potential

```
DataAll <- merge(DensityWeight$complete [, c("tx", "data","sta","x","y", "dens","ID", "meanW")],
                 DensityTrait$complete[, c("ID","Motility", "Body.size", "Burrowing.depth", "Morphology
                                           "Mixing.rate", "Mi", "Mi2", "Ri", "Morphology.2", "Morphology.
                                          "Body.size.2", "Burrowing.depth.2","Mixing.rate.2")], by = "ID"

dim(DataAll)
```

```
## [1] 880265     23
```

```
dim(density.full)
```

```
## [1] 1103726      14
```

The fraction of data that has all information to estimate bioturbation potential = 0.7975394

Check on the representativeness of the reduced data set

For the mwtl data we now compare the measured biomass with the biomass that we estimate, based on the mean weights. Note that in the MWTLdata, the density is called "ind" (from number of individuals)

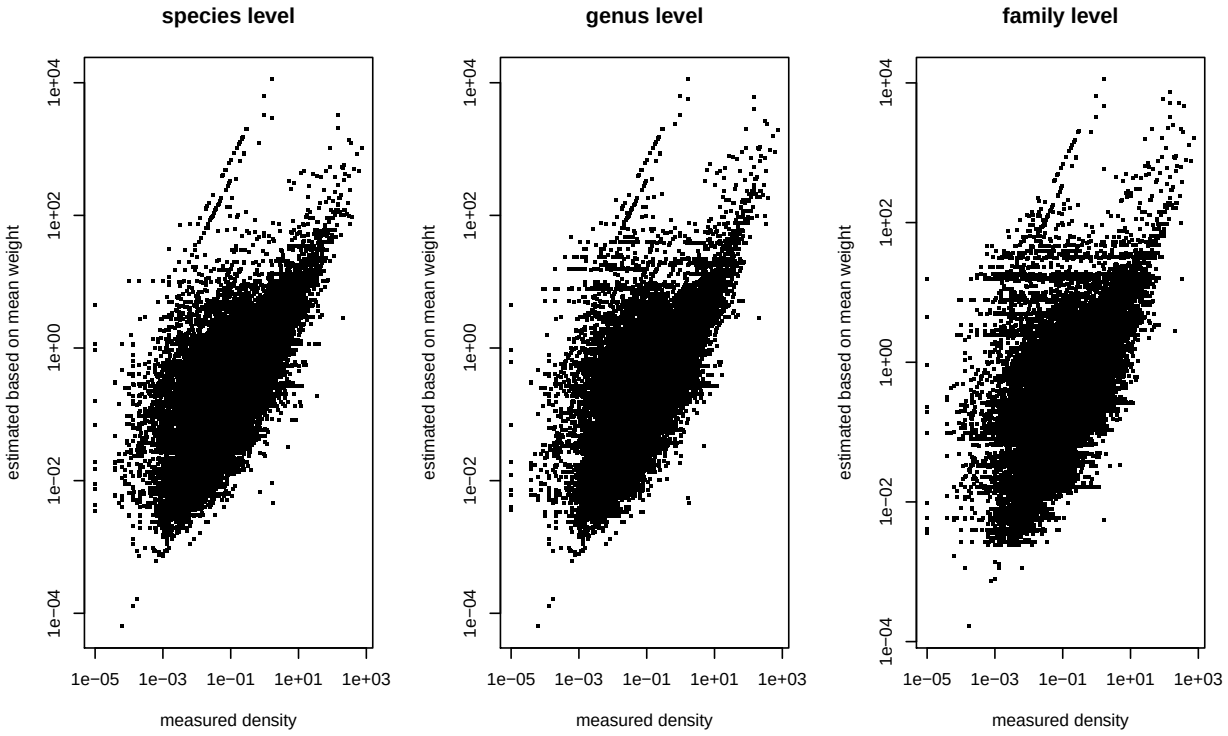
```
par(mfrow = c(1,3))
mws.species$tax <- rownames(mws.species)
MWTL.data <- subset(MWTLdata, ind > 0 & biom > 0)
#MWTL.data <- merge(MWTL.data,Taxo)
mwscompare     <- merge(mws.species, MWTL.data)
with(mwscompare, plot(biom, ind*meanW, log = "xy", pch = ".", cex = 3,
          xlab = "measured density", ylab = "estimated based on mean weight", main = "species level"))
with(mwscompare, summary(ind*meanW-biom))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  -274.205    -0.001     0.011     1.743     0.128 11264.787
```

```
mws.genus$gen <- rownames(mws.genus)
mwscompare     <- merge(mws.genus, MWTL.data)
with(mwscompare, plot(biom, ind*meanW, log = "xy", pch = ".", cex = 3,
          xlab = "measured density", ylab = "estimated based on mean weight", main = "genus level"))
with(mwscompare, summary(ind*meanW-biom))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  -264.275    -0.001     0.010     2.966     0.171 11264.787
```

```
mws.family$fam <- rownames(mws.family)
mwscompare     <- merge(mws.family, MWTL.data)
with(mwscompare, plot(biom, ind*meanW, log = "xy", pch = ".", cex = 3,
          xlab = "measured density", ylab = "estimated based on mean weight", main = "family level"))
```




```
with(mwscompare, summary(ind*meanW-biom))
```

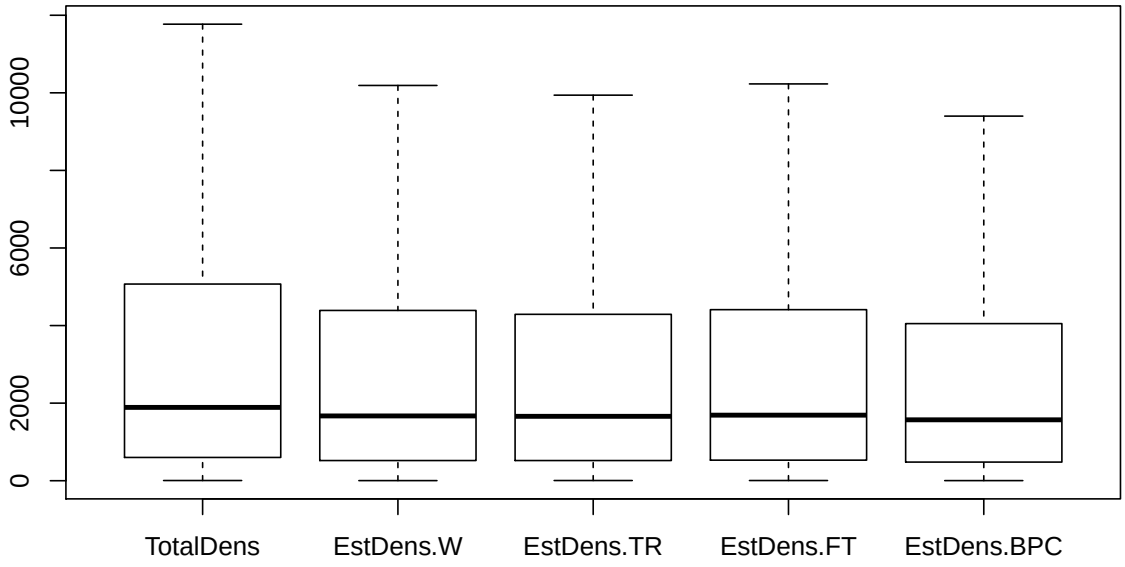
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
##  -323.832     -0.001      0.028      4.318      0.357 11264.787
```

We also check the error we would make in the density data if we would estimate them on the merged data sets.

