

How to Mizer - how to parametrise a Mizer model

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

2021-05-12

Beginner level tutorial - how to parameterise a Mizer model

In this tutorial you will learn

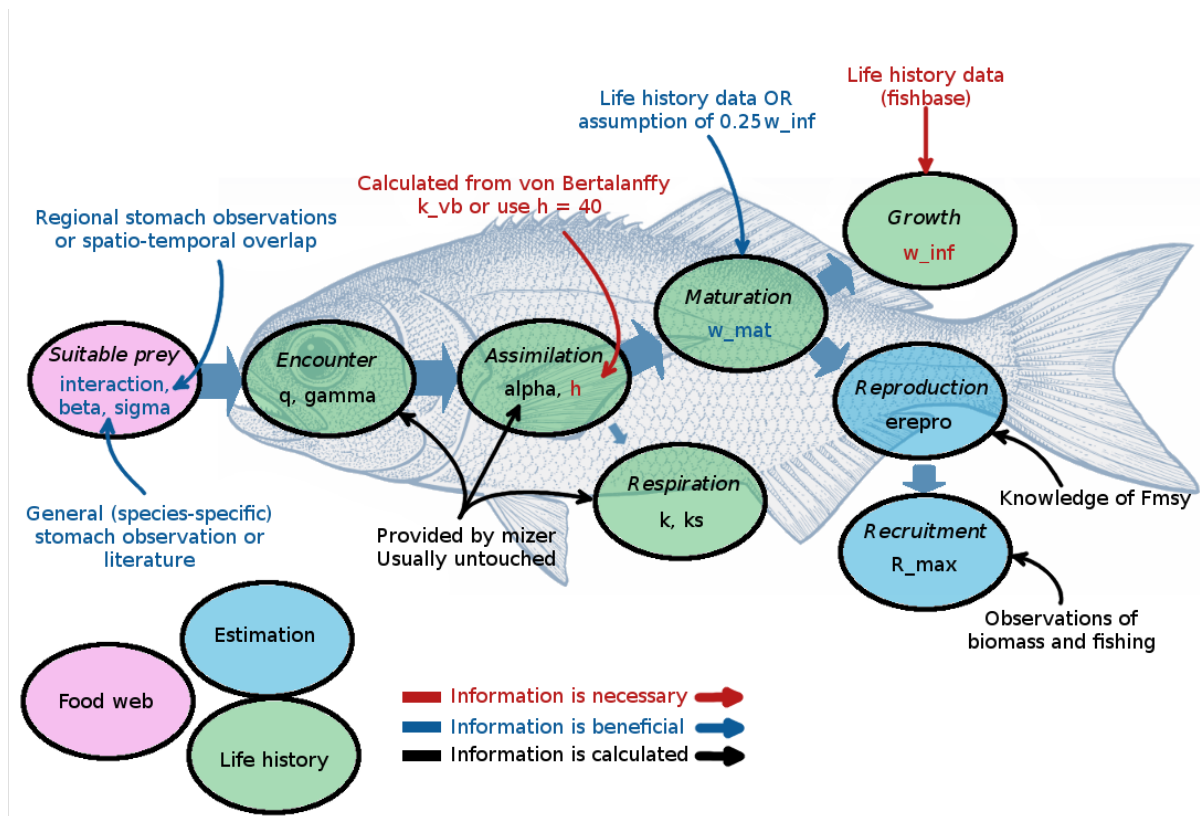
- What type of data is needed to parametrise a Mizer model
- How to convert raw data into a `mizerParams` object
- Check the several assumptions you are making about your parameters
- Explore the model for the first time with your own data set

Step 1 - What data are typically needed to parameterise a mizer model?

The multispecies model in mizer allows you to resolve species-specific differences in life history and feeding parameters that are important for modelling particular ecosystems.

Figure 1 is an illustration of the parameters needed (blue boxes) and calibrated (green boxes) by Mizer. The need for data is hierarchical: a model can be setup and calibrated with the information in red: knowledge of the asymptotic size and observations of biomass and fishing. The calibration can be refined by adding further information in life history parameters and by using knowledge of Fmsy to calibrate the reproductive efficiency (orange). Additional refinement can be done by specifying the interaction matrix, theta (blue). Other parameters can be adjusted but they are rarely known accurately on a species-by-species basis (black).

```
library(knitr)
# All defaults
include_graphics("../paper/parameterSketch.png")
```



Let's start with the minimal amount of information

First you are going to need a data frame of species specific 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 availability of each predator and prey species).

