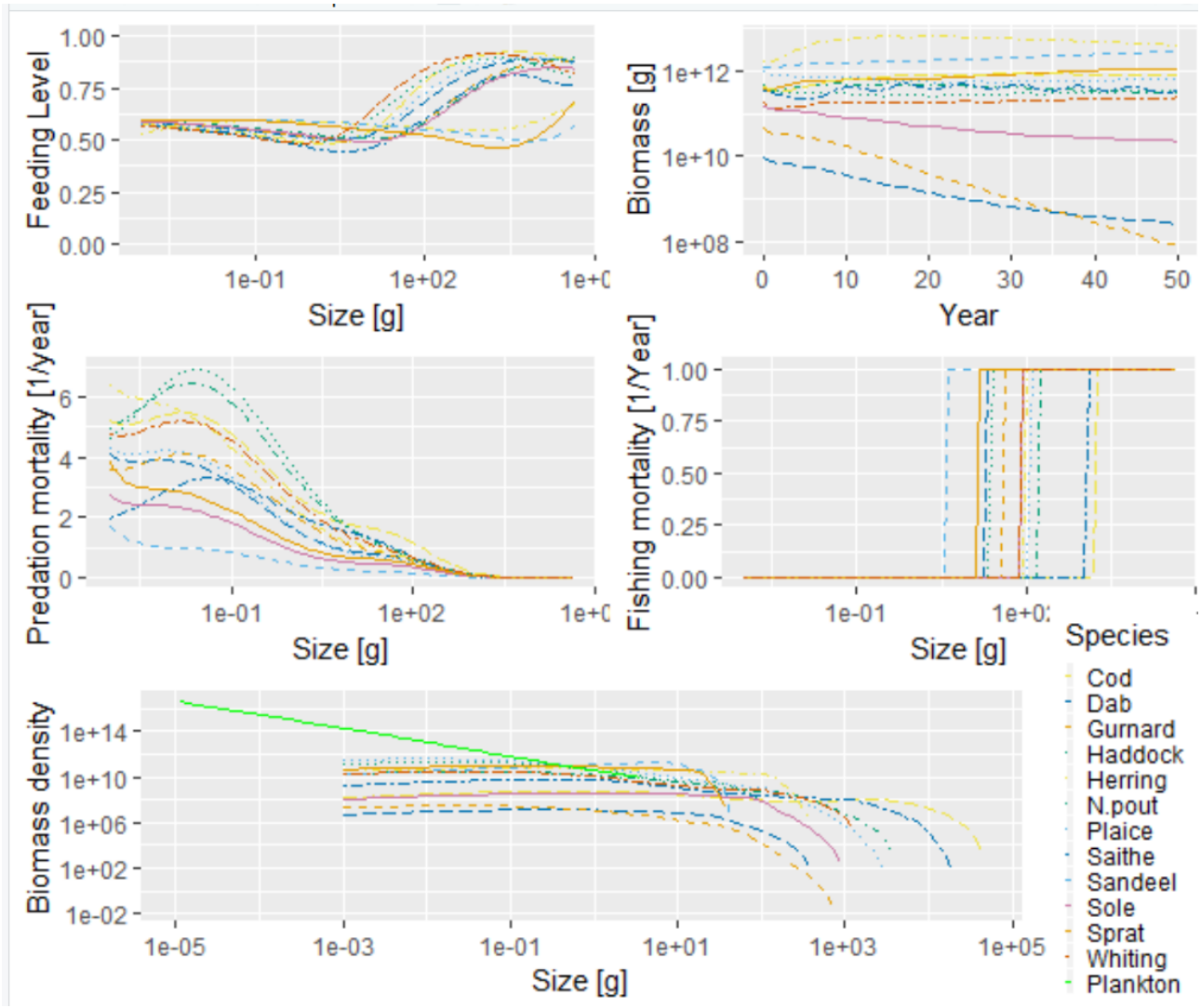
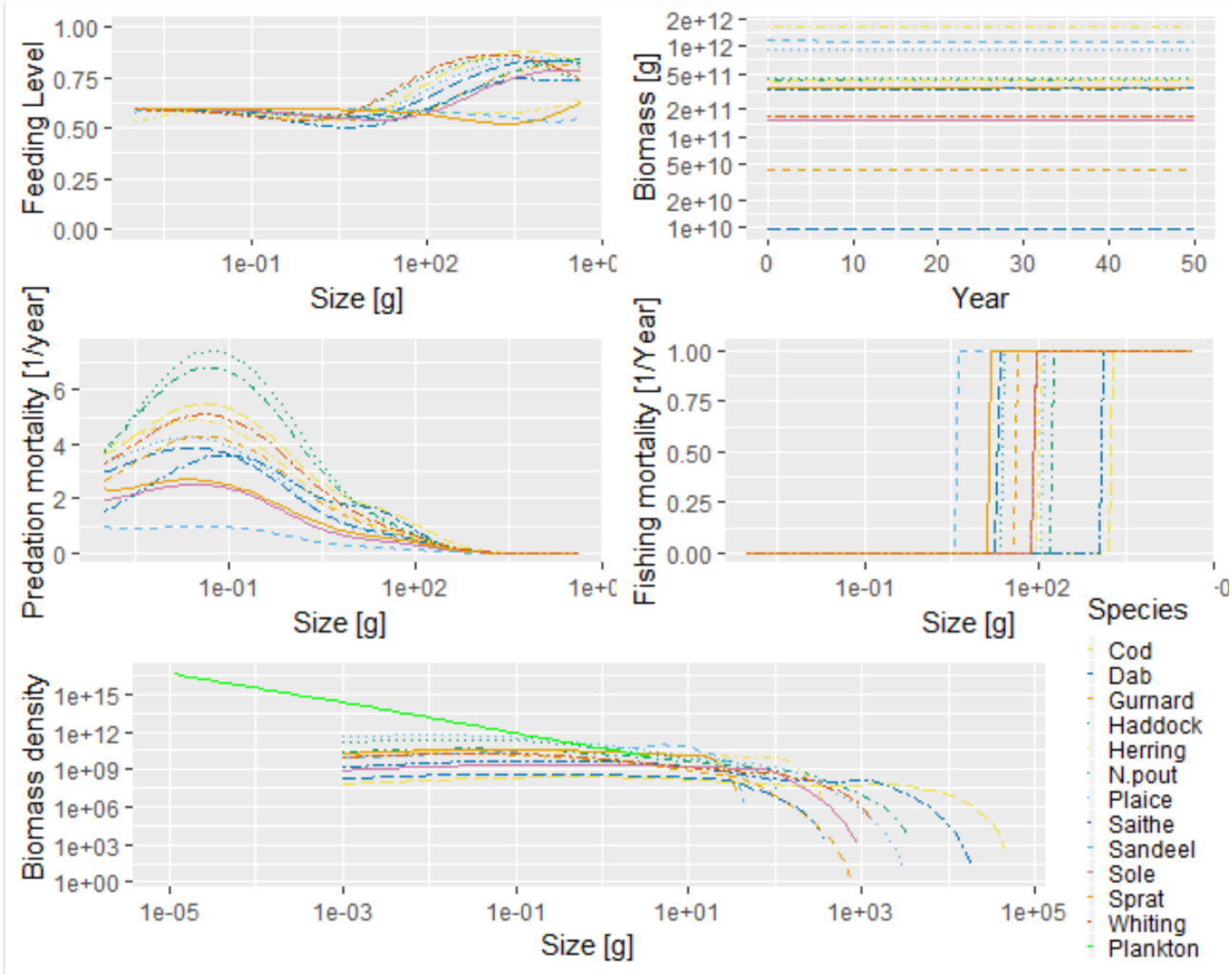
NS plankton Exp 1 Get near old steady state without r_max, then vary the value of r_pp and see what happens to stability.