```
DENS <- merge(TotalDensity, DensityWeight$DENS); colnames(DENS)[ncol(DENS)] <- "EstDens.W"
DENS <- merge(DENS,         DensityTrait$DENS); colnames(DENS)[ncol(DENS)] <- "EstDens.TR"
DENS <- merge(DENS,         DensityFT$DENS); colnames(DENS)[ncol(DENS)] <- "EstDens.FT"

TotDensAll <- with(DataAll, aggregate(dens, by = list(data = data, sta = sta, x = x, y = y), FUN = sum))
colnames(TotDensAll)[5] <- "EstDens.BPC"
DENS <- merge(DENS, TotDensAll)

boxplot(DENS[,-(1:4)], log = "", outline = FALSE)
```




```
cat("fraction of density based on species for which weight is known:\n")
```

```
## fraction of density based on species for which weight is known:
```

```
with(DENS, summary(EstDens.W/TotalDens))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001316 0.830189 0.929717 0.883898 0.996558 1.000000
```

```
cat("fraction of density based on species for which traits are known:\n")
```

```
## fraction of density based on species for which traits are known:
```

```
with(DENS, summary(EstDens.TR/TotalDens))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001121 0.813509 0.936709 0.878555 0.999525 1.000000
```

```
cat("fraction of density based on species for which feeding type is known:\n")
```

```
## fraction of density based on species for which feeding type is known:
```

```
with(DENS, summary(EstDens.FT/TotalDens))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001316 0.840909 0.957272 0.896053 1.000000 1.000000
```

```
cat("fraction of density based on species for which all is known:\n")
```

```
## fraction of density based on species for which all is known:
```

```
with(DENS, summary(EstDens.BPC/TotalDens))
```

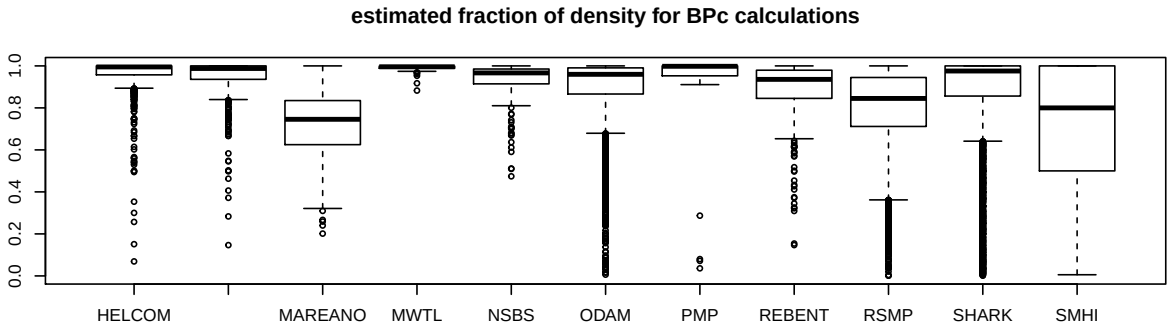
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

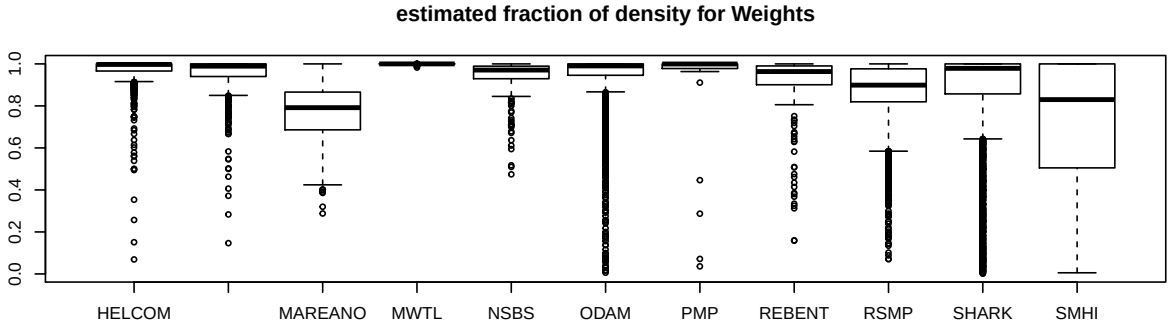
```
## 0.001121 0.741935 0.888889 0.832443 0.984615 1.000000
```

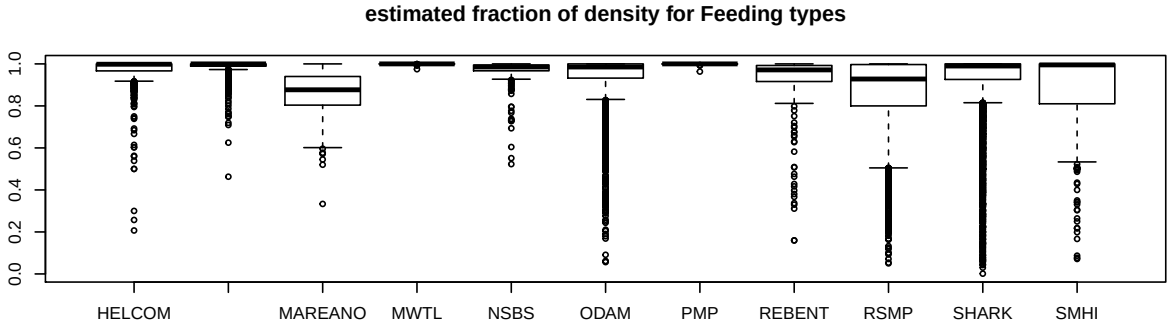
The estimated density if we only use species for which have all data required for BPc estimation is thus on average 0.8324433 of the true value.

As we want to estimate bioturbation potential, we now split this number for the different data providers to see for which data sets we make the largest errors.

```
par(mfrow = c(3,1))
DENS$r.bpc <- DENS$EstDens.BPC/DENS$TotalDens
DENS$r.w   <- DENS$EstDens.W/DENS$TotalDens
DENS$r.ft  <- DENS$EstDens.FT/DENS$TotalDens
with(DENS, boxplot(r.bpc ~ data, main = "estimated fraction of density for BPc calculations"))
with(DENS, boxplot(r.w ~ data, main = "estimated fraction of density for Weights"))
with(DENS, boxplot(r.ft ~ data, main = "estimated fraction of density for Feeding types"))
```







The mean recovered fraction of density for the various data providers is :

```
D <- data.frame(BPC = with(DENS, tapply(r.bpc, INDEX =  data, FUN = mean)),
           W = with(DENS, tapply(r.w, INDEX =  data, FUN = mean)),
           FT = with(DENS, tapply(r.ft, INDEX =  data, FUN = mean)))
knitr:::kable(D, digits = 1)
```

	BPC	W	FT
HELCOM	0.9609198	0.9672662	0.9702654
MACROBEL	0.9500470	0.9520774	0.9826944
MAREANO	0.7202315	0.7688202	0.8585159
MWTL	0.9907961	0.9992683	0.9992434
NSBS	0.9297956	0.9352938	0.9611760
ODAM	0.8868025	0.9363271	0.9308547
PMP	0.8376353	0.8564493	0.9974889
REBENT	0.8687459	0.9011789	0.9079913
RSMP	0.8114143	0.8821951	0.8777615
SHARK	0.8589040	0.8606155	0.9233342
SMHI	0.7239096	0.7388481	0.8860511

Estimating BPc, the bioturbation potential

The bioturbation potential is now estimated on the reduced data set. First the contribution of each species to BPc is estimated, based on the indivual weight, the abundancy, and their mobility and reworking mode.