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

Let's start with a small made up example assuming we have already found some parameters from FishBase or the literature.

Here we consider two interacting species and a background resource spectrum (we will come back to this). Using the North Sea as an example, let's start with the iconic Atlantic cod and a key focal prey species sandeel.

Remember to minimally set up this model we can look on Fishbase (www.fishbase.se or use the `rfishbase` package) to find the asymptotic weights of these two species.

A search on fishbase reveals some parameters from empirical von Bertalanffy growth curves which can be used to set up the life history parameters - `w_inf` and `k_vb`.

Often von Bertalanffy relationships are based on length not weight and the length parameters can be converted to weight using length-weight regressions.

We find an asymptotic length of 132 cm and k of 0.2 for cod in the North Sea here: <https://www.fishbase.se/popdyn/PopGrowthList.php?ID=69&GenusName=Gadus&SpeciesName=morhua&fc=183>

And for sandeel, 18.5 cm and $k_{vb}=0.4$

<https://www.fishbase.se/popdyn/PopGrowthList.php?ID=37&GenusName=Ammodytes&SpeciesName=marinus&fc=402>

We then need to convert these asymptotic lengths to weights using length-weight regression parameters, which also can be found on fishbase or in the literature.

For cod:

<https://www.fishbase.se/popdyn/LWRelationshipList.php?ID=69&GenusName=Gadus&SpeciesName=morhua&fc=183>

this estimates a $w_{inf} = 24600$ g

For sandeel: <https://www.fishbase.se/popdyn/LWRelationshipList.php?ID=37&GenusName=Ammodytes&SpeciesName=marinus&fc=402>

this estimates a $w_{inf} = 22.83$ g

If w_{inf} or l_{inf} are not available you could use maximum observed sizes, these values are often systematically larger than estimates of asymptotic weight, and we recommend you check the literature or size-at-age data for your system to check reliability of fishbase estimates.

We also know from the literature that the preferred predator-prey mass ratio for cod is approx. 100 for cod and we might guess that it is approx. 10000 for sandeel as they feed on prey much smaller than themselves. This means that cod of any given size are 100 times larger than their preferred prey size and sandeels are 10000 larger than theirs.

For simplicity, we will assume defaults for all other parameters and that both species occur in the same environment throughout their lives (interaction matrix = 1).

```
smallExample <- data.frame("species" = c("sandeel","cod"), "w_inf" = c(24600,23), "k_vb" = c(0.2,0.1), "beta" = c(100,10000))
smallExample
```

```
##   species w_inf k_vb  beta
## 1 sandeel 24600  0.2   100
## 2      cod    23  0.1 10000
```

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

```
##           sandeel cod
## sandeel         1   1
## cod             1   1
```

The ‘smallExample’ shows the format of the data.frame for the species parameters and the ‘smallInter’ shows the format of the interaction matrix. Both of these are required as inputs for creating a “mizerParams” object. Many mizer users will collect this information in a spreadsheet prior to reading in the data into mizer.

To illustrate this next step we will use pre-existing set of species’ specific parameters for a previously published North Sea model, that used more detailed fisheries dependent and independent data to calculate species parameters.

