

Mizer model

Beth Sleath

Loading library

Load the packages required to design mizer model and produce figures

```
library(mizer)
library(mizerExperimental)
library(tidyverse)
```

Setting initial parameters

```
params <- newSingleSpeciesParams()
params_initial <- setResource(params,
                              resource_dynamics = "resource_semichemostat")
params_double <- params_initial
params_double <- setResource(params_double,
                              resource_level = 0.1)
initialN(params_double) <- 2*initialN(params_double)
initialNResource(params_double) <- initialNResource(params_double)/2
```

Implementing fishing gear

```
gear_params(params_double) <- data.frame(
  gear = "gear",
  species = "Target species",
  catchability = 0.3,
  sel_func = "sigmoid_weight",
  sigmoidal_weight = 15,
  sigmoidal_sigma = 5)
```

```
params_double <- setFishing(params_double, gear_params = gear_params)
gear_params(params_double)
```

	gear	species	catchability	sel_func
Target species, gear	gear	Target species	0.3	sigmoid_weight
		sigmoidal_weight	sigmoidal_sigma	
Target species, gear		15		5

Simulating ecosystem behaviour

```
sim_double <- project(params_double, t_max = 15, effort = 1)
```

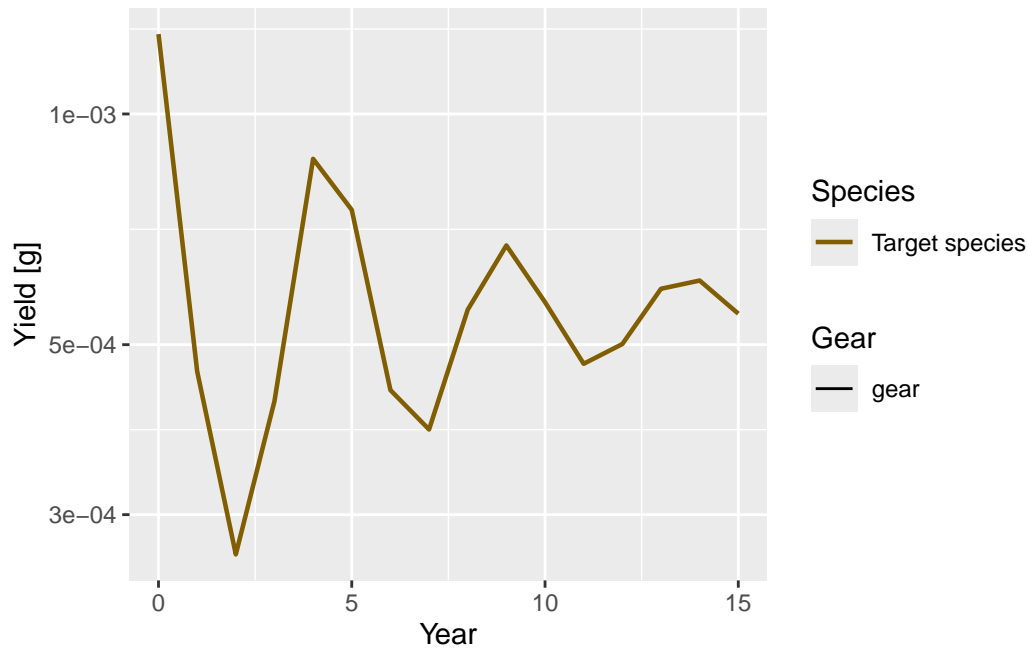
Extract and plot yield

```
getYieldGear(sim_double)
```

```
, , sp = Target species
```

	gear
time	gear
0	0.0012712274
1	0.0004614314
2	0.0002662361
3	0.0004220103
4	0.0008738575
5	0.0007496112
6	0.0004361817
7	0.0003875088
8	0.0005553817
9	0.0006736737
10	0.0005680980
11	0.0004720735
12	0.0005009610
13	0.0005914342
14	0.0006062247
15	0.0005487976

```
plotYieldGear(sim_double)
```



