How to Mizer - how to parametrise a Mizer model

Julia L. Blanchard, Ken H. Andersen, Gustav Delius, Romain Forestier

2021-04-12

```
source("../utility.R")
```

Beginner level tutorial - how to parameterise a Mizer model

In this tutorial you will learn

- What type of data is needed to parametrise a Mizer model
- How to convert raw data into a `mizerParams` object
- Check the several assumptions you are making about your parameters
- Explore the model for the first time with your own data set

Step 1 - What data are typically needed

In this section you will:

- learn what data you need to run Mizer

First you are going to need a data frame of species specfic parameters to input into the `newMultispeciesParams()` function. This function requires at least three columns of parameters:

- species name (`species`)
- asymptotic size (`w_inf`)
- maximum intake rate (`h`) or von Bertalanffy growth parameter (`k_vb`)

The data frame is arranged as species by parameter, so each column of the parameter data frame is a parameter and each row has the values of the parameters for one of the species in the model.

You will also need an interaction matrix that defines the overlapping interaction between each species (default is set to 1, meaning full time overlap). Usually when species' interactions are set to 1, no species can coexist as preys are always eaten by predators.

All other parameters either have default values or will be calculated from the supplied parameters.

```
smallExample <- data.frame("species" = c("george","ginette"),"w_inf" = c(1.6,1.7), "k_vb" = c(.6,1))
smallExample
```

```
##   species w_inf k_vb
## 1  george   1.6  0.6
## 2 ginette   1.7  1.0
```

```
smallInter <- matrix(c(.2,.4,.1,.3),ncol = 2, dimnames = list(smallExample$species,smallExample$species
smallInter
```

```
##         george ginette
## george     0.2     0.1
## ginette    0.4     0.3
```

The set of species' specific parameters for the North Sea is availalble in the "How to Mizer" GitHub repository as shown below.

```
# loading North Sea data
nsParams <- read.csv("../data/nsparams.csv")[,-1]

# This data frame already has Rmax values, let's remove them to calibrate them again later
nsParams[,"r_max"] <- Inf
# ordering by asymptotic size for color gradient
nsParams <- nsParams[order(nsParams$w_inf),]
# need to order the interaction matrix as well | it's so ugly please help
inter <- select(as.data.frame(inter),nsParams$species)
inter <- as.matrix(select(as.data.frame(t(inter)),nsParams$species))

# If you want to make it less multi-species and more trait-based model
# nsParams[,"beta"] <-100
# nsParams[,"sigma"] <-1.5

nsParams
```

```
##    X1 species   w_inf w_mat   beta sigma r_max  k_vb       sel_func  l25  l50
## 1   1   Sprat    33.0    13  51076   0.8   Inf 0.681 sigmoid_length  7.6  8.1
## 2   2 Sandeel    36.0     4 398849   1.9   Inf 1.000 sigmoid_length  9.8 11.8
## 3   3  N.pout   100.0    23     22   1.5   Inf 0.849 sigmoid_length  8.7 12.2
## 5   5     Dab   324.0    21    191   1.9   Inf 0.536 sigmoid_length 11.5 17.0
## 4   4 Herring   334.0    99 280540   3.2   Inf 0.606 sigmoid_length 10.1 20.8
## 8   8 Gurnard   668.0    39    283   1.8   Inf 0.266 sigmoid_length 19.8 29.0
## 7   7    Sole   866.0    78    381   1.9   Inf 0.284 sigmoid_length 16.4 25.8
## 6   6 Whiting  1192.0    75     22   1.5   Inf 0.323 sigmoid_length 19.8 29.0
## 9   9  Plaice  2976.0   105    113   1.6   Inf 0.122 sigmoid_length 11.5 17.0
## 10 10 Haddock  4316.5   165    558   2.1   Inf 0.271 sigmoid_length 19.1 24.3
## 12 12  Saithe 39658.6  1076     40   1.1   Inf 0.175 sigmoid_length 35.3 43.6
## 11 11     Cod 39851.3  1606     66   1.3   Inf 0.216 sigmoid_length 13.2 22.9
##        a     b catchability         gear
## 1  0.007 3.014   1.29533333 sigmoid_gear
## 2  0.001 3.320   0.06510547 sigmoid_gear
## 3  0.009 2.941   0.31380000 sigmoid_gear
## 5  0.010 2.986   0.97800000 sigmoid_gear
## 4  0.002 3.429   0.18150000 sigmoid_gear
## 8  0.004 3.198   0.46250569 sigmoid_gear
## 7  0.008 3.019   0.37383333 sigmoid_gear
## 6  0.006 3.080   0.24266667 sigmoid_gear
## 9  0.007 3.101   0.18483333 sigmoid_gear
## 10 0.005 3.160   0.30150000 sigmoid_gear
## 12 0.007 3.075   0.39300000 sigmoid_gear
## 11 0.005 3.173   0.26666749 sigmoid_gear
```

```
inter
```