These files are available in the “How to Mizer” GitHub repository as shown below.

```
# loading North Sea species parameters
nsParams <- read.csv("../data/nsparams0.csv")
inter <- read.csv("../data/inter.csv")
inter<- data.matrix(inter[-1])
rownames(inter) =nsParams$species

# more information on how to set this up is here:
#https://sizespectrum.org/mizer/articles/a_multispecies_model_of_the_north_sea.html

nsParams
```

```
##      species    w_inf w_mat    beta sigma    k_vb      sel_func  125  150    a
## 1   Sprat      33.0    13  51076    0.8 0.681 sigmoid_length  7.6  8.1 0.007
## 2  Sandeel     36.0     4 398849    1.9 1.000 sigmoid_length  9.8 11.8 0.001
## 3   N.pout    100.0    23    22    1.5 0.849 sigmoid_length  8.7 12.2 0.009
## 4     Dab     324.0    21    191    1.9 0.536 sigmoid_length 11.5 17.0 0.010
## 5  Herring    334.0    99 280540    3.2 0.606 sigmoid_length 10.1 20.8 0.002
## 6  Gurnard    668.0    39    283    1.8 0.266 sigmoid_length 19.8 29.0 0.004
## 7     Sole    866.0    78    381    1.9 0.284 sigmoid_length 16.4 25.8 0.008
## 8  Whiting   1192.0    75     22    1.5 0.323 sigmoid_length 19.8 29.0 0.006
## 9   Plaice   2976.0   105    113    1.6 0.122 sigmoid_length 11.5 17.0 0.007
## 10 Haddock   4316.5   165    558    2.1 0.271 sigmoid_length 19.1 24.3 0.005
## 11 Saithe  39658.6  1076     40    1.1 0.175 sigmoid_length 35.3 43.6 0.007
## 12     Cod  39851.3  1606     66    1.3 0.216 sigmoid_length 13.2 22.9 0.005
##      b catchability      gear
## 1  3.014  1.29533333 sigmoid_gear
## 2  3.320  0.06510547 sigmoid_gear
## 3  2.941  0.31380000 sigmoid_gear
## 4  2.986  0.97800000 sigmoid_gear
## 5  3.429  0.18150000 sigmoid_gear
## 6  3.198  0.46250569 sigmoid_gear
## 7  3.019  0.37383333 sigmoid_gear
## 8  3.080  0.24266667 sigmoid_gear
## 9  3.101  0.18483333 sigmoid_gear
## 10 3.160  0.30150000 sigmoid_gear
## 11 3.075  0.39300000 sigmoid_gear
## 12 3.173  0.26666749 sigmoid_gear
```

```
inter
```

```
##      Sprat    Sandeel    N.pout    Herring    Dab    Whiting
## Sprat  0.72912919 0.03408440 0.06354517 0.27416982 0.36241552 0.26525924
## Sandeel 0.03408440 0.68119882 0.04892432 0.05888214 0.09736663 0.07510011
## N.pout  0.06354517 0.04892432 0.79660429 0.29755069 0.09088798 0.29989886
## Dab     0.27416982 0.05888214 0.29755069 0.65890104 0.28963957 0.37373927
## Herring 0.36241552 0.09736663 0.09088798 0.28963957 0.80817768 0.33389727
## Gurnard 0.26525924 0.07510011 0.29989886 0.37373927 0.33389727 0.70928229
## Sole    0.29795558 0.06020860 0.01679020 0.20014139 0.38047464 0.19227456
## Whiting 0.17515576 0.05992649 0.30624141 0.27510627 0.22041200 0.37109904
## Plaice  0.37065975 0.07801855 0.07855818 0.27791867 0.56492206 0.29503807
## Haddock 0.08135547 0.09395730 0.54917554 0.34835469 0.13168065 0.39164787
## Saithe  0.01681321 0.01609022 0.29498937 0.12620591 0.03138197 0.10228168
```

## Cod	0.33757969	0.09943453	0.32502256	0.40477930	0.41647801	0.44060879
## Sole						
## Gurnard						
## Plaice						
## Haddock						
## Saithe						
## Cod						
## Sprat	0.29795558	0.17515576	0.37065975	0.08135547	0.01681321	0.33757969
## Sandeel	0.06020860	0.05992649	0.07801855	0.09395730	0.01609022	0.09943453
## N.pout	0.01679020	0.30624141	0.07855818	0.54917554	0.29498937	0.32502256
## Dab	0.20014139	0.27510627	0.27791867	0.34835469	0.12620591	0.40477930
## Herring	0.38047464	0.22041200	0.56492206	0.13168065	0.03138197	0.41647801
## Gurnard	0.19227456	0.37109904	0.29503807	0.39164787	0.10228168	0.44060879
## Sole	0.71558049	0.10677895	0.39137317	0.03447799	0.01242055	0.25761229
## Whiting	0.10677895	0.88010500	0.16492120	0.35735444	0.12351994	0.35183282
## Plaice	0.39137317	0.16492120	0.71922391	0.11248513	0.03294939	0.35043671
## Haddock	0.03447799	0.35735444	0.11248513	0.85830725	0.26167470	0.39577341
## Saithe	0.01242055	0.12351994	0.03294939	0.26167470	0.66383553	0.20894497
## Cod	0.25761229	0.35183282	0.35043671	0.39577341	0.20894497	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 available in this repository. We will use these data later in our second tutorial on calibration.