Locally reduce resource parameters

Calculate current flux

```
N <- finalN(sim_double)["Target species", , drop = TRUE]
w <- w(params_double)

E_growth <- getEGrowth(params_double)["Target species", , drop = TRUE]
gr <- w * E_growth
flux <- gr * N
```

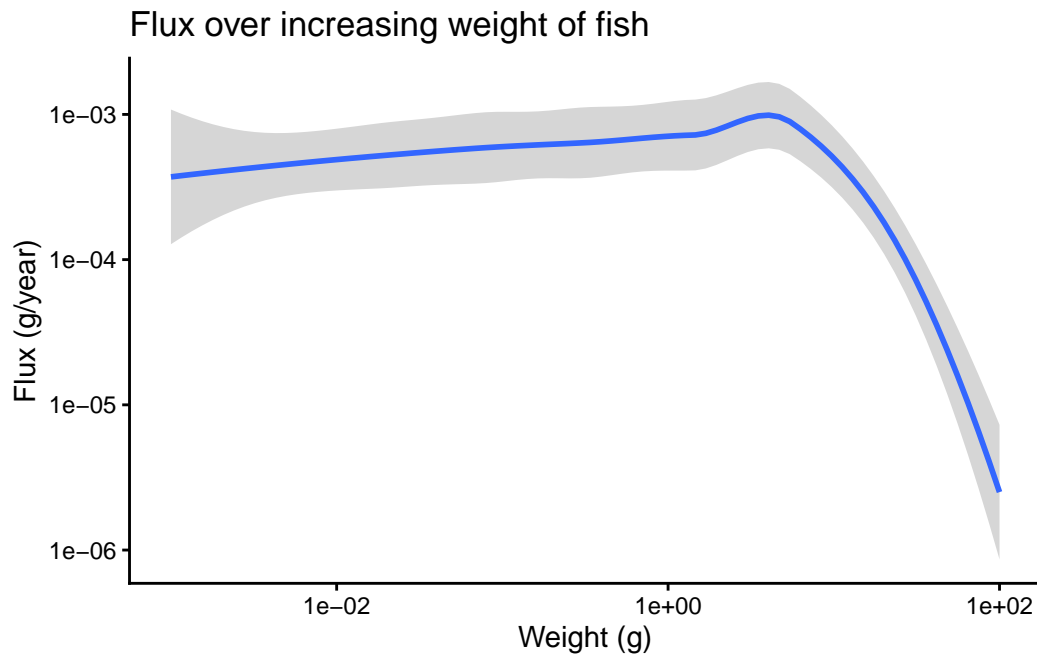
Plot initial flux without local reductions

```
initial_flux_data <- data.frame(Weight = w,
                                Flux = flux)
initial_flux_plot <- ggplot(initial_flux_data,
                             aes(x = Weight, y = Flux)) +
  geom_smooth() +
```

```

scale_x_log10() +
scale_y_log10() +
labs(
  x = paste0("Weight (g)"),
  y = paste0("Flux (g/year)"),
  title = "Flux over increasing weight of fish") +
theme_classic()
initial_flux_plot

```



Locally reduce resource rate

```

rr <- resource_rate(params_double)
w_full <- w_full(params_double)
w_full[215:240]

```

```

[1] 0.04466836 0.05011872 0.05623413 0.06309573 0.07079458 0.07943282
[7] 0.08912509 0.10000000 0.11220185 0.12589254 0.14125375 0.15848932
[13] 0.17782794 0.19952623 0.22387211 0.25118864 0.28183829 0.31622777
[19] 0.35481339 0.39810717 0.44668359 0.50118723 0.56234133 0.63095734
[25] 0.70794578 0.79432823