[message: increasing r_pp from 10 to 15 seems better for co-existence than decreasing it in the adaptive changing equilibrium case]

r_pp = 5



r_pp = 10



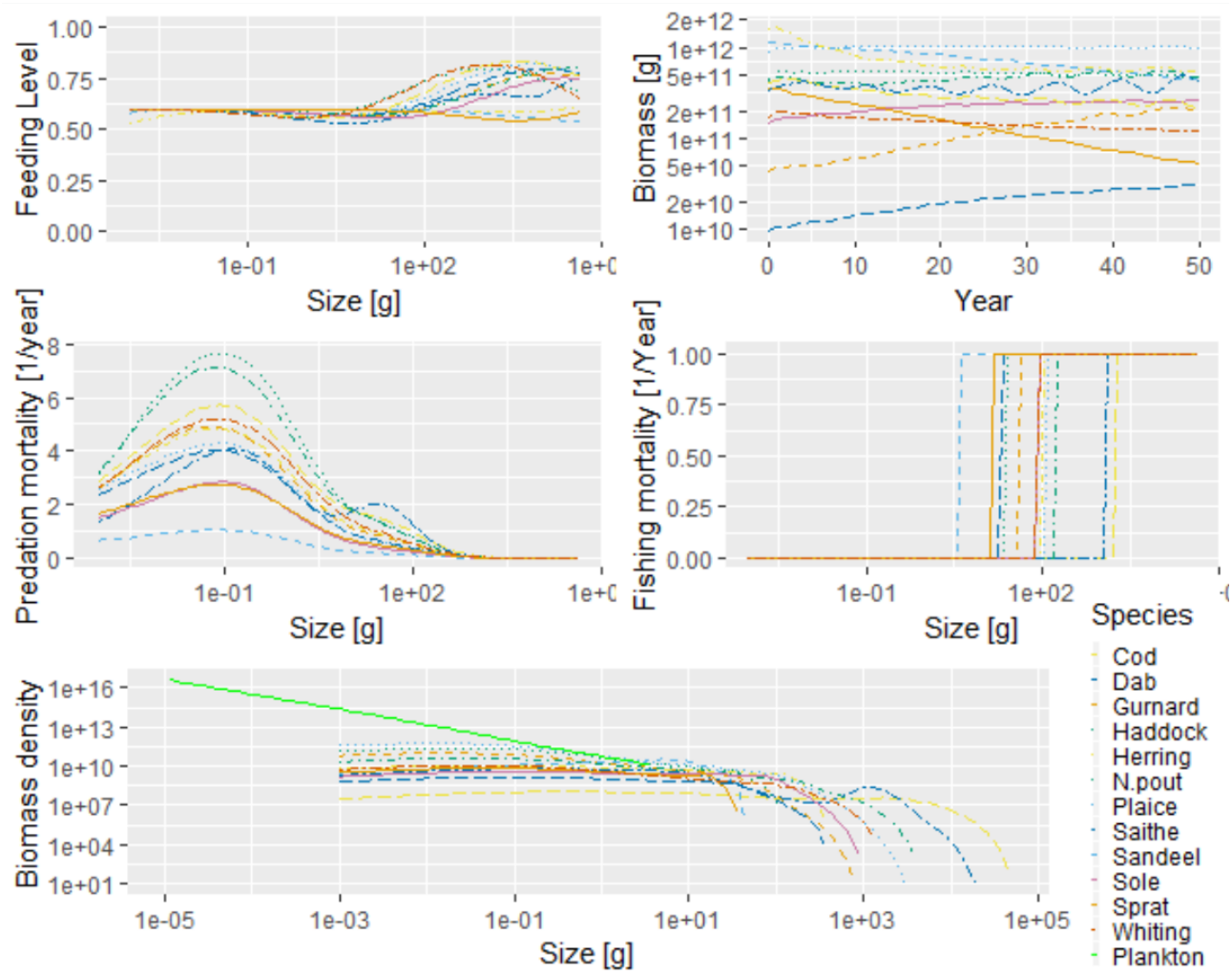
r_pp = 15 (below)

NsreplenishExp.R in scaling brach near

https://www.google.com/url?q=https://github.com/gustavdelius/mizer/commit/fe6ec8b8aea766d97886338b4ea19d8e401b3795&sa=D&source=hangouts&ust=1547721913409000&usg=AFQjCNFVVrt33qMP_tRLdqV7soUmvoNiDw