Because the North Sea is heavily fished we also need information on fishing intensity and other parameters have been entered into the species parameter file that relate to the type of fishing gear selectivity (which is assumed to be species-specific).

In mizer, fishing mortality rates at size for each gear are calculated as

$$F = \text{catchability} \times \text{selectivity} \times \text{effort}$$

The selectivity parameters are determined from `l_25`, `l_50`, and `sel_func` but simpler approaches can also be used. See <https://sizespectrum.org/mizer/reference/setFishing.html>

For the North Sea we assumed $\text{catchability} \times \text{effort}$ could be estimated from the fishing mortality rates of fully selected sizes/ages of fish from single-species stock assessments. The following .csv are extracted from the ICES stock assessment database using "data/getICESFishdata_param.R". Fishing mortality 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.

** Note the species are not in the right order here, but we can fix that later...

```
# fisheries mortality F
fMat <- read.csv("../data/fmat.csv")
# fMatWeighted <- read.csv("data/fmatWeighted.csv") # Sandeel and Cod have multiple data base so averaged
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 / catches

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

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

Inputting 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  beta   w_mat w_min alpha interaction_resource
## sandeel sandeel 24600 0.2   100 6150.00 0.001   0.6                      1
## cod      cod    23  0.1 10000   5.75 0.001   0.6                      1
##      n  p      q pred_kernel_type sigma      h k      ks
## sandeel 0.6666667 0.7 0.7166667      lognormal      2 45.821729 0 4.1110890
## cod     0.6666667 0.7 0.7166667      lognormal      2  2.126856 0 0.2407671
##      z0      gamma   w_mat25 m erepro R_max
## sandeel 0.02063033 1.002286e-10 5510.144528 1      1      Inf
## cod     0.21098033 3.661644e-12   5.151761 1      1      Inf
```

You can see that `mizer` has estimated several missing parameters. [Go through each of these, take a look at the above figure again]

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 with
```

```
## Note: Dimnames of interaction matrix do not match the order of species names in the species data.frame
```