```
BPc <- function(weight, abundance, mobility, rework)
  sqrt(weight) * abundance * mobility * rework

DataAll$BPC <- BPc(DataAll$meanW, DataAll$dens, as.numeric(DataAll$Mi), as.numeric(DataAll$Ri))
DataAll$Biomass <-DataAll$meanW * DataAll$dens
```

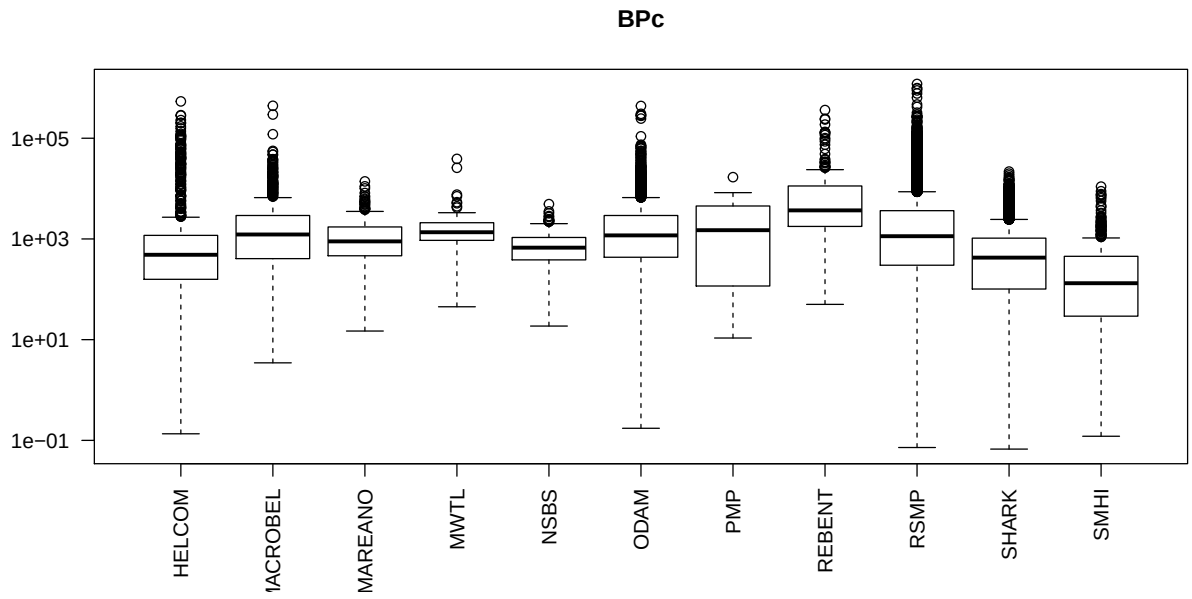
Then the BPcs of all species per station are added.

```
TotalBPC <- with (DataAll, aggregate(BPC, by=list(data = data, sta = sta, x = x, y = y), FUN=sum, na.rm

names(TotalBPC)[5] <- "BPC"
```

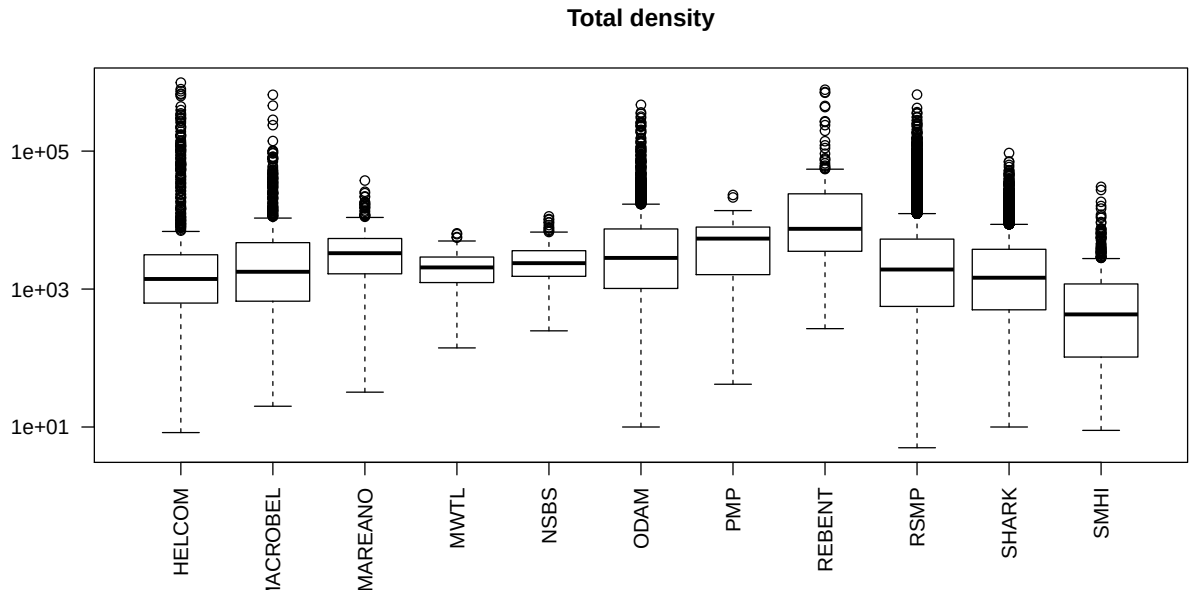
Characteristics per data provider

```
with(TotalBPC, boxplot(BPC~data, main = "BPc", log = "y", las = 2))
```



TotalDensity was already calculated

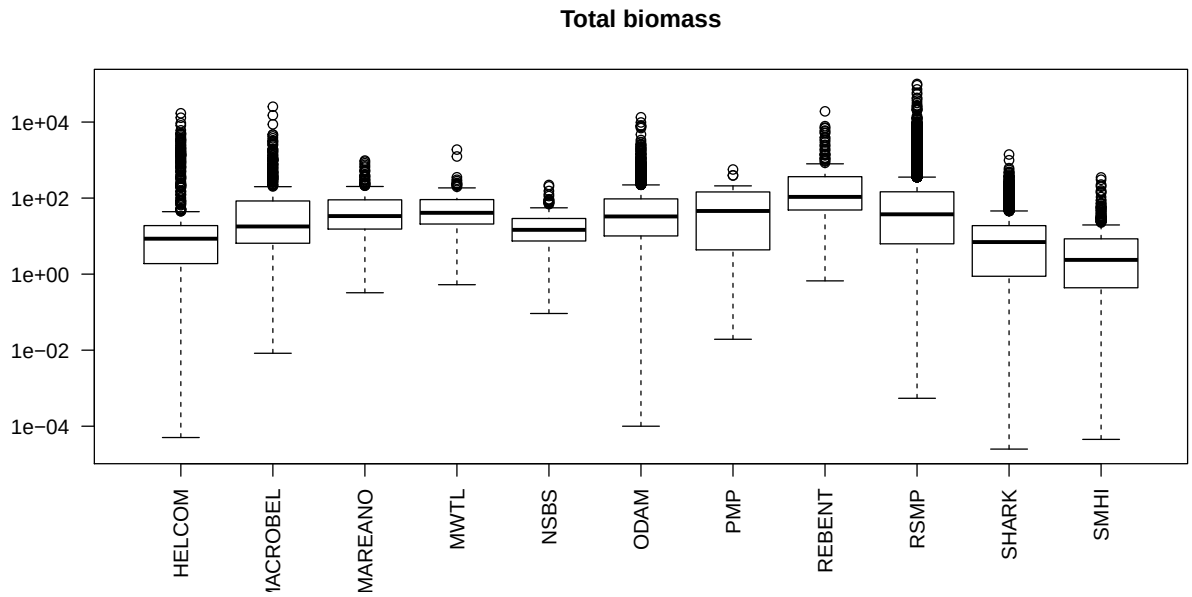
```
with(TotalDensity, boxplot(TotalDens~data, main = "Total density", log = "y", las = 2))
```



The total biomass per station:

```
TotalBiomass <- with(DensityWeight$complete, aggregate(dens*meanW, by=list(data = data, sta = sta, x = x
colnames(TotalBiomass)[5] <- "AFDW"
```