```

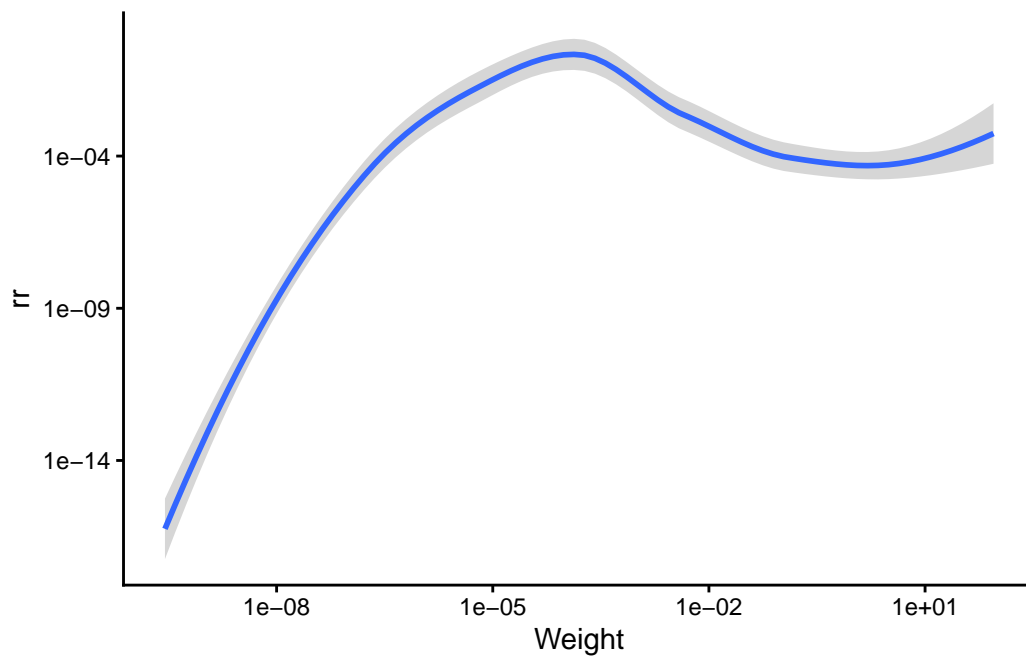
```

rr[215:240] <- rr[215:240] / 100000000

rr_data <- data.frame(
  Weight = params_double@w_full,
  rr = rr
)

rr_plot <- ggplot(rr_data,
  aes(x = Weight, y = rr)) +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  theme_classic()
rr_plot

```



Locally reduce resource capacity

```

rc <- resource_capacity(params_double)
w_full[215:240]

```

```
[1] 0.04466836 0.05011872 0.05623413 0.06309573 0.07079458 0.07943282
```

```

[7] 0.08912509 0.10000000 0.11220185 0.12589254 0.14125375 0.15848932
[13] 0.17782794 0.19952623 0.22387211 0.25118864 0.28183829 0.31622777
[19] 0.35481339 0.39810717 0.44668359 0.50118723 0.56234133 0.63095734
[25] 0.70794578 0.79432823