```
## 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 screen
params_uncalibrated@species_params[,~which(colnames(params_uncalibrated@species_params) %in%
c("sel_func", "gear", "interaction_resource", "pred_kernel_ty

```

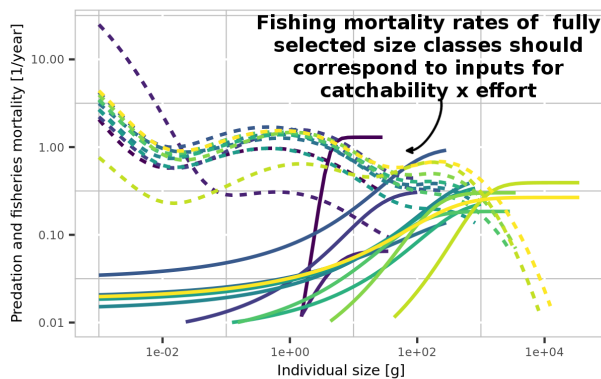
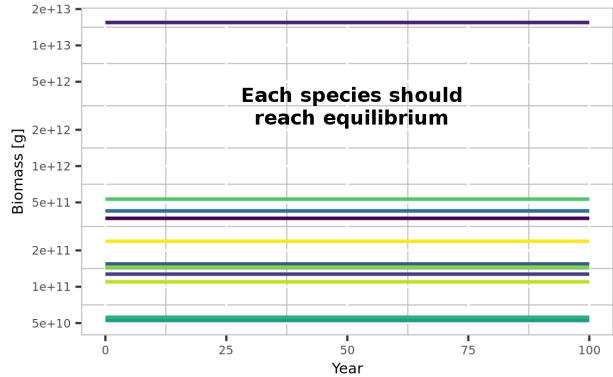
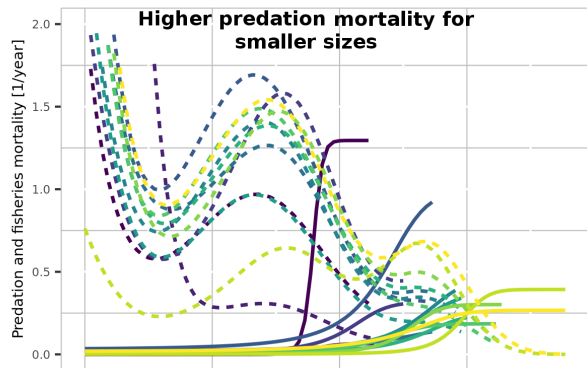
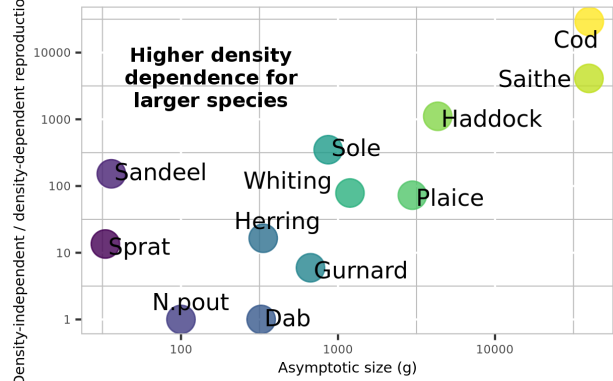
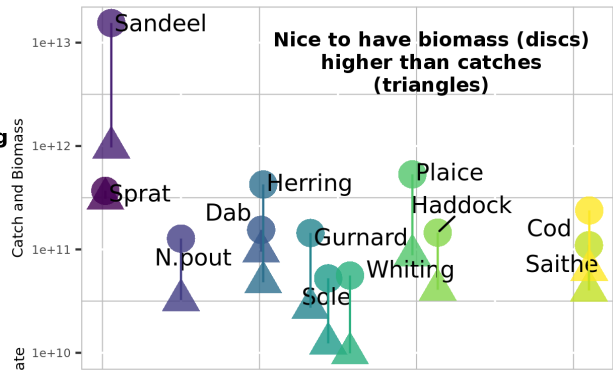
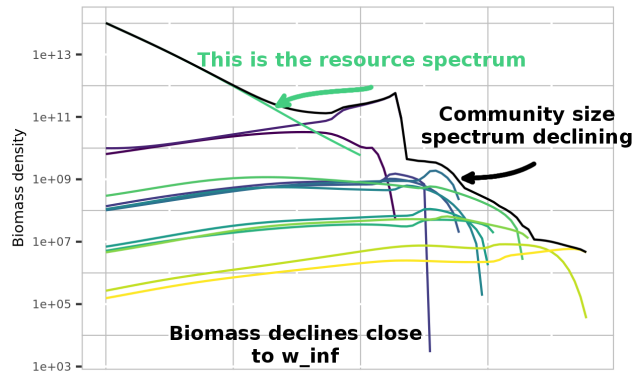
##	species	w_inf	w_mat	beta	sigma	k_vb	l25	l50	a	b	
##	Sprat	Sprat	33.0	13	51076	0.8	0.681	7.6	8.1	0.007	3.014
##	Sandeel	Sandeel	36.0	4	398849	1.9	1.000	9.8	11.8	0.001	3.320
##	N.pout	N.pout	100.0	23	22	1.5	0.849	8.7	12.2	0.009	2.941
##	Dab	Dab	324.0	21	191	1.9	0.536	11.5	17.0	0.010	2.986
##	Herring	Herring	334.0	99	280540	3.2	0.606	10.1	20.8	0.002	3.429
##	Gurnard	Gurnard	668.0	39	283	1.8	0.266	19.8	29.0	0.004	3.198
##	Sole	Sole	866.0	78	381	1.9	0.284	16.4	25.8	0.008	3.019
##	Whiting	Whiting	1192.0	75	22	1.5	0.323	19.8	29.0	0.006	3.080
##	Plaice	Plaice	2976.0	105	113	1.6	0.122	11.5	17.0	0.007	3.101
##	Haddock	Haddock	4316.5	165	558	2.1	0.271	19.1	24.3	0.005	3.160
##	Saithe	Saithe	39658.6	1076	40	1.1	0.175	35.3	43.6	0.007	3.075
##	Cod	Cod	39851.3	1606	66	1.3	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	R_max									
##	Sprat	0.01	Inf								
##	Sandeel	0.01	Inf								
##	N.pout	0.01	Inf								
##	Dab	0.01	Inf								
##	Herring	0.01	Inf								
##	Gurnard	0.01	Inf								
##	Sole	0.01	Inf								
##	Whiting	0.01	Inf								
##	Plaice	0.01	Inf								
##	Haddock	0.01	Inf								
##	Saithe	0.01	Inf								
##	Cod	0.01	Inf								

Step 3 - Checking assumptions

In this section you will run the model to equilibrium using the function `project()` and will be able to plot the model simulation output.

First, let's go through an example we have made previously to provide some tips of what to look for once you run the model.