```
with(TotalBiomass, boxplot(AFDW~data, main = "Total biomass", log = "y", las = 2))
```

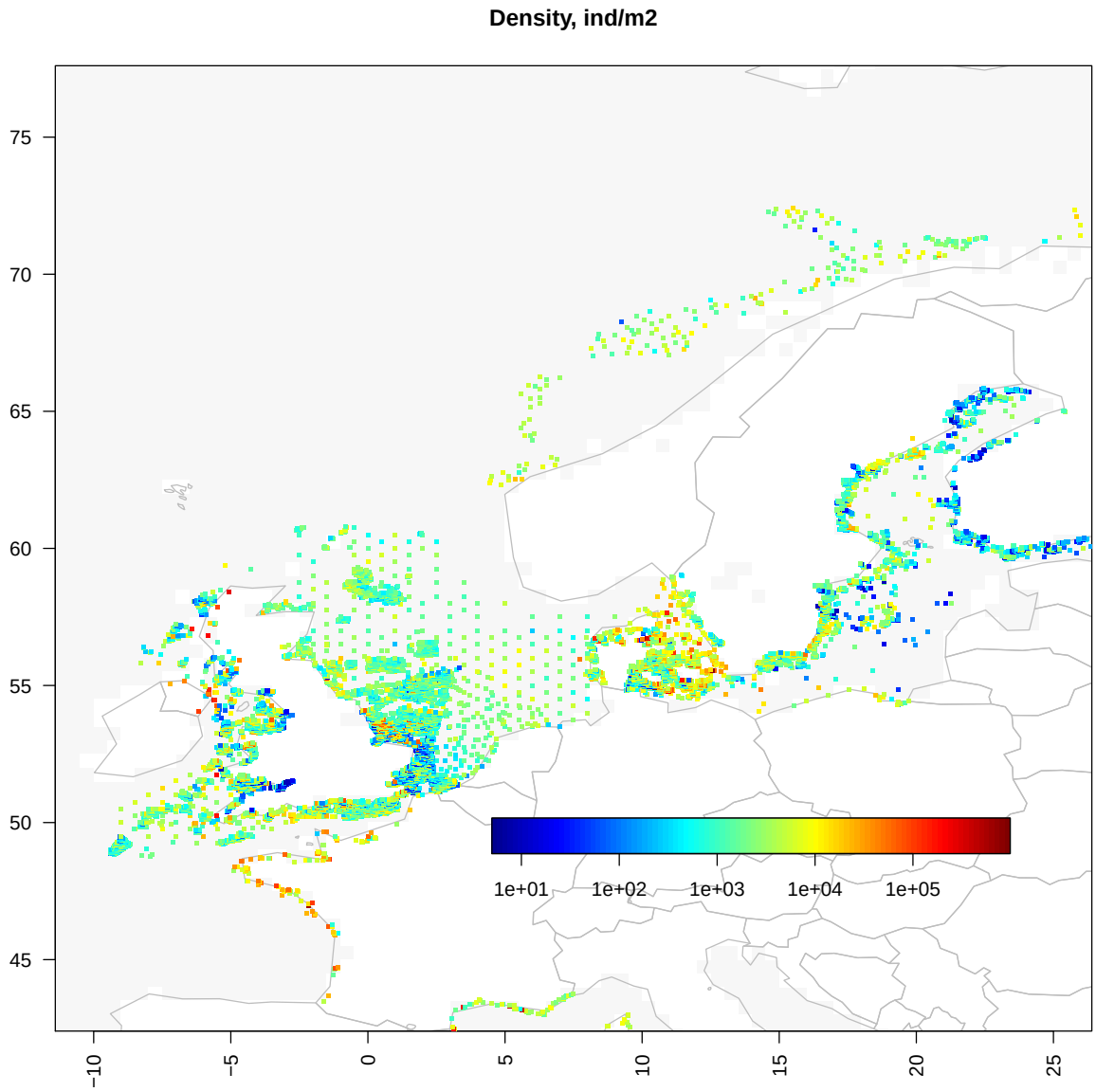


A function is created to generate image plots

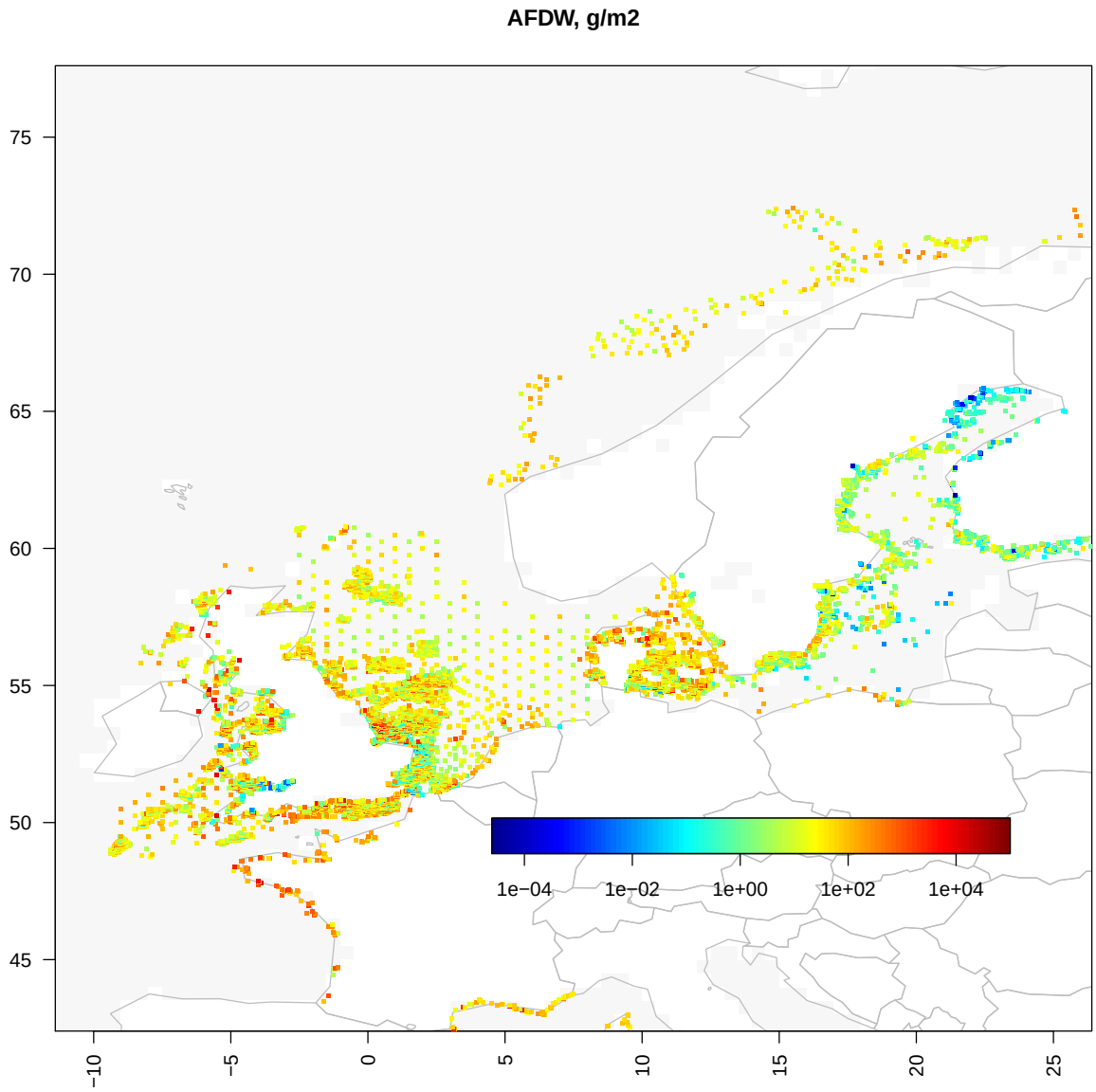
```
par(mfrow = c(2,2), oma = c(0,0,0,2))
ImagePlot <- function(x, y, colvar, main = "", ...){
  par(las = 2)
  require(rworldmap)
  mapGriddedData(colourPalette = rep("white", 5), oceanCol = grey(0.97), addLegend = FALSE,
                 xlim = c(-10,25), ylim = c(45,75))
  axis(side = 1); axis (side = 2)
  box()
  title(main)
  points2D(x, y, colvar = colvar, pch = ".", cex = 4, las = 1, add = TRUE,
           colkey = list(side = 1, dist = -0.15, length = 0.5, shift = 0.15), ...)