---

```
library(progress)
library(mizer)

############################
# load data
```




```
data(inter)
params_data_NS <- read.csv("./vignettes/NS_species_params.csv")
params <- MizerParams(params_data_NS, interaction = inter,r_pp=10)
# default r_pp = 10
# run to a steady state of the original system
sim <- project(params, effort = 1, t_max = 500, dt = 0.1, t_save = 1)
plot(sim)
# use steady state as new initial condition
params@initial_n <- sim@n[dim(sim@n)[1],,]
params@initial_n_pp <- sim@n_pp[dim(sim@n_pp)[1],]
# set erepro to compensate for turning off r_max
params@species_params$erepro <- params@species_params$erepro*

getRDD(params,params@initial_n,params@initial_n_pp)/getRDI(params,params@initial_n,params
@initial_n_pp)
#params@srr <- function(rdi, species_params) {
#    return(rdi)
#}

#params@rr_pp[] <-

#res@rr_pp[] <- r_pp * res@w_full^(n - 1)
params@species_params$r_max  <- Inf
```

```
########## run simulation without using r_max (note we get a better steady state if we repeat
# the above code with r_pp = 15)

sim <- project(params, effort = 1, t_max = 50, dt = 0.1, t_save = 1)
plot(sim)

######### run experiment again with higher r_pp

rr_pp2 <- 15
params2 <- params