```
# All defaults  
include_graphics("../paper/tempSummaryAnnotated.png")
```

Species

in order from smaller to larger species

Sprat Sandeel N.pout Dab Herring Gurnard Sole Whiting Plaice Haddock Saithe Cod Resource Total

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

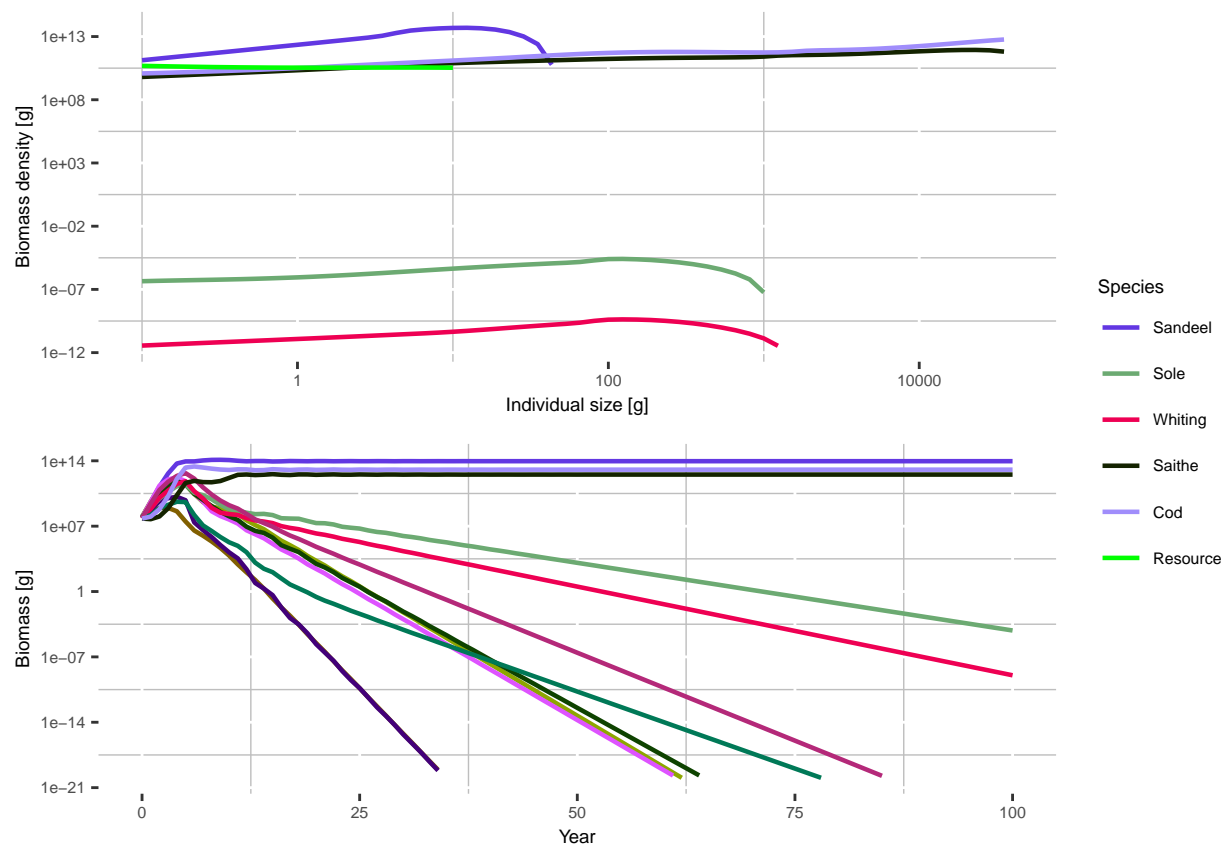
In this section you will:

- 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.

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

plotSummary(sim_uncalibrated, short = T)
```