}
```

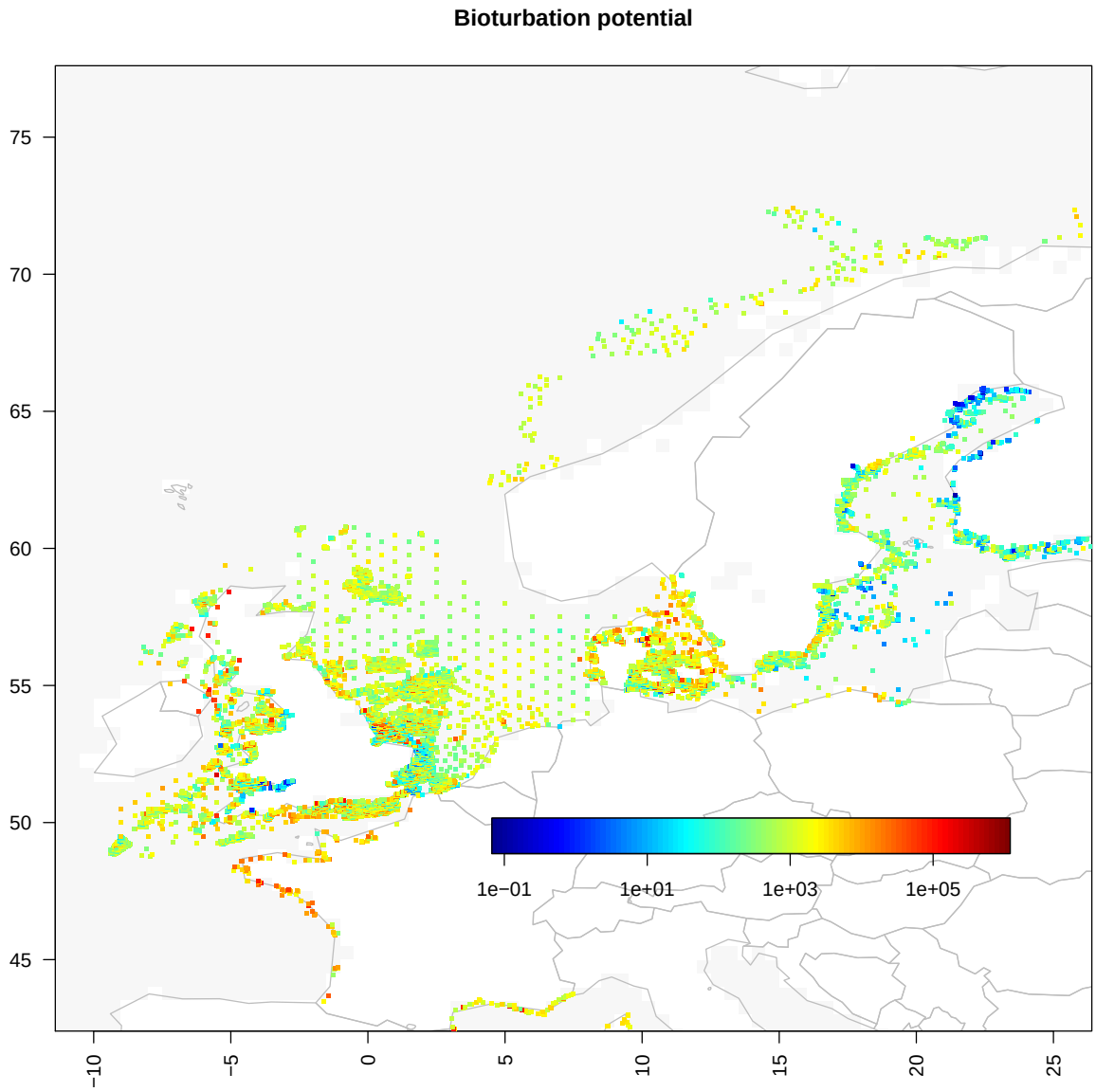
```
ImagePlot(x = TotalDensity$x, y = TotalDensity$y, colvar = TotalDensity$TotalDens,
          main = "Density, ind/m2", log = "c")
```




```
ImagePlot(x = TotalBiomass$x, y = TotalBiomass$y, colvar = TotalBiomass$AFDW,
          main = "AFDW, g/m2", log = "c")
```




```
ImagePlot(x = TotalBPC$x, y = TotalBPC$y, colvar = TotalBPC$BPC,
          main = "Bioturbation potential", log = "c")
```



Biomass per feeding type

```
Weight_FT <- merge(DensityFT$complete[,c(1:7,15)], DensityWeight$complete[,c(7,15)], by = "ID")
Weight_FT$Biomass <- Weight_FT$dens * Weight_FT$meanW

FT_Stats <- with(Weight_FT, tapply(Biomass, INDEX = list(sta, ft), FUN = sum, na.rm = TRUE))
FT_Stats <- data.frame(sta = rownames(FT_Stats), FT_Stats,
                       AFDW_FT = rowSums(FT_Stats, na.rm = TRUE))
FT_Stats <- merge(TotalBiomass,FT_Stats, by = "sta")
head(FT_Stats)
```

```
##         sta   data         x        y       AFDW      CaSc         De
## 1   HELCOM1 HELCOM  8.267167 56.72267 1448.25300 17.149599  42.126847
```