```

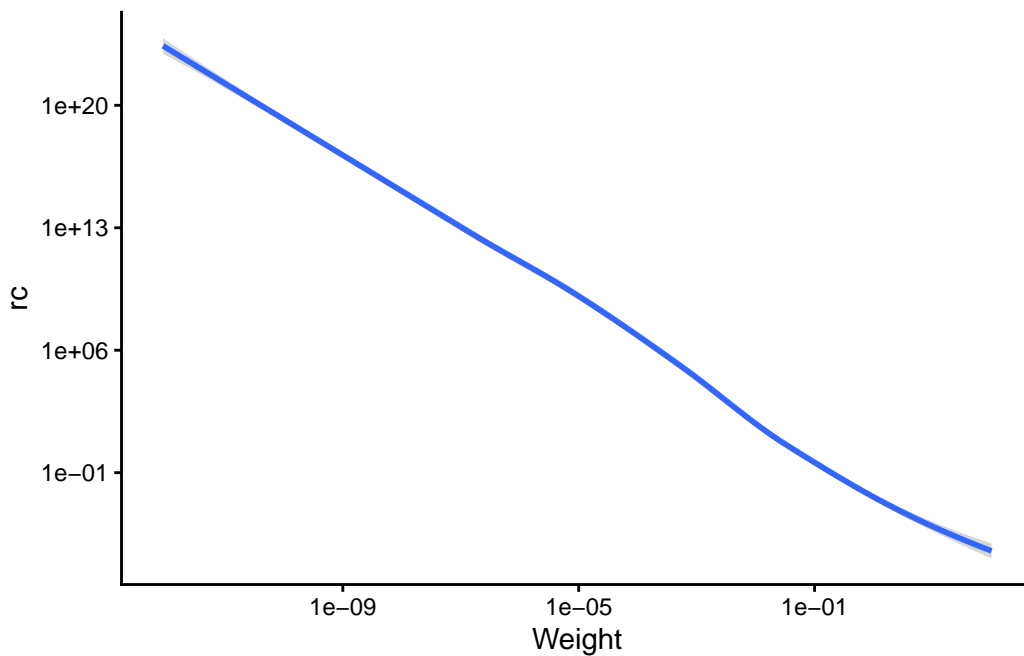
```

rc[215:240] <- rc[215:240] / 10000

rc_data <- data.frame(
  Weight = params_double@w_full,
  rc = rc
)

rc_plot <- ggplot(rc_data,
  aes(x = Weight, y = rc)) +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  theme_classic()
rc_plot

```



Set new parameters with local resource parameter reductions

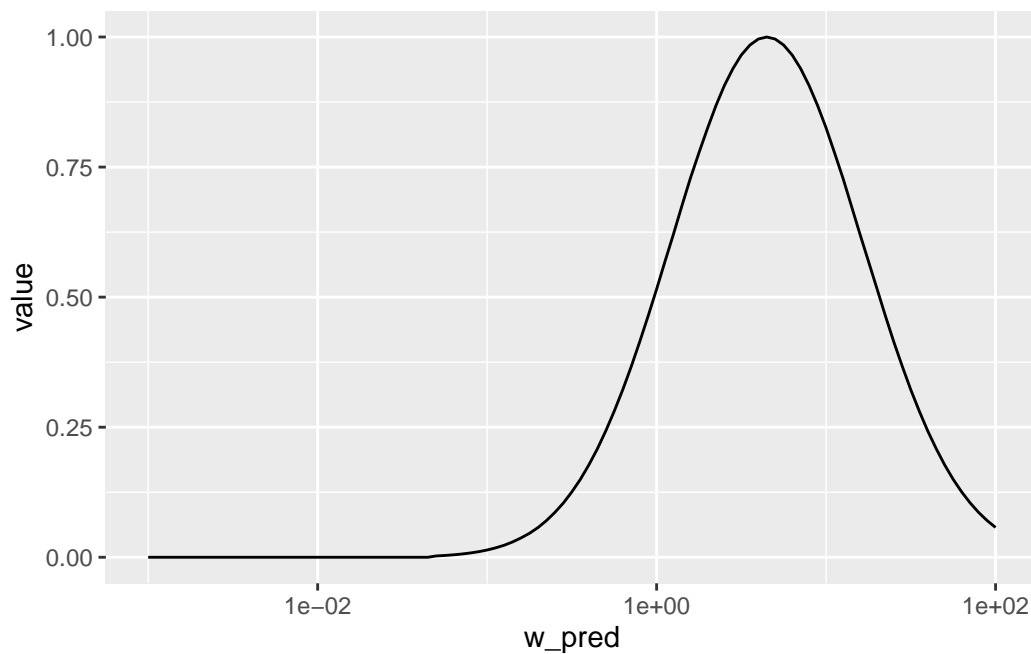
```
params_reduced <- setResource(params_double,
                              resource_capacity = rc,
                              resource_rate = rr,
                              balance = FALSE)
```