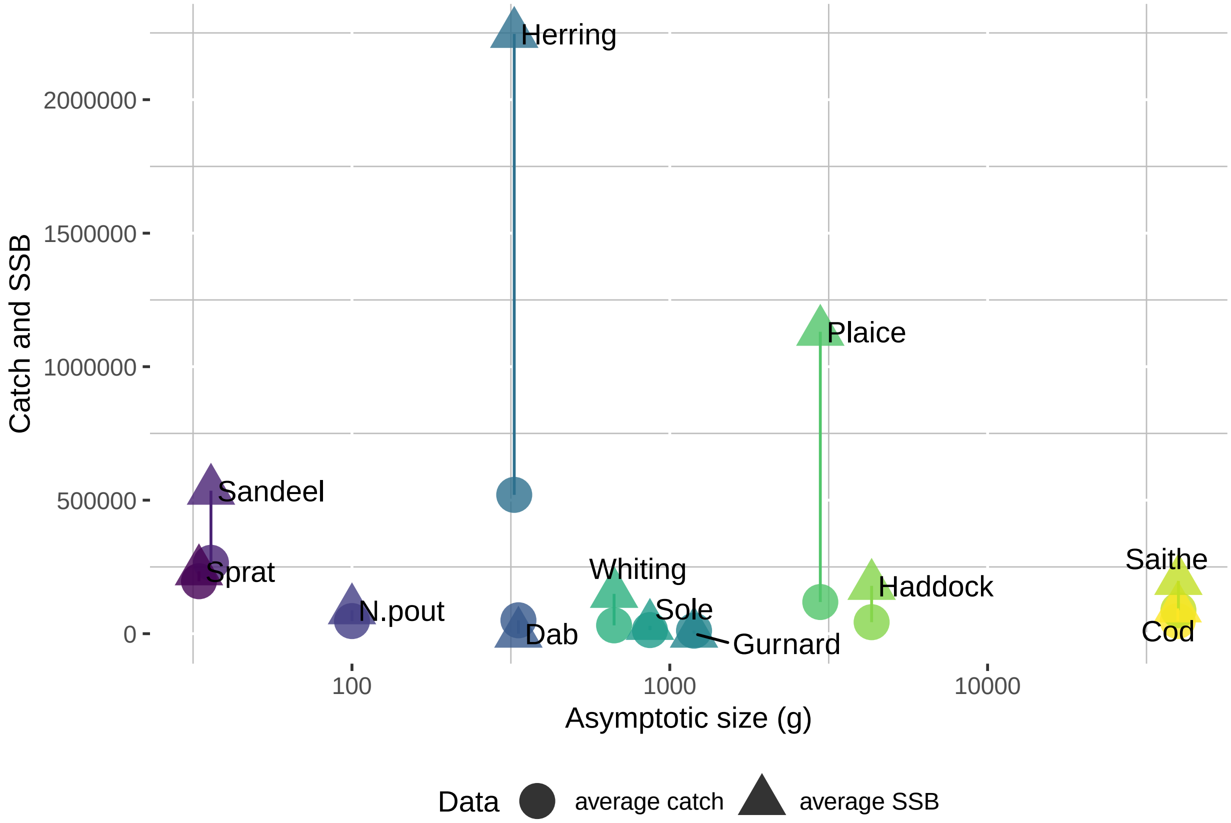
```
##              Sprat    Sandeel     N.pout        Dab    Herring    Gurnard
## Sprat   0.72912919 0.03408440 0.06354517 0.36241552 0.27416982 0.17515576
## Sandeel 0.03408440 0.68119882 0.04892432 0.09736663 0.05888214 0.05992649
## N.pout  0.06354517 0.04892432 0.79660429 0.09088798 0.29755069 0.30624141
## Dab     0.36241552 0.09736663 0.09088798 0.80817768 0.28963957 0.22041200
## Herring 0.27416982 0.05888214 0.29755069 0.28963957 0.65890104 0.27510627
## Gurnard 0.17515576 0.05992649 0.30624141 0.22041200 0.27510627 0.88010500
## Sole    0.29795558 0.06020860 0.01679020 0.38047464 0.20014139 0.10677895
## Whiting 0.26525924 0.07510011 0.29989886 0.33389727 0.37373927 0.37109904
## Plaice  0.37065975 0.07801855 0.07855818 0.56492206 0.27791867 0.16492120
## Haddock 0.08135547 0.09395730 0.54917554 0.13168065 0.34835469 0.35735444
## Saithe  0.01681321 0.01609022 0.29498937 0.03138197 0.12620591 0.12351994
## Cod     0.33757969 0.09943453 0.32502256 0.41647801 0.40477930 0.35183282
##               Sole    Whiting     Plaice    Haddock     Saithe        Cod
## Sprat   0.29795558 0.26525924 0.37065975 0.08135547 0.01681321 0.33757969
## Sandeel 0.06020860 0.07510011 0.07801855 0.09395730 0.01609022 0.09943453
## N.pout  0.01679020 0.29989886 0.07855818 0.54917554 0.29498937 0.32502256
## Dab     0.38047464 0.33389727 0.56492206 0.13168065 0.03138197 0.41647801
## Herring 0.20014139 0.37373927 0.27791867 0.34835469 0.12620591 0.40477930
## Gurnard 0.10677895 0.37109904 0.16492120 0.35735444 0.12351994 0.35183282
## Sole    0.71558049 0.19227455 0.39137317 0.03447799 0.01242055 0.25761229
## Whiting 0.19227455 0.70928230 0.29503807 0.39164787 0.10228168 0.44060879
## Plaice  0.39137317 0.29503807 0.71922391 0.11248513 0.03294939 0.35043671
## Haddock 0.03447799 0.39164787 0.11248513 0.85830725 0.26167470 0.39577341
## Saithe  0.01242055 0.10228168 0.03294939 0.26167470 0.66383553 0.20894496
## Cod     0.25761229 0.44060879 0.35043671 0.39577341 0.20894496 0.78654705
```