```
## 2  HELCOM10 HELCOM  9.238333 56.65717 2397.50931  1.317834 222.075764
## 3 HELCOM100 HELCOM 11.485833 56.20500  103.42123 16.583690  19.015784
## 4 HELCOM101 HELCOM 11.500000 57.83333   16.53744  4.145739   1.867202
## 5 HELCOM102 HELCOM 11.516667 57.75000   14.20329  2.931471   2.878464
## 6 HELCOM103 HELCOM 11.525000 57.54998  149.55492 57.640406  23.736603
##           He           Om Pa          Su      SuDe    AFDW_FT
## 1 0.07849892 514.74474652 NA  788.696816 81.543487 1444.34000
## 2         NA 123.90006780 NA 2045.790024  4.425617 2397.50931
## 3 0.09126297           NA NA   24.847185 32.864295   93.40222
## 4 0.05977934   0.06218908 NA    4.780328  5.622207   16.53744
## 5 0.08151728   0.09328362 NA    4.373339  3.845220   14.20329
## 6 0.14673111   2.01267472 NA   34.686426 31.332081  149.55492
```

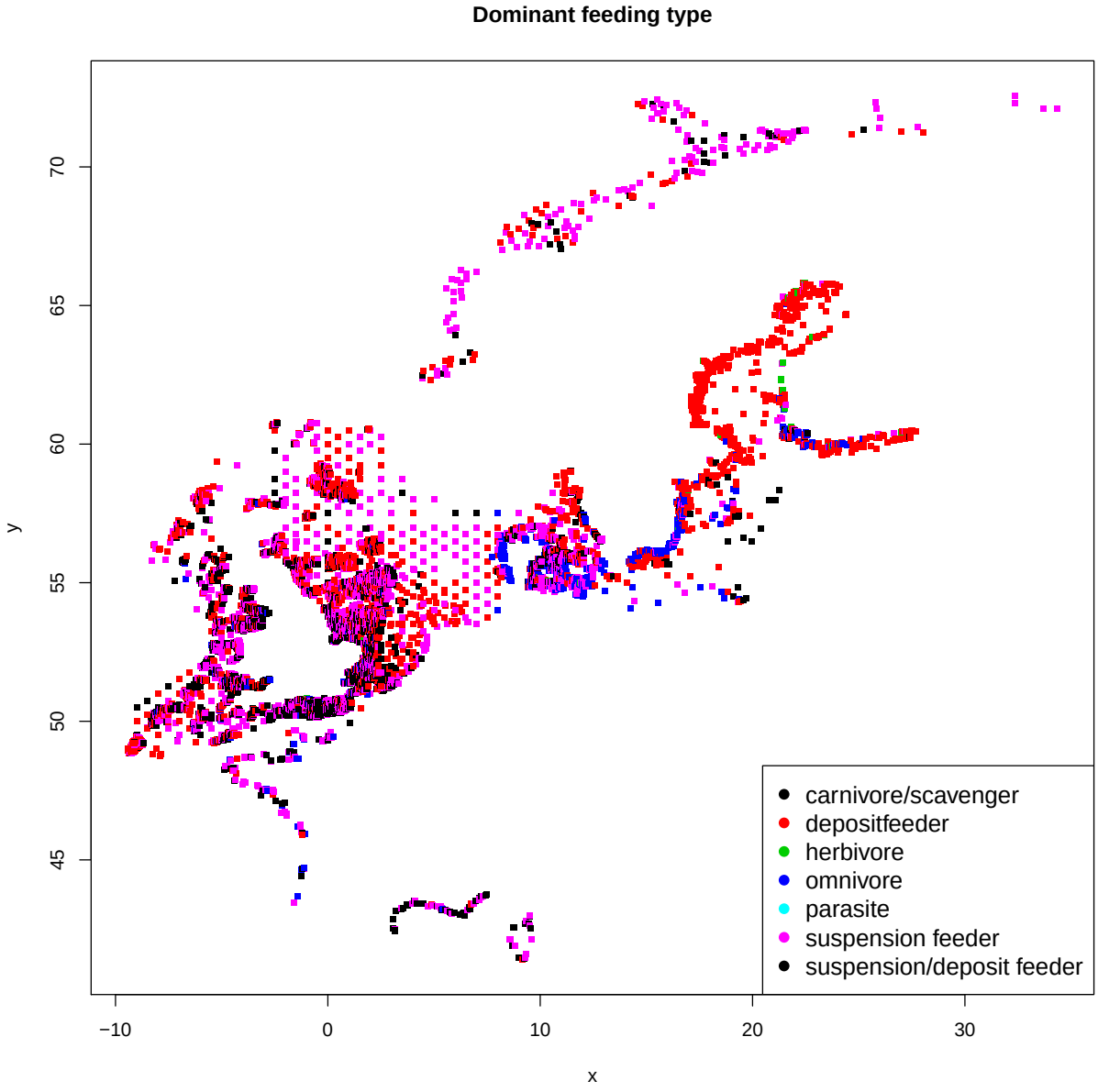
The stations where none of the feeding types are known are removed, and the most dominant feeding type selected.

```
ISNAS <- unlist(apply(FT_Stats[,6:12], FUN = function(x) sum(is.na(x)), MARGIN = 1))
table(ISNAS)
```

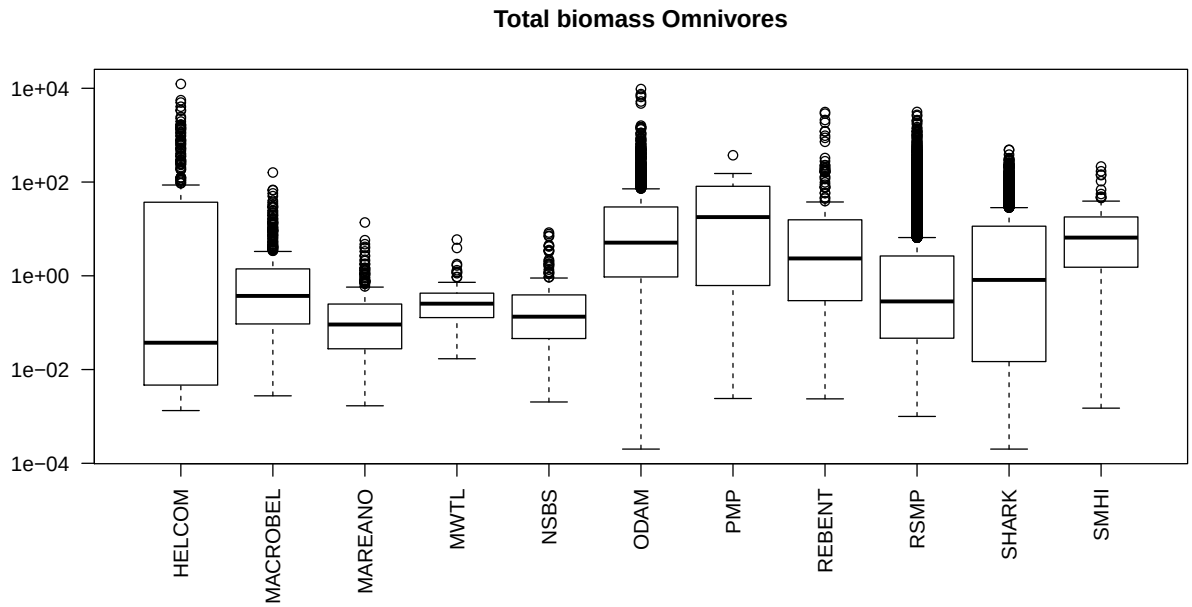
```
## ISNAS
##     0     1     2     3     4     5     6     7
##  1077  5117 13905  6480  4270  3601  2360    48
```

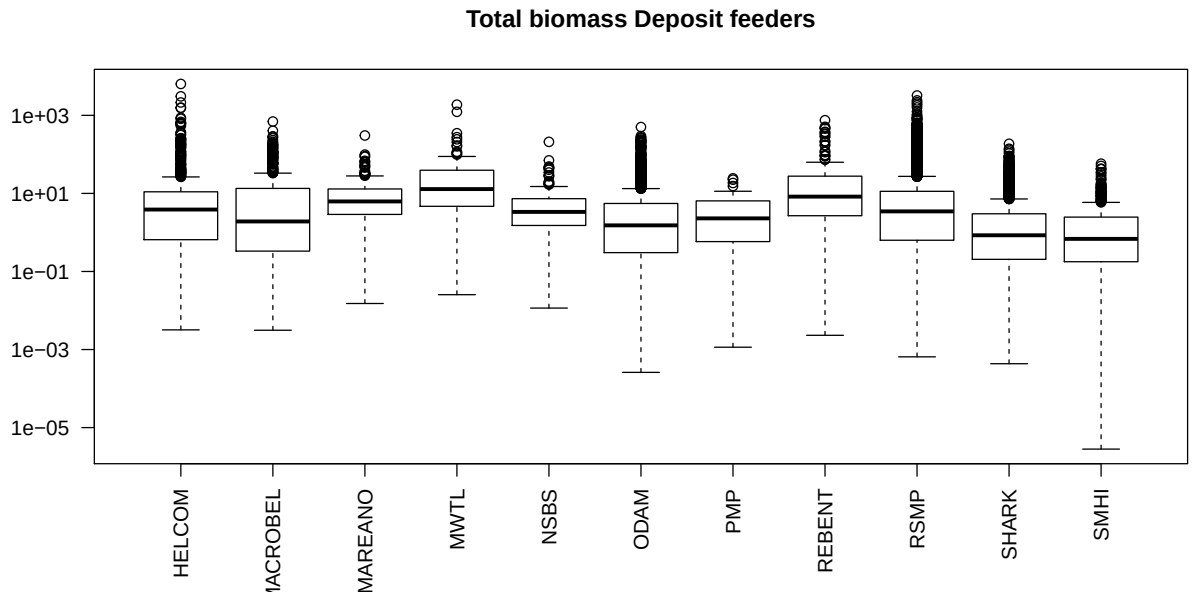
```
FT_Stats <- FT_Stats[ISNAS < 7,]
NAMES <- colnames(FT_Stats)[6:12]
FT_Stats$dominant <- unlist(apply(FT_Stats[,6:12], FUN = function(x) NAMES[which.max(x)], MARGIN = 1))
FT_Stats$dominant <- as.factor(FT_Stats$dominant)
```