Narrow predation kernel

```
pred_kernel <- getPredKernel(params_reduced)

pred_kernel_reduced <- pred_kernel[, , 215, drop = FALSE]

ggplot(melt(pred_kernel_reduced)) +
  geom_line(aes(x = w_pred, y = value)) +
  scale_x_log10()
```



```
select(species_params(params_reduced), beta, sigma)
```

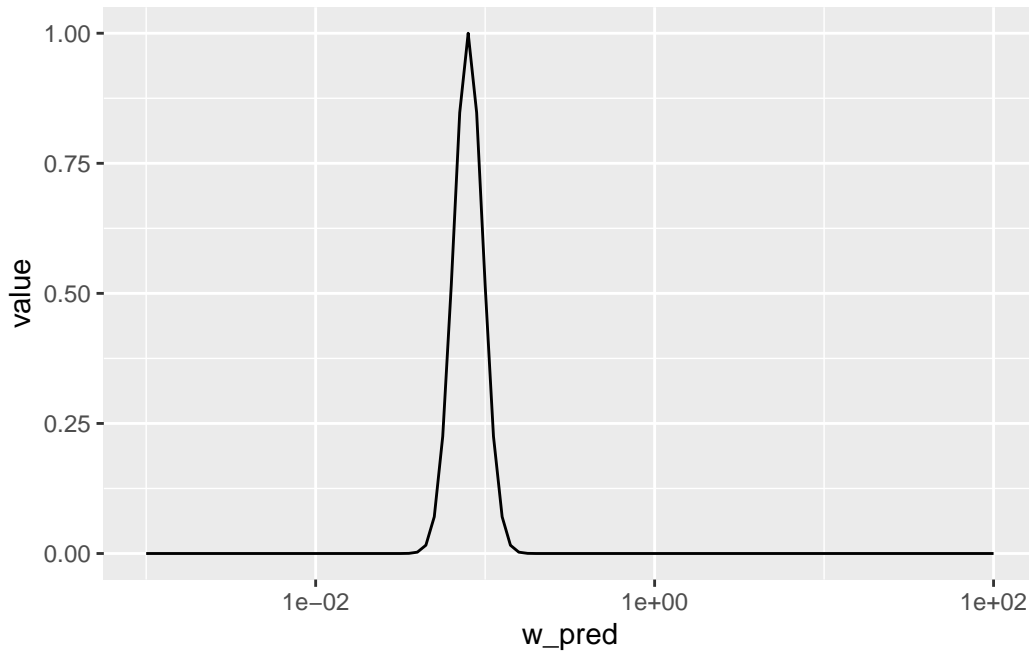
	beta	sigma
Target species	100	1.3

```

given_species_params(params_reduced)$sigma <- 0.2
given_species_params(params_reduced)$beta <- 100

getPredKernel(params_reduced)[, , 180, drop = FALSE] %>%
  melt() %>%
  ggplot() +
  geom_line(aes(x = w_pred, y = value)) +
  scale_x_log10()

```



Calculate flux after local reductions in resource parameters

```

sim_reduced <- project(params_reduced, t_max = 15, effort = 1)

N_reduced <- finalN(sim_reduced)["Target species", , drop = TRUE]
w <- w(params_reduced)

E_growth_reduced <- getEGrowth(params_reduced)["Target species", , drop = TRUE]
grr <- w * E_growth_reduced
flux_reduced <- grr * N_reduced

```



```

reduced_flux_data <- data.frame(Weight = w,
                                Flux = flux_reduced)

reduced_flux_plot <- ggplot(reduced_flux_data,
                            aes(x = Weight, y = Flux)) +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  labs(
    x = paste0("Weight (g)"),
    y = paste0("Flux (g/year)"),
    title = "Flux with locally reduced resource replenishment rate and resource capacity") +
  theme_classic()

reduced_flux_plot

```