The second type of data you are going to need is a data set of catch and/or spawning stock biomass (SSB) of the selected species, so we can compare the model output's to real data. The fisheries time-series of the North Sea are also avaible on this repository.

The following .csv are extracted from the ICES database using "data/getICESFishdata_param.R". Fishing data is averaged over 2014-2019 as it is a relatively stable period in catches and has the maximum amount of data across all species concerned.

```
# fisheries mortality F
fMat <- read.csv("../data/fmat.csv")
# fMatWeighted <- read.csv("data/fmatWeighted.csv") # Sandeel and Cod have multiple data base so averag
fMatWeighted <- readRDS("../data/FmatWeightedInterpolated.rds") # to get Gurnard data


# read in time-averaged  catches
catchAvg <-read.csv("../data/time-averaged-catches.csv") # only that one is used at the moment | catche

# ssb
ssbAvg <- read.csv("../data/time-averaged-SSB.csv")
```

```r
# plot_dat <- reshape2::melt(fMatWeighted,"X")
# colnames(plot_dat) <- c("Time", "Species", "F")
#
# ggplot(plot_dat)+
#   geom_line(aes(x = Time, y = F, color = Species))+
#   scale_y_continuous(name = "Fisheries mortality rate") +
#   scale_color_manual(values = params_uncalibrated@linecolour) +
#   theme(                   panel.background = element_blank(),
#                    panel.grid.minor = element_line(color = "gray"),
#                    legend.key = element_rect(fill = "white"))



# overlapping fisheries mortality with next plot

# plot_dat <- apply(as.matrix(fMatWeighted[67:73,]),2,mean, na.rm = T)
# plot_dat <- plot_dat[2:length(plot_dat)]
# plot_dat <- melt(plot_dat)
# plot_dat$Species <- factor(as.character(rownames(plot_dat)),levels = c(as.character(params_uncalibrat
# plot_dat <- plot_dat[order(plot_dat$Species),]
# plot_dat$w_inf <- params_uncalibrated@species_params$w_inf
# plot_dat2 <- plot_dat


plot_dat <- data.frame(catchAvg,ssbAvg)
plot_dat$species.1 <- NULL
colnames(plot_dat) <- c("Species", "average catch", "average SSB")
plot_dat$Species <- factor(as.character(plot_dat$Species),levels = c(as.character(nsParams$species)))
plot_dat <- reshape2::melt(plot_dat,"Species")
plot_dat$w_inf <- rep(nsParams$w_inf,2)


  # ggplot(plot_dat) +
  #   geom_bar(aes(x = Species,y = value, fill = Species, alpha = variable), stat = "identity", positio
  #   geom_point(aes(x = Species))
  #   scale_y_continuous(trans = "log10", name = "Catch (clear) and SSB (solid)") + #, limits = c(0.5*m
  #   scale_fill_manual(name = "Species", values = params_uncalibrated@linecolour) +
  #   scale_alpha_manual(name = "Stat", values = c(0.5, 1), labels = c("Catch", "SSB")) +
  #   theme(
  #         legend.position = "none", legend.key = element_rect(fill = "white"))



ggplot(plot_dat)+
  geom_point(aes(x = w_inf, y = value, color = Species, shape = variable), size = 6, alpha = .8) +
  # geom_point(data = plot_dat2, aes(x = w_inf, y = value*1562500, color = Species, shape = "averaged f
  geom_text_repel(data = filter(plot_dat,variable == "average SSB"), aes(x = w_inf, y = value, label = S
  geom_line(aes(x = w_inf, y = value, color = Species)) +
  scale_y_continuous(name = "Catch and SSB", limits = c(0,NA)) +#,sec.axis = sec_axis(trans = ~./1562500
  scale_x_continuous(name = "Asymptotic size (g)", trans = "log10") +
  scale_color_manual(name = "Species", values = viridis(dim(nsParams)[1])) +
  scale_shape_manual(name = "Data", values = c(16,17)) + # add 4 if fisheries mortality present
  theme(panel.background = element_blank(),
        panel.grid.minor = element_line(color = "gray"),
        legend.position = "bottom",legend.key = element_rect(fill = "white"))+
  guides(color = FALSE)
```