params2@rr_pp[] <- rr_pp2*params2@w_full^(params2@n-1)
sim2 <- project(params2, effort = 1, t_max = 50, dt = 0.1, t_save = 1)
plot(sim2)

######### run experiment again with lower r_pp

rr_pp3 <- 5
params3 <- params
params3@rr_pp[] <- rr_pp3*params3@w_full^(params3@n-1)
sim3 <- project(params3, effort = 1, t_max = 50, dt = 0.1, t_save = 1)
plot(sim3)

# It seems decreasing r_pp to 5 hurts co-existence more than increasing r_pp from 10 to 15 does.

# wrote code that sets up NS steady state withour rmax, and investigates how
# altering the value of r_pp effects stability in a couple of instances.
```

NS plankton Exp 2 (reconstruct near steady states based on different r_pp values). Result: near steady state is only close because we just get close using steady (have to check for osc), and more instability/growth away from the steady state is .

[message: higher r_pp → more growth based instability -am I setting up the background resourse properly.

Nslocal.R in scaling branch near

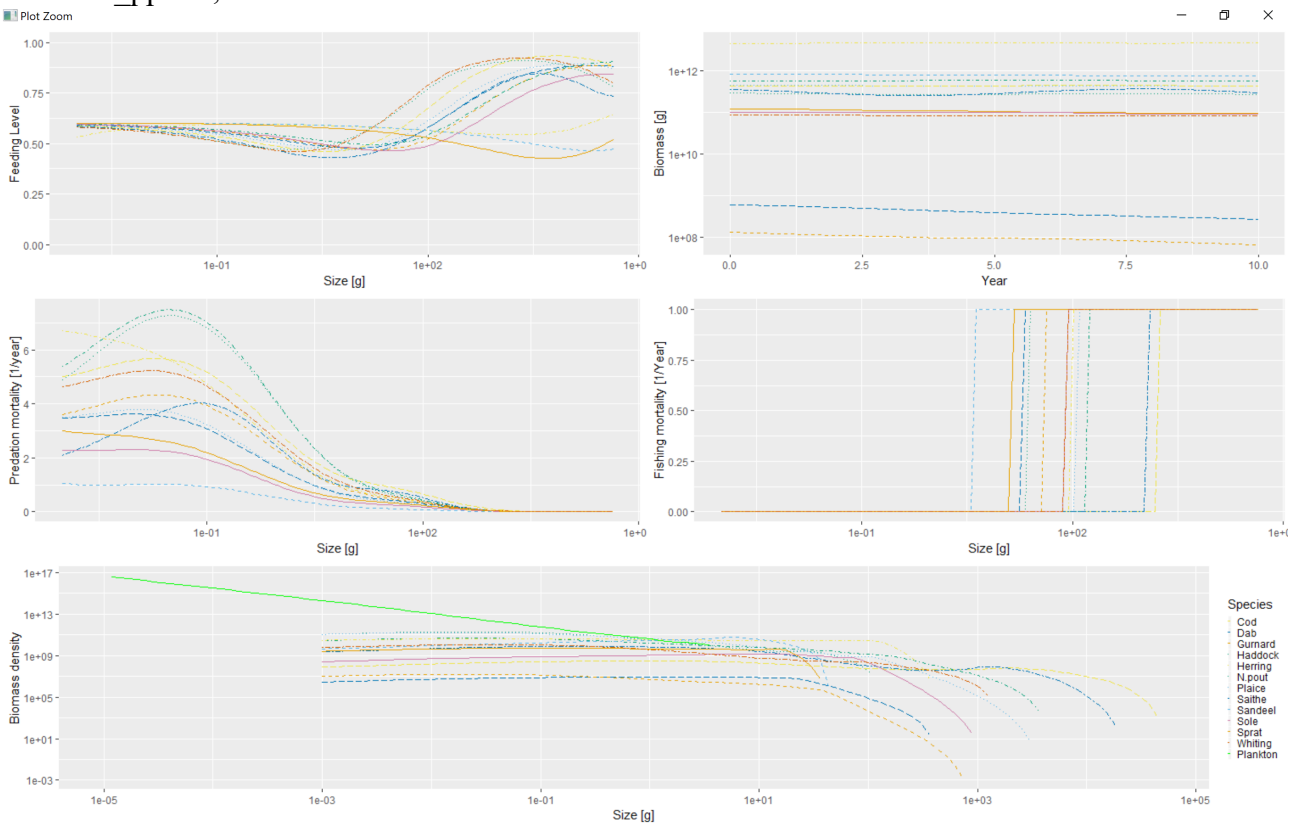
https://www.google.com/url?q=https://github.com/gustavdelius/mizer/commit/fe6ec8b8aea766d97886338b4ea19d8e401b3795&sa=D&source=hangouts&ust=1547721913409000&usg=AFQjCNFVVrt33qMP_tRLdqV7soUmvoNiDw