JB - note error in plotSummary long =T - why?

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.

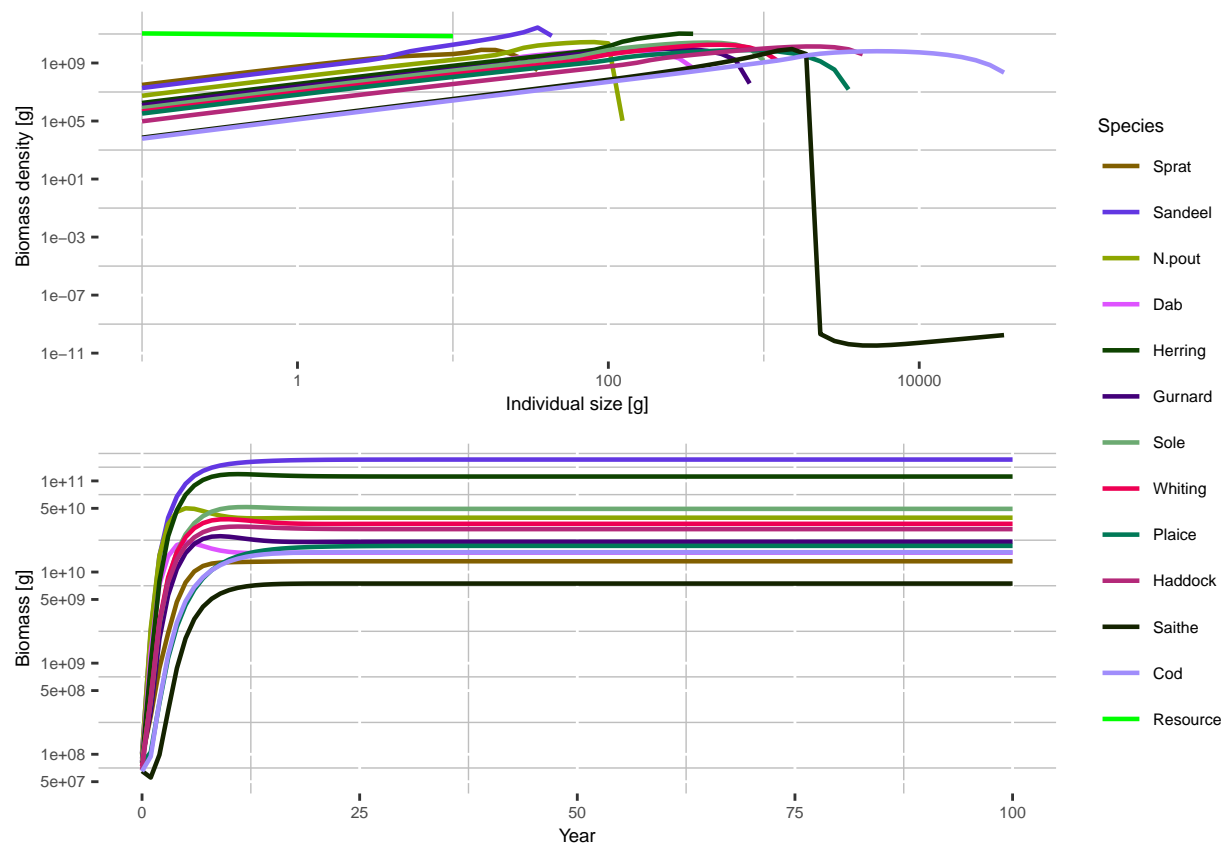
RF: should we put earlier a typical looking size spectrum?