This plot shows the averaged data over the 2014-2019 period. Species are ordered per asympotic size on the x-axis. There are 2 data points per species. The average catch (disc) and average SSB (triangle) are linked by a line. Units is . . .

RF: Dab and Gurnard have some issues, it comes from the data though, the calculations are correct.

Step 2 - How to convert the data into a valid `mizerParams` object

In this section you will:

- learn to format raw data into a Mizer compatible format

Inputing the previous data frame and interaction matrix into the `newMultispeciesParams()` function output a fully fleshed `mizerParams` object

```
param <- newMultispeciesParams(smallExample,smallInter)
```

```
## Note: No h provided for some species, so using f0 and k_vb to calculate it.
```

```
## Note: Because you have n != p, the default value is not very good.
## Note: No ks column so calculating from critical feeding level.

## Note: Using z0 = z0pre * w_inf ^ z0exp for missing z0 values.

## Note: Using f0, h, lambda, kappa and the predation kernel to calculate gamma.
```

```
param@species_params
```

```
##         species w_inf k_vb w_mat w_min alpha interaction_resource         n   p
## george   george   1.6  0.6 0.400 0.001   0.6                    1 0.6666667 0.7
## ginette ginette   1.7  1.0 0.425 0.001   0.6                    1 0.6666667 0.7
##                 q pred_kernel_type beta sigma        h k        ks        z0
## george  0.7166667        lognormal   30     2 4.804173 0 0.5943805 0.5129928
## ginette 0.7166667        lognormal   30     2 8.196074 0 1.0119852 0.5027302
##                gamma   w_mat25 m erepro R_max
## george  1.206953e-11 0.3583834 1      1   Inf
## ginette 2.059102e-11 0.3807823 1      1   Inf
```