---

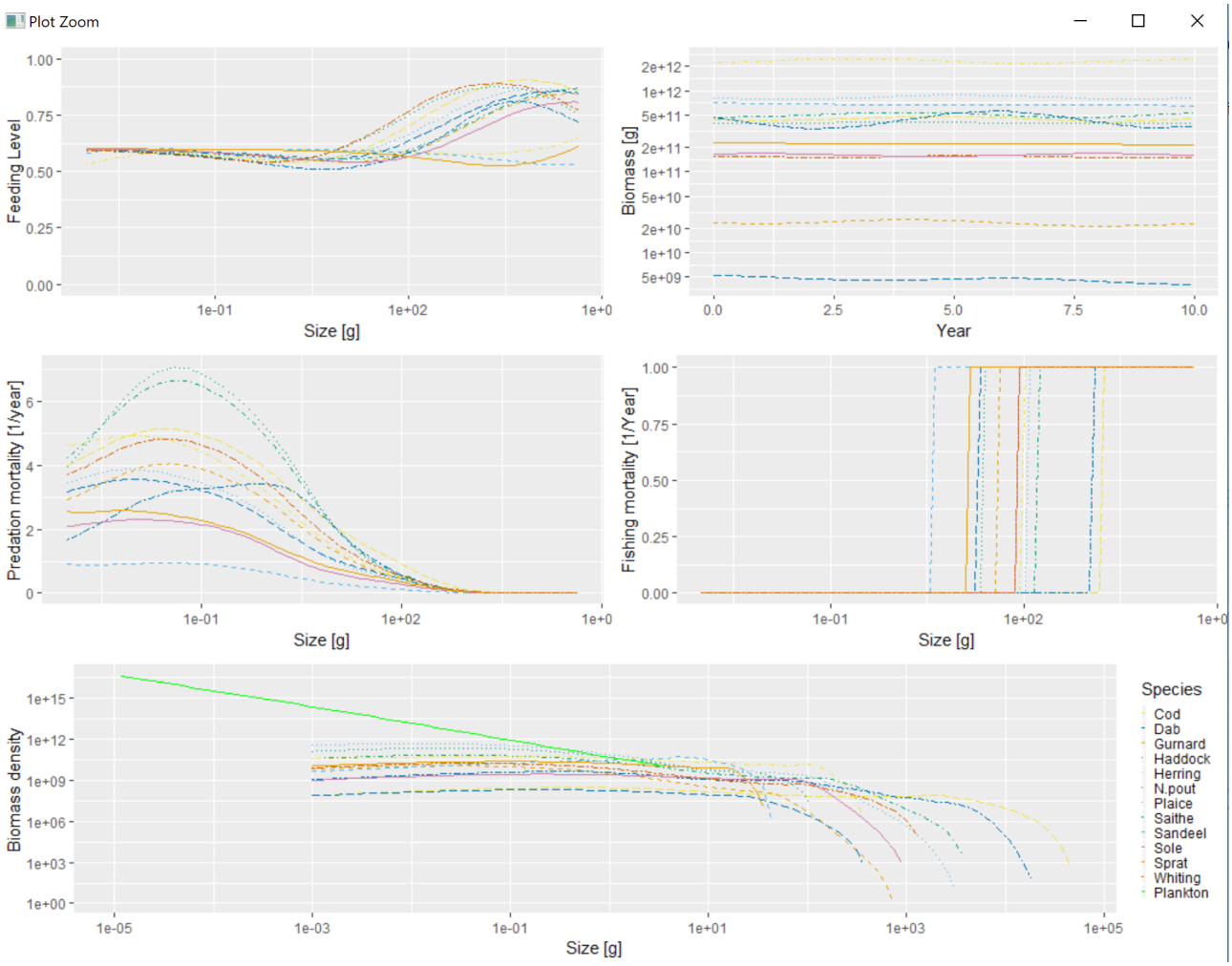
```
library(progress)
library(mizer)

############################
```

```
data(inter)
params_data_NS <- read.csv("./vignettes/NS_species_params.csv")
r_pp <- 10
params <- MizerParams(params_data_NS, interaction = inter, r_pp=r_pp)
sim <- project(params, effort = 1, t_max = 50, dt = 0.1, t_save = 1)
#plot(sim)
params@initial_n <- sim@n[dim(sim@n)[1],,]
params@initial_n_pp <- sim@n_pp[dim(sim@n_pp)[1],]
params@species_params$erepro <- params@species_params$erepro*

getRDD(params,params@initial_n,params@initial_n_pp)/getRDI(params,params@initial_n,params
@initial_n_pp)
#params@srr <- function(rdi, species_params) {
#    return(rdi)
#}
params@species_params$r_max  <- Inf
sim <- project(params, effort = 1, t_max = 50, dt = 0.1, t_save = 1)
plot(sim)
###################


params_data_NS_no_rmax <- read.csv("./vignettes/NS_species_params.csv")

params_data_NS_no_rmax$r_max[] <- Inf
params <- MizerParams(params_data_NS_no_rmax, interaction = inter, r_pp=r_pp)
params@initial_n <- sim@n[dim(sim@n)[1],,]
params@initial_n_pp <- sim@n_pp[dim(sim@n_pp)[1],]

#repro <- erepro *RDD/RDI

params@species_params$erepro <- params@species_params$erepro*

getRDD(params,params@initial_n,params@initial_n_pp)/getRDI(params,params@initial_n,params
@initial_n_pp)



#sim <- project(params, effort = 1, t_max = 10, dt = 0.1, t_save = 1)
#plot(sim)
params <- steady(params, effort = 1)

rdd_temp <- getRDD(params,params@initial_n,params@initial_n_pp)
gg <- getEGrowth(params,params@initial_n,params@initial_n_pp)

#params@species_params$w_min
#params@species_params$w_min[12]

# all w_min = w[1] in NS case, so below is simpler
rdd_desired <- gg[,1]*params@initial_n[,1]

# since r_max -> inf, rescaling the erepos is simpler
params@species_params$erepro <- params@species_params$erepro*rdd_desired/rdd_temp
```