```
FTnames <- as.data.frame(matrix(ncol = 2, byrow = TRUE, data = c(
 "CaSc" , "carnivore/scavenger",
 "De" , "depositfeeder",
 "He" , "herbivore",
 "Om" , "omnivore",
 "Pa" , "parasite",
 "Su" , "suspension feeder",
 "SuDe" , "suspension/deposit feeder"
)))
with (FT_Stats, plot(x, y, col = c(1:6)[dominant], pch = ".", cex = 5, main = "Dominant feeding type"))
legend("bottomright", col = 1:6, legend = FTnames[,2], pch = 16, cex = 1.25)
```

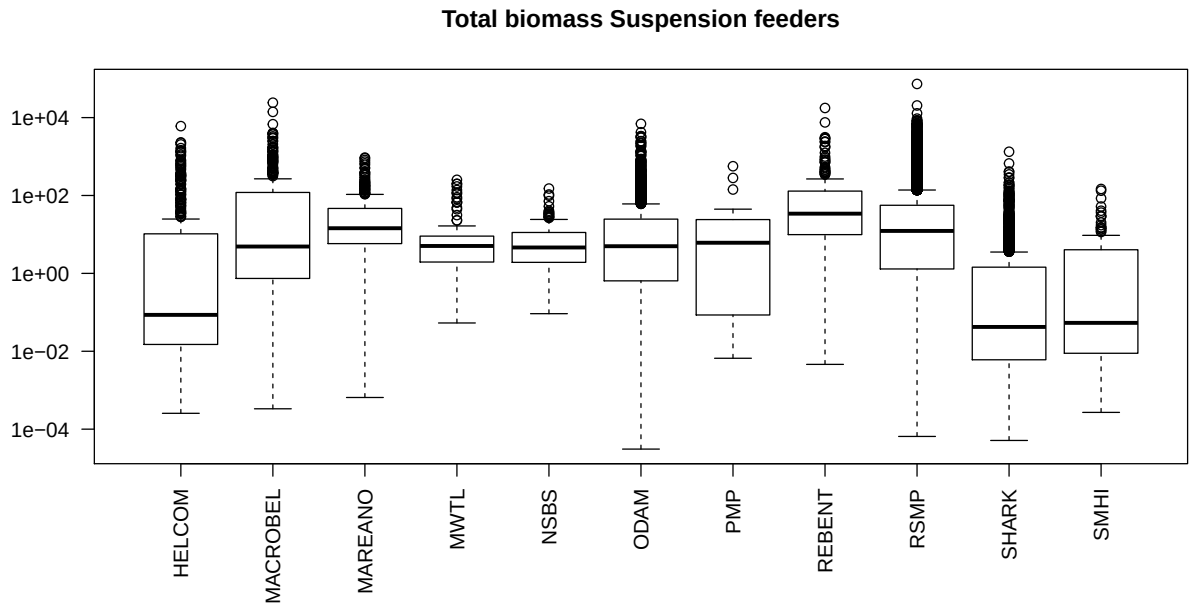


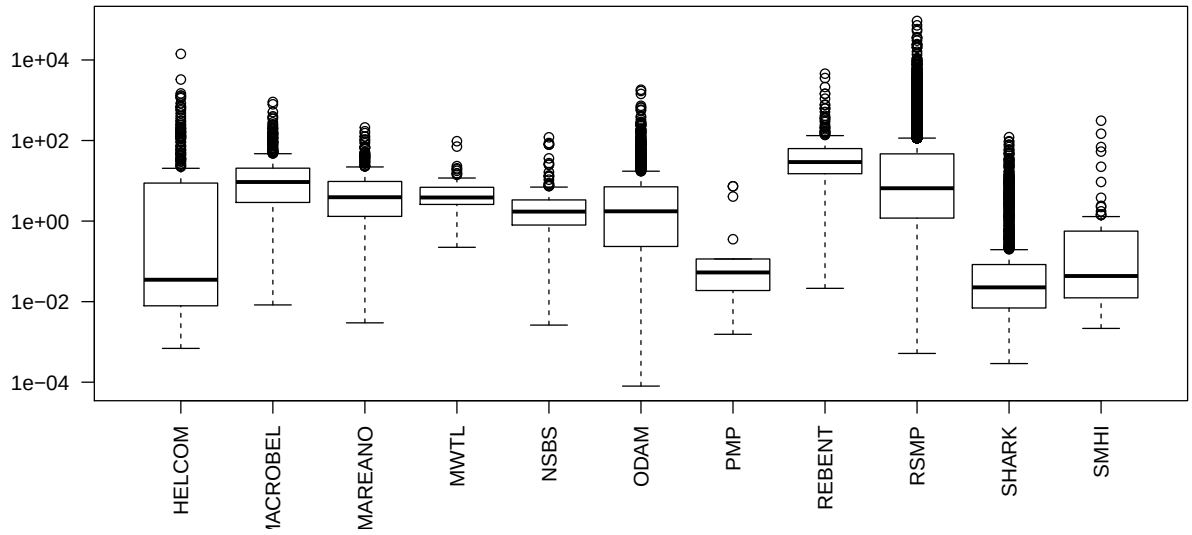

```
par(mfrow = c(2,1))
with(FT_Stats, boxplot(Om~data, main = "Total biomass Omnivores", log = "y", las = 2))
with(FT_Stats, boxplot(De~data, main = "Total biomass Deposit feeders", log = "y", las = 2))
```





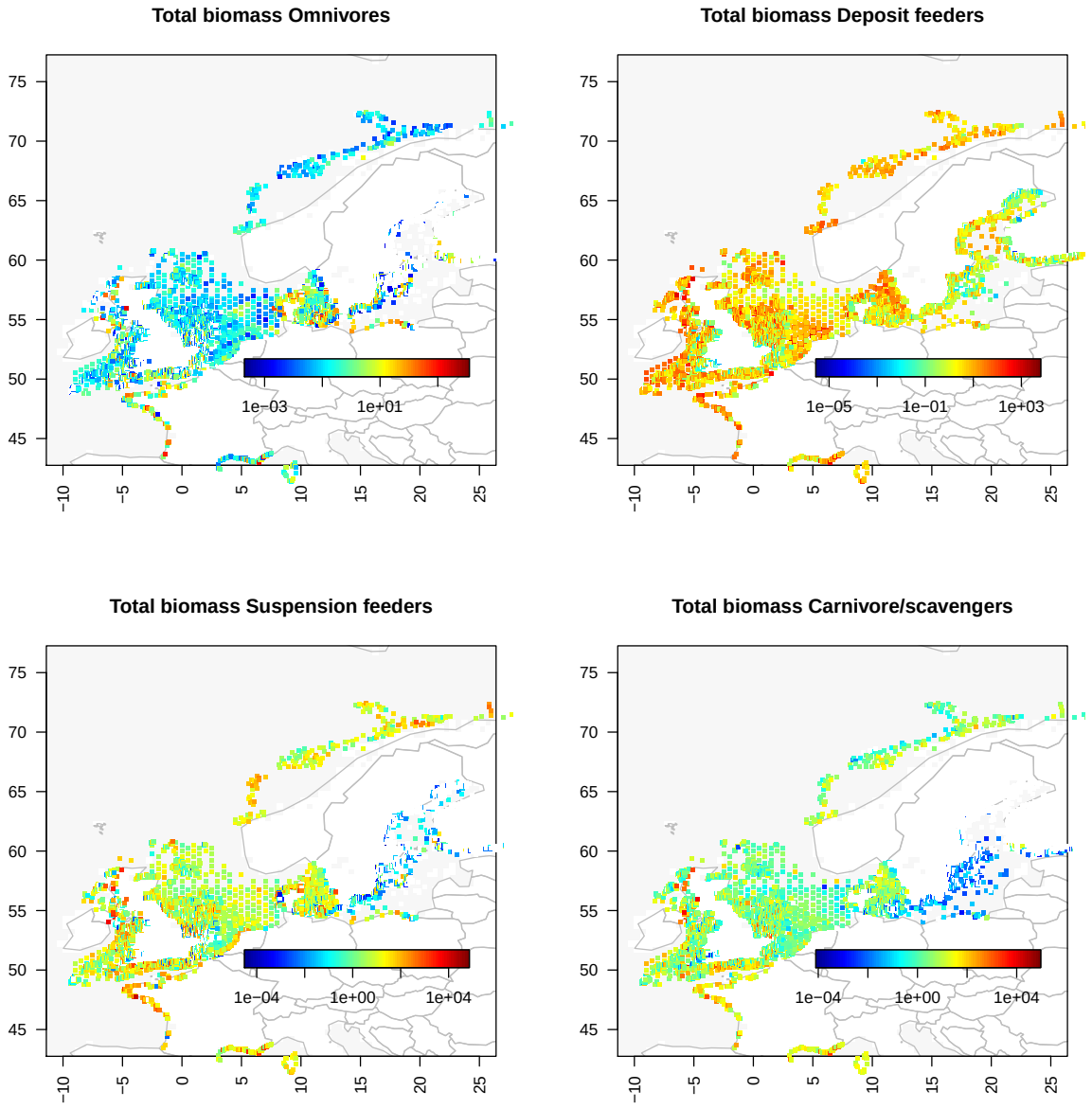