Let's do the same for our North Sea data set and look at what parameters we can find in the `mizerParams` object:

```
params_uncalibrated <- newMultispeciesParams(nsParams, inter, kappa = 1e11, max_w=1e6) # inter comes wi
```

```
## Note: No h provided for some species, so using f0 and k_vb to calculate it.

## Note: Because you have n != p, the default value is not very good.

## Note: No ks column so calculating from critical feeding level.

## Note: Using z0 = z0pre * w_inf ^ z0exp for missing z0 values.

## Note: Using f0, h, lambda, kappa and the predation kernel to calculate gamma.
```

```
params_uncalibrated@species_params$erepro <- .01 # default is 1

# note the volume of this model is set to the reflect the entire volume of the North Sea - hence the ve

#  Add other params for info
#  param$Volumecubicmetres=5.5e13    #unit of volume. Here total volume of North sea is used (Andersen 

# have a look at species parameters that have been calculated
# params_uncalibrated@species_params

# alternative params without redundant parameters to reduce the size of the dataframe on the screeen
params_uncalibrated@species_params[,-which(colnames(params_uncalibrated@species_params) %in%
                                              c("sel_func","gear","interaction_resource","pred_kernel_ty
```

```
##         X1 species   w_inf w_mat    beta sigma R_max  k_vb  l25  l50     a     b
## Sprat    1   Sprat    33.0    13  51076    0.8   Inf 0.681  7.6  8.1 0.007 3.014
## Sandeel  2 Sandeel    36.0     4 398849    1.9   Inf 1.000  9.8 11.8 0.001 3.320
## N.pout   3  N.pout   100.0    23     22   1.5   Inf 0.849  8.7 12.2 0.009 2.941
## Dab      5     Dab   324.0    21    191   1.9   Inf 0.536 11.5 17.0 0.010 2.986
## Herring  4 Herring   334.0    99 280540   3.2   Inf 0.606 10.1 20.8 0.002 3.429
## Gurnard  8 Gurnard   668.0    39    283   1.8   Inf 0.266 19.8 29.0 0.004 3.198
## Sole     7    Sole   866.0    78    381   1.9   Inf 0.284 16.4 25.8 0.008 3.019
## Whiting  6 Whiting  1192.0    75     22   1.5   Inf 0.323 19.8 29.0 0.006 3.080
## Plaice   9  Plaice  2976.0   105    113   1.6   Inf 0.122 11.5 17.0 0.007 3.101
## Haddock 10 Haddock  4316.5   165    558   2.1   Inf 0.271 19.1 24.3 0.005 3.160
## Saithe  12  Saithe 39658.6  1076     40   1.1   Inf 0.175 35.3 43.6 0.007 3.075
## Cod     11     Cod 39851.3  1606     66   1.3   Inf 0.216 13.2 22.9 0.005 3.173
##         catchability        h k       ks         z0        gamma     w_mat25
## Sprat     1.29533333 14.46675 0 1.593753 0.18705957 5.652974e-11   11.647460
## Sandeel   0.06510547 25.62741 0 2.936414 0.18171206 3.790575e-11    3.583834
## N.pout    0.31380000 31.20422 0 3.372902 0.12926608 9.750228e-11   20.607045
## Dab       0.97800000 34.87720 0 3.781368 0.08735805 7.579184e-11   18.815128
## Herring   0.18150000 28.36363 0 2.920263 0.08647736 2.514308e-11   88.699888
## Gurnard   0.46250569 20.64990 0 2.193126 0.06863713 4.638552e-11   34.942380
## Sole      0.37383333 24.73805 0 2.567302 0.06294752 5.184323e-11   69.884760
## Whiting   0.24266667 31.77220 0 3.301616 0.05658819 9.927702e-11   67.196884
## Plaice    0.18483333 16.94072 0 1.740765 0.04171321 4.489177e-11   94.075638
## Haddock   0.30150000 41.46028 0 4.196598 0.03685027 7.707429e-11  147.833146
## Saithe    0.39300000 59.95343 0 5.700788 0.01759431 2.435635e-10  964.051303
## Cod       0.26666749 69.40226 0 6.511735 0.01756590 2.325827e-10 1438.909287
##         erepro
## Sprat     0.01
## Sandeel   0.01
## N.pout    0.01
## Dab       0.01
## Herring   0.01
## Gurnard   0.01
## Sole      0.01
## Whiting   0.01
## Plaice    0.01
## Haddock   0.01
## Saithe    0.01
## Cod       0.01
```

RF: should this parameter description be shown sooner?

w_inf: asymptotic size

w_mat: maturation size (determines when 50% of the population has matured / not sure!)

beta: preferred predator/prey mass ratio

sigma: width of the feeding kernel

R_max: Beverton-Holt density dependence parameter

k_vb: von Bertalanffy growth parameter

l25: length at …

l50: length at …

a: coefficient for age to size conversion

b: constant for age to size conversion

catchability: fisheries efficiency

h: maximum intake rate

k: metabolism constant

ks: metabolism coefficient

z0: background mortality coefficient

gamma: search volume (obtained from beta and sigma)

w_mat25: weight at which 25% of individuals are mature

erepro: coefficent that weights reproductive output

Step 3 - Checking assumptions

In this section you will:

???