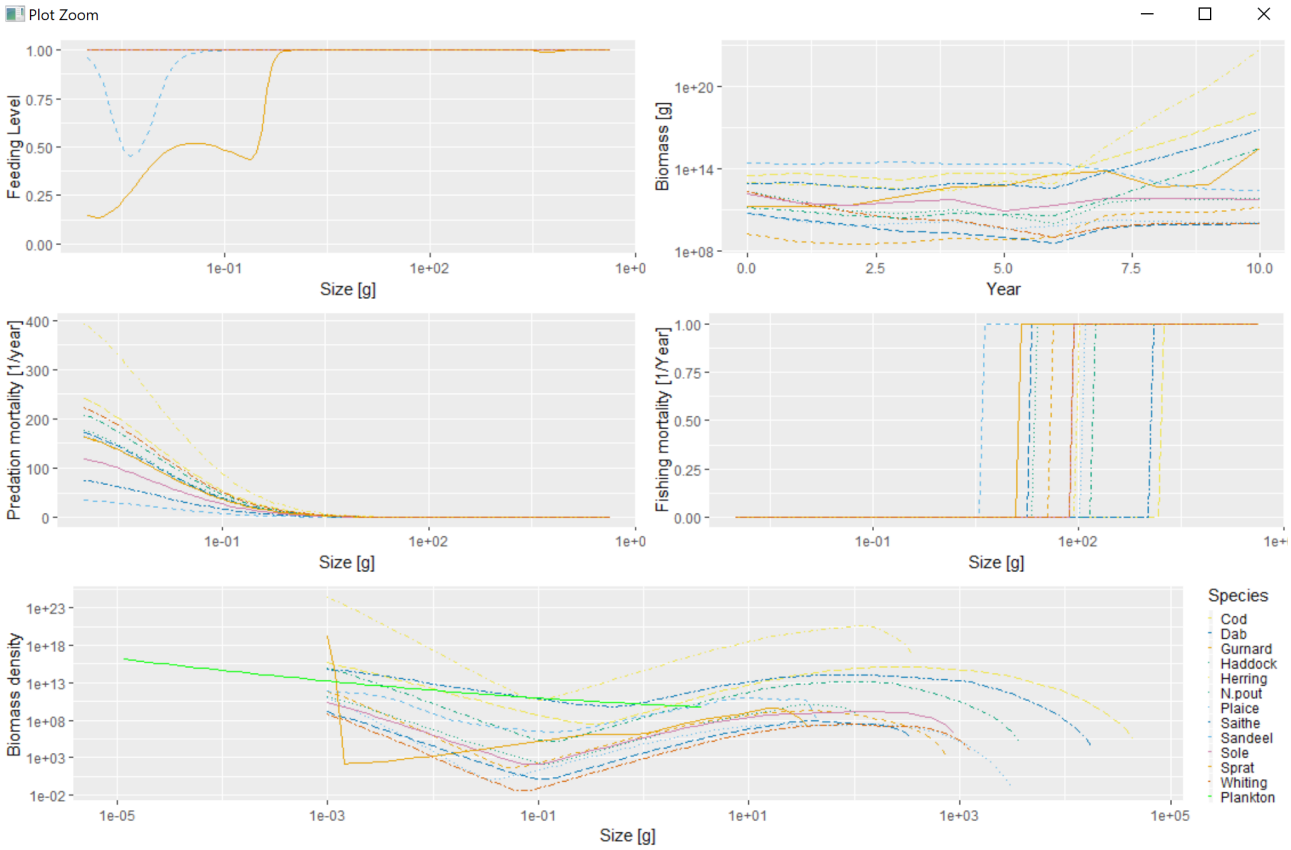
```
sim <- project(params, effort = 1, t_max = 10, dt = 0.1, t_save = 1)
plot(sim)

params@species_params$erepro
# got the north sea model from the vignette, and turned off the r_max, and used steady so run
# to a new steady state (when egg influx is held constant), and then the erepros are retunned
# so that we find a new steady state where no stock recruitmemnt relationship is imposed.

# got code that sets up a system with a given r_pp, runs it to steady state with repro off,
# and retunes erepro so the resulting system is steady

# when r_pp = 5, more stable but still oscillations
```




```
# When r_pp = 10 there are small oscillations
```



when r_pp = 15 less stable (small deviation promotes expansion)


Plot Zoom
Feeding Level
1.00
0.75
0.50
0.25
0.00
1e-01
1e+02
1e+0
Size [g]
Biomass [g]
1e+20
1e+14
1e+08
0.0
2.5
5.0
7.5
10.0
Year
Predation mortality [1/year]
400
300
200
100
0
1e-01
1e+02
1e+0
Size [g]
Fishing mortality [1/Year]
1.00
0.75
0.50
0.25
0.00
1e-01
1e+02
1e+
Size [g]
Biomass density
1e+23
1e+18
1e+13
1e+08
1e+03
1e-02
1e-05
1e-03
1e-01
1e+01
1e+03
1e+05
Size [g]
Species
Cod
Dab
Gurnard
Haddock
Herring
N.pout
Plaice
Saithe
Sandeel
Sole
Sprat
Whiting
Plankton

##