To get coexistence one needs to guess reasonable R_{max} values which will stop out-competition from a few species. We assume that the density dependence is going to be positively 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_params$R_max
# and reduce erepro
params_guessed@species_params$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)
```



```
saveRDS(sim_guessed, "HTM0_sim.rds")
```

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

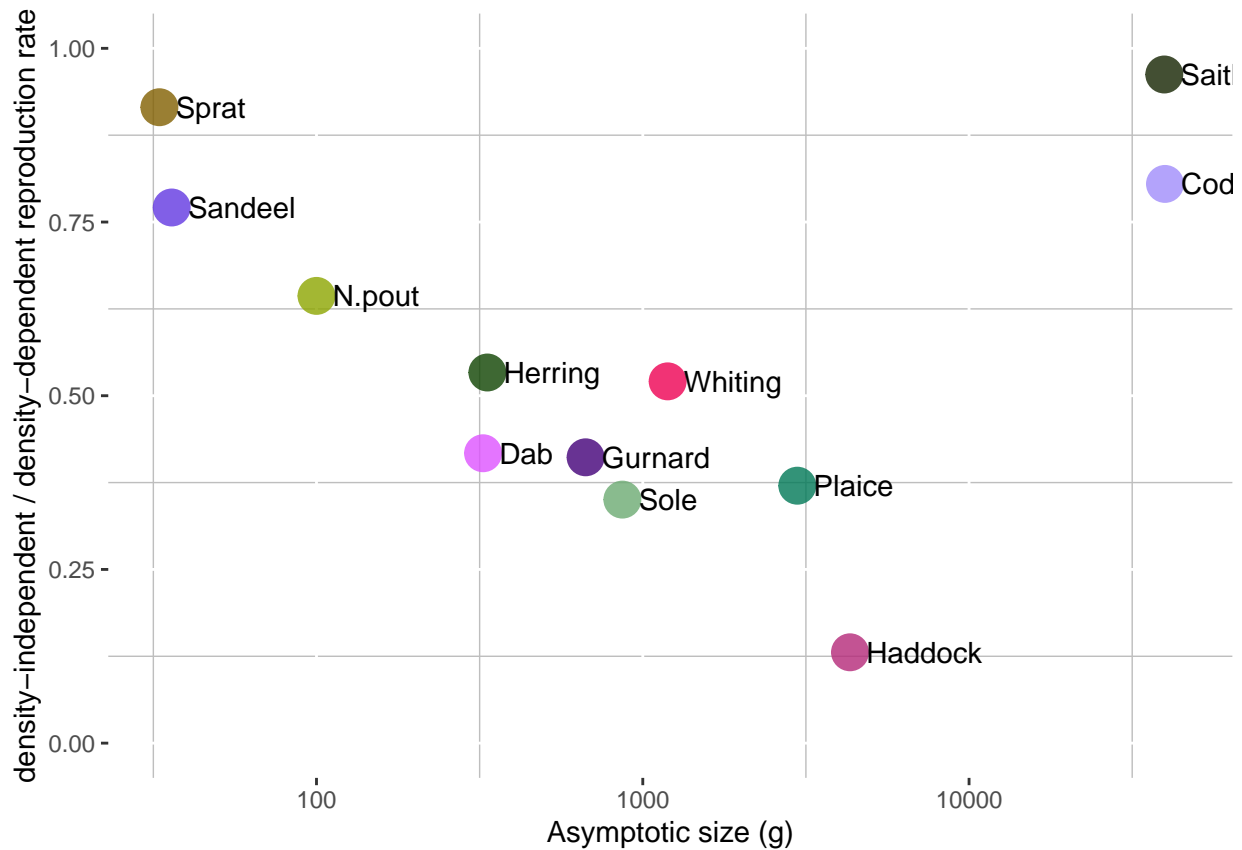
R_{max} affects the density-dependent reproduction rate (RDD) but limiting the maximum amount of spawn calculated from the density-independent reproduction rate (RDI). Let's look at the RDD/RDI ratio to see how strong acts R_{max} on our different species.

```

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_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 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.