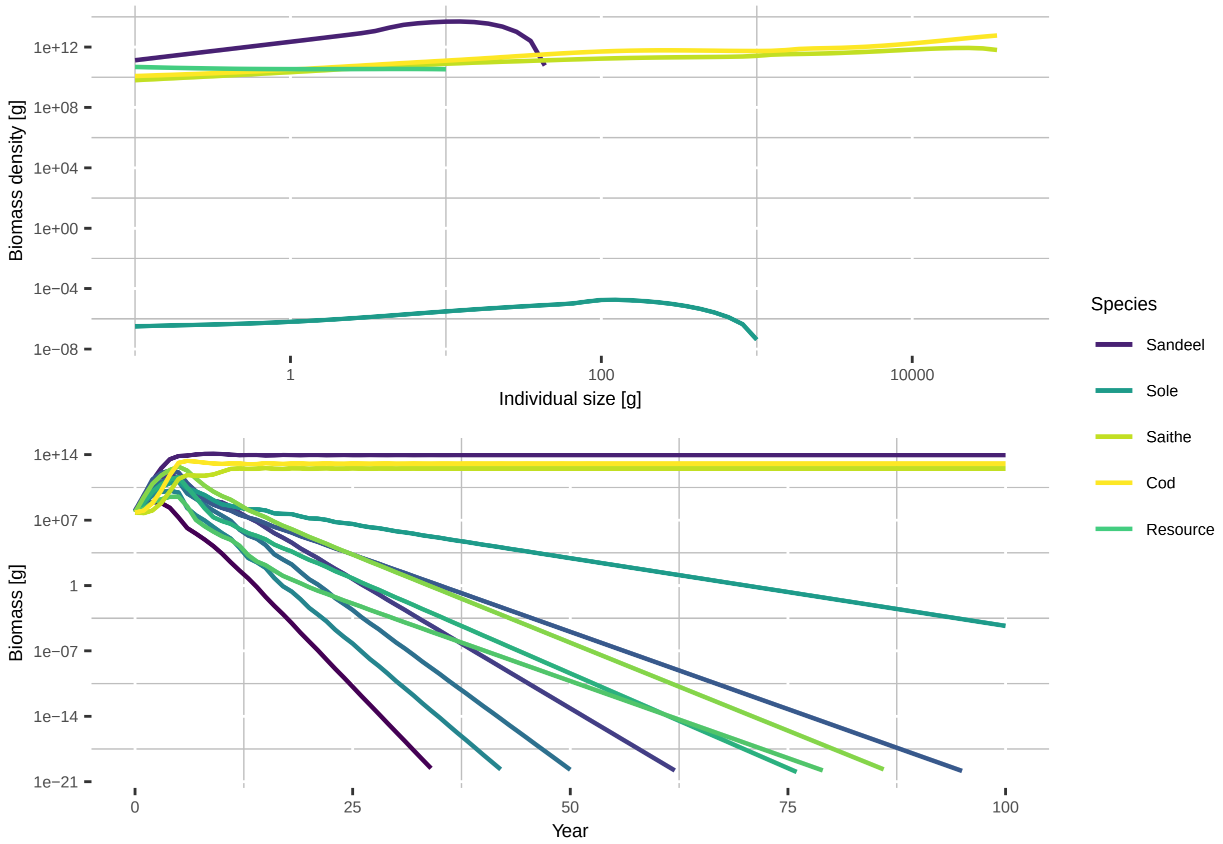
Step 4 - Running and exploring the model for the first time to check coexistence (e.g. first guess)

In this section you will:

- Try to get coexistence between your species manually

First, let's use the `project()` function with our `mizerParams` object to project in time our initial ecosystem and see if species manage to coexist together.

```r
# run with fishing
sim_uncalibrated <- project(params_uncalibrated, t_max = 100, effort = 1)

plotSummary(sim_uncalibrated, short = T)
```