```
par(mfrow = c(2,1))
with(FT_Stats, boxplot(Su~data, main = "Total biomass Suspension feeders", log = "y", las = 2))
with(FT_Stats, boxplot(CaSc~data, main = "Total biomass Carnivore/scavengers", log = "y", las = 2))
```





The total biomasses of feeding types

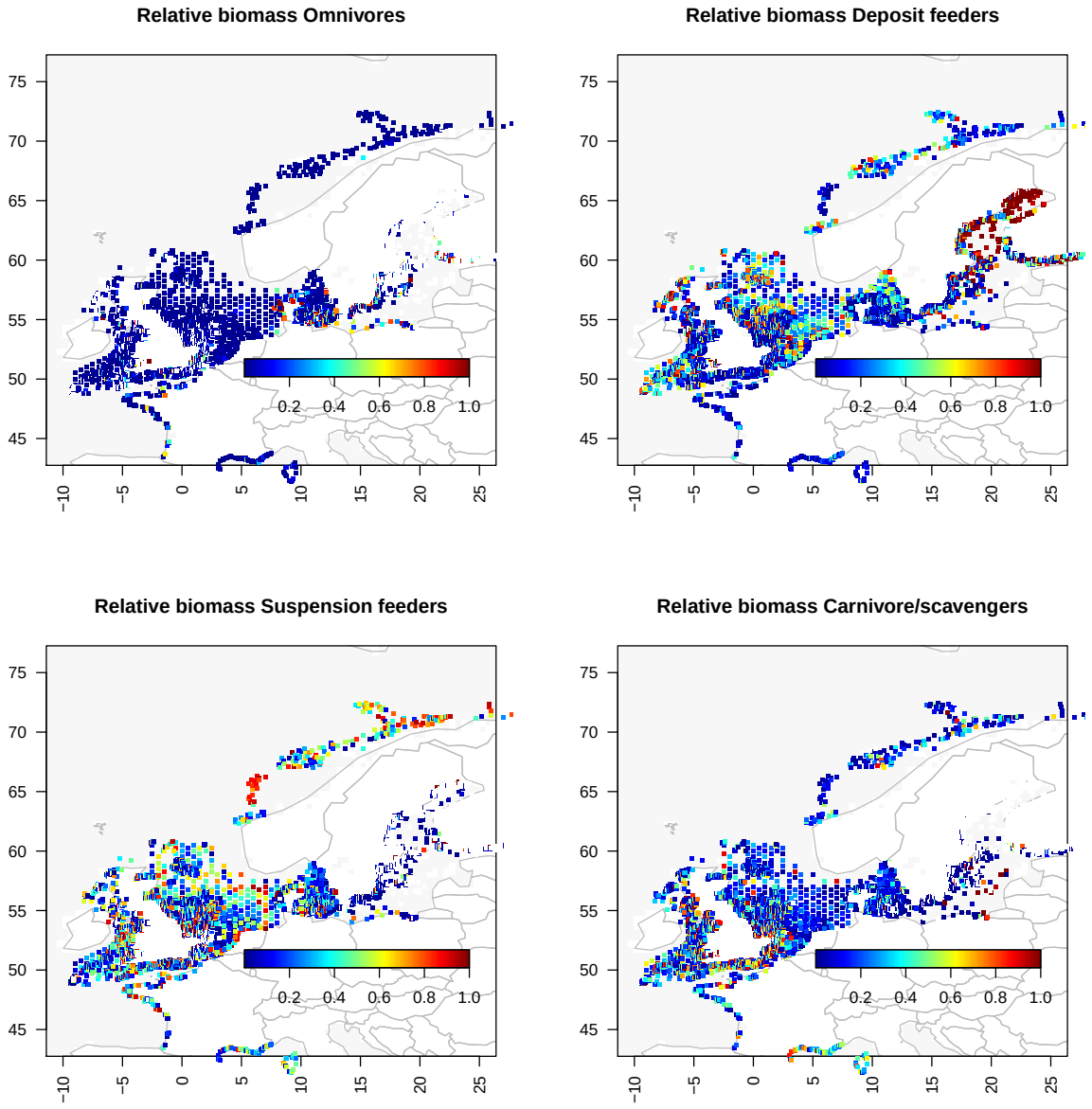
```
par(mfrow = c(2,2), oma = c(0,0,0,2))
ImagePlot(x = FT_Stats$x, y = FT_Stats$y, colvar = FT_Stats$Om,
          main = "Total biomass Omnivores", log = "c")
ImagePlot(x = FT_Stats$x, y = FT_Stats$y, colvar = FT_Stats$De,
          main = "Total biomass Deposit feeders", log = "c")
ImagePlot(x = FT_Stats$x, y = FT_Stats$y, colvar = FT_Stats$Su,
          main = "Total biomass Suspension feeders", log = "c")
ImagePlot(x = FT_Stats$x, y = FT_Stats$y, colvar = FT_Stats$CaSc,
          main = "Total biomass Carnivore/scavengers", log = "c")
```



The relative biomasses of feeding types

```
par(mfrow = c(2,2), oma = c(0,0,0,2))
FT_rel <- FT_Stats
FT_rel[,6:12] <- FT_rel[,6:12]/FT_rel$AFDW_FT

ImagePlot(x = FT_rel$x, y = FT_rel$y, colvar = FT_rel$Om,
          main = "Relative biomass Omnivores", log = "")
ImagePlot(x = FT_rel$x, y = FT_rel$y, colvar = FT_rel$De,
          main = "Relative biomass Deposit feeders", log = "")
ImagePlot(x = FT_rel$x, y = FT_rel$y, colvar = FT_rel$Su,
          main = "Relative biomass Suspension feeders", log = "")
ImagePlot(x = FT_rel$x, y = FT_rel$y, colvar = FT_rel$CaSc,
          main = "Relative biomass Carnivore/scavengers", log = "")
```



Writing the results

```
write.csv(file = "results/Density.csv", TotalDensity)
write.csv(file = "results/Biomass.csv", TotalBiomass)
write.csv(file = "results/BPC.csv", TotalBPC)
SuspensionFeeders <- FT_Stats[,c(2,1,3,4,11)]; names(SuspensionFeeders)[5] <- "AFDW_S"
DepositFeeders <- FT_Stats[,c(2,1,3,4,7)]; names(DepositFeeders)[5] <- "AFDW_D"
write.csv(file = "results/SuspensionFeeders.csv", SuspensionFeeders)
write.csv(file = "results/DepositFeeders.csv", DepositFeeders)
```