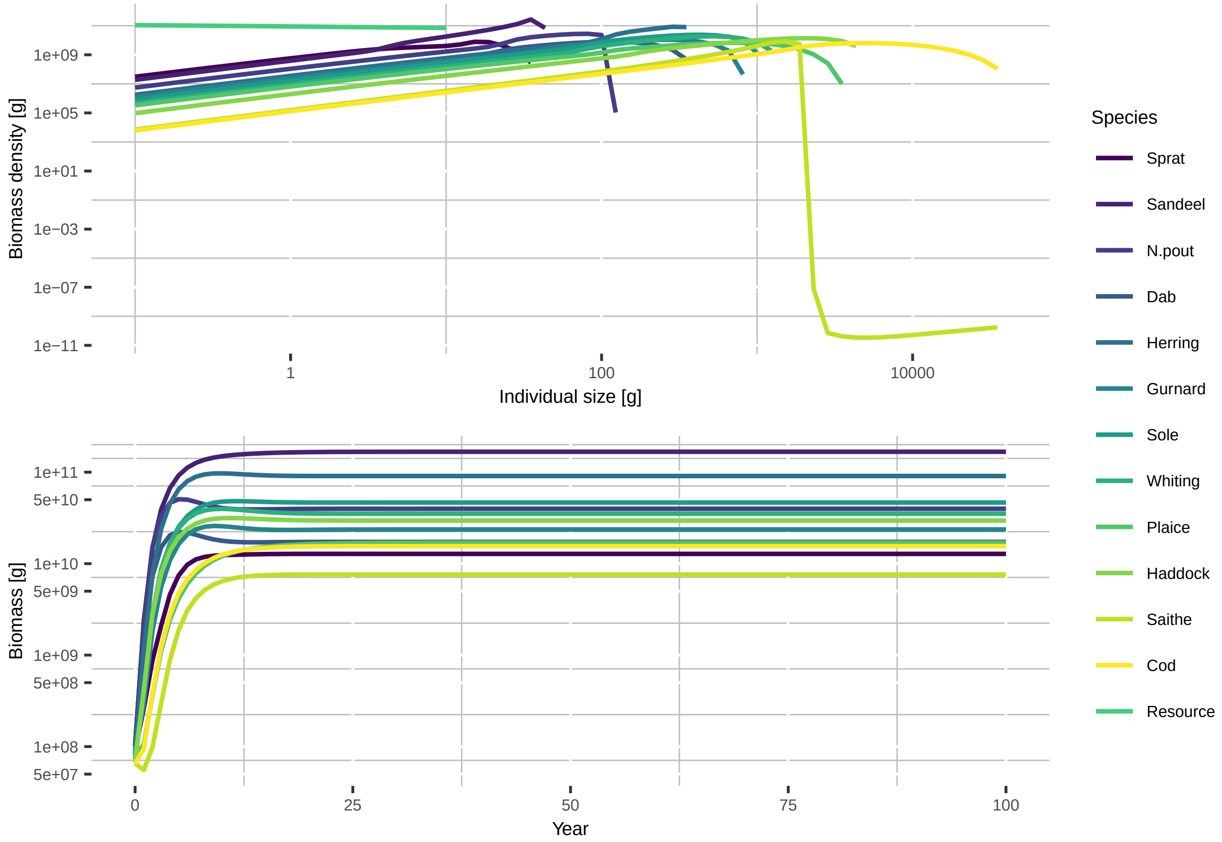


The top panel shows the different species size spectrum at the last time step of the simulation while the bottom panel shows the abundance per species through time. These plots show that species do not coexist and several go extinct. This is because there was no external density dependence (R_{max} is set at Inf) and the largest species (Cod and Saithe) are out-competing the rest.

RF: should we put earlier a typical looking size spectrum?

To get coexistence one needs to guess reasonable R_{max} values which will stop out-competition from a few species. We assume that a the density dependence is going to be posively related to body size, meaning that large individuals will have a stronger density-dependence applied to them (and therefore a lower R_{max} value)

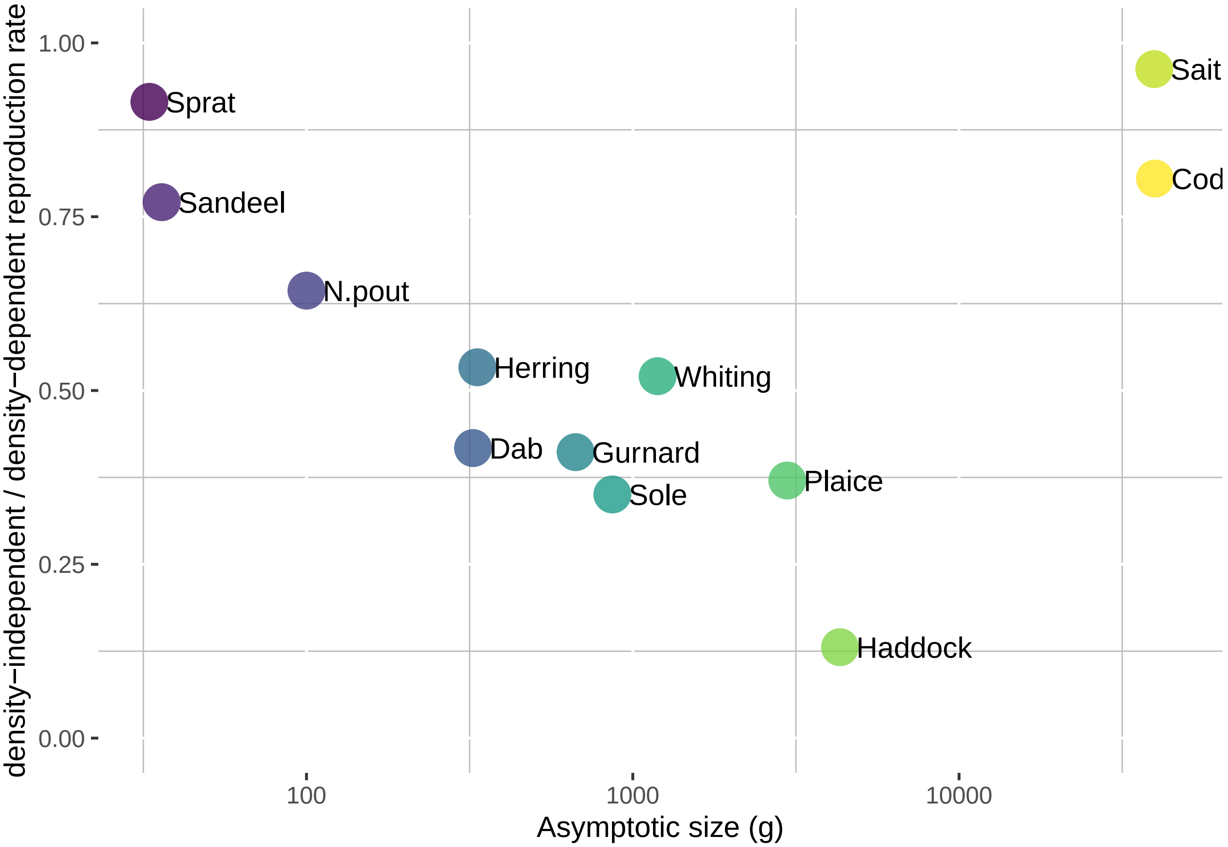
```
# let's start again and replace with the initial pre-calibration "guessed" Rmax
params_guessed <- params_uncalibrated
# penalise the large species with higher density dependence
params_guessed@species_params$R_max <- params_guessed@resource_params$kappa*params_guessed@species_para
# and reduce erepro
# params_guessed@species_params$erepro <- 1e-3

params_guessed <- setParams(params_guessed)
# run with fishing
sim_guessed <- project(params_guessed, t_max = 100, effort =1)
plotSummary(sim_guessed, short = T)
```



The ecosystem looks way better. Saithe's largest individuals are having a hard time, but at least species coexist.

R_{max} affects the density-dependent reproduction rate (RDD) but limiting the maximum amout of spawn calculated from the density-independent reproduction rate (RDI). Let's look at the RDD/RDI ratio to see how strong acts R_{max} on our different species.

```
  plot_dat <- as.data.frame(getRDD(sim_guessed@params)/getRDI(sim_guessed@params))
  plot_dat$species <- factor(rownames(plot_dat),sim_guessed@params@species_params$species)
  colnames(plot_dat)[1] <- "ratio"
  plot_dat$w_inf <- sim_guessed@params@species_params$w_inf

# ggplot(plot_dat) +
#   geom_bar(aes(x = species, y = ratio, fill = species),stat="identity") +
#   scale_fill_manual(name = "Species", values = sim_guessed@params@linecolour) +
#   scale_y_continuous(name = "density-independent / density-dependent reproduction rate", trans = "log
#   scale_x_discrete(name = "Species") +
#    theme(panel.background = element_rect(fill = "white", color = "black"),
#         legend.position = "none")

ggplot(plot_dat)+
  geom_point(aes(x = w_inf, y = ratio, color = species), size = 6, alpha = .8) +
  geom_text(aes(x = w_inf, y = ratio, label = species), hjust = 0, nudge_x = 0.05)+
  # geom_line(aes(x = w_inf, y = value, color = Species)) +
  scale_y_continuous(name = "density-independent / density-dependent reproduction rate", limits = c(0,1
  scale_x_continuous(name = "Asymptotic size (g)", trans = "log10") +
  scale_color_manual(name = "Species", values = params_uncalibrated@linecolour) +
  theme(panel.background = element_blank(),
                   panel.grid.minor = element_line(color = "gray"),
       legend.position = "none")
```



Is the physiological recruitment, RDI, much higher than the realised recruitment, RDD? Low RDD/RDI ratio indicates strong density dependence, meaning that the carrying capacity is controlling the population rather than predation or competition. Larger species often require more of this density dependent control than smaller ones. If RDD/RDI is too low, the efficiency of reproduction ($erepro$) can be lowered to ensure species do not outcompete others or or over-resilient to fishing. The largest species that were the most limited by our new R_{max} do not show a strong density dependence. The medium-sized species are the most affected here.