

Document

Sijia Wang

2025-05-22

```
library(conflicted)
conflict_prefer("map", "pomp")
conflict_prefer("filter", "dplyr")

library(pomp)
library(ggplot2)
library(tidyverse)
library(dplyr)
knitr::opts_chunk$set(echo = TRUE)
```

EVO Dynamics Plot

The following plots show the smoothed time series of algae, flagellates, and rotifers from experimental runs with evolutionary dynamics (EVO). These data and visualizations are based on:

Teppo Hiltunen, Stephen P. Ellner, Giles Hooker, Laura E. Jones, and Nelson G. Hairston Jr., Eco-Evolutionary Dynamics in a Three-Species Food Web with Intraguild Predation: Intriguingly Complex, *Advances in Ecological Research*, 2014.

The underlying data were extracted from the file "NoEvoData_SE_Feb14_2012.csv". The goal of this visualization is to examine whether the observed population dynamics are consistent with the presence of **canard cycles**, a phenomenon hypothesized in the final section of the paper to explain the complex oscillatory behavior in this predator-prey system.

```
setwd("/Users/sijiawang/Desktop/Honor Thesis")
require(locpol);

X123=read.csv("NoEvoData=SE=Feb14 2012.csv");
lagsr=lagsf=periods=numeric(0);
```

In all plots, green represents algae (A, prey), purple represents flagellates (F, intermediate predator), and red represents rotifers (R, top predator). The y-axis shows relative abundance, which reflects each species' abundance scaled by its maximum value in the run, allowing for direct comparison of dynamic patterns across species.

```
# Orange is panel (A)
X=cbind(X123$day,X123$Algae.orange1,X123$Flag.orange1,X123$Rot.orange1);
X=data.frame(X); colnames(X)<-c("day","algae","flagellates","rotifers");
X=X[X$day>=41,]; X=X[X$day<= 107,]; rownames(X)<-NULL;

adjust=1; npeaks=3;
```

```

#smoothed algae
acv=adjust*pluginBw(X$day,sqrt(X$algae),deg=3,kernel=gaussK);
fita=locpol(sqrt(algae)~day,data=X,deg=3,bw=acv,kernel=gaussK,xevalLen=1500);
pxa=fita$xeval; pya=fitted(fita)^2; maxa=max(pya);

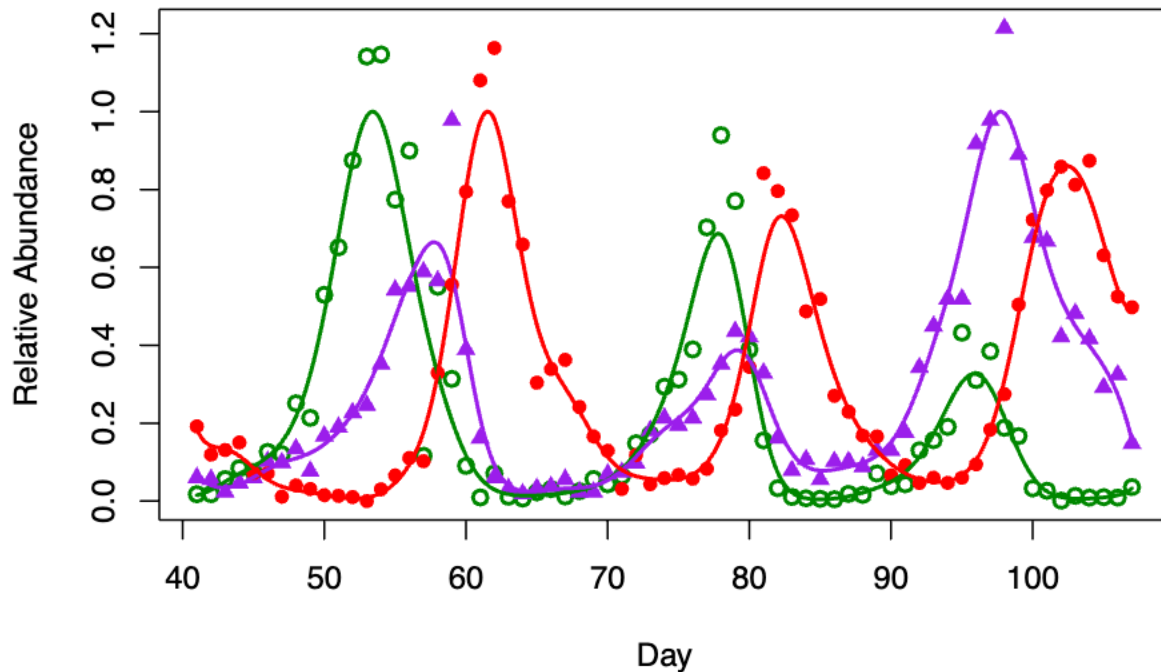
#smoothed flagellates
fcv=adjust*pluginBw(X$day,sqrt(X$flagellates),deg=3,kernel=gaussK);
fitf=locpol(sqrt(flagellates)~day,data=X,deg=3,bw=fcv,kernel=gaussK,xevalLen=1500);
pxf=fitf$xeval; pyf=fitted(fitf)^2; maxf=max(pyf);

#smoothed rotifers
rcv=adjust*pluginBw(X$day,sqrt(X$rotifers),deg=3,kernel=gaussK);
fitr=locpol(sqrt(rotifers)~day,data=X,deg=3,bw=rcv,kernel=gaussK,xevalLen=1500);
pxr=fitr$xeval; pyr=fitted(fitr)^2; maxr=max(pyr);

Xs=X; Xs[,2:4]=scale(Xs[,2:4],center=FALSE,scale=c(maxa,maxf,maxr));
matplot(Xs$day, Xs[,2:4], type="p", lty=1, col=c("green4", "purple", "red"),
        pch=c(1,17,16), lwd=2, ylab="Relative Abundance", xlab="Day", main="Orange (Panel 1)")
matpoints(pxa, cbind(pya/maxa, pyf/maxf, pyr/maxr), type="l", lty=1, lwd=2, col=c("green4", "purple",

```

Orange (Panel 1)



The **orange** panel corresponds to experimental run 1 with algae, flagellates, and rotifers sampled from days 41 to 107.

```

# Brown
X=cbind(X123$day,X123$Algae.brown2,X123$Flag.brown2,X123$Rot.brown2);
X=data.frame(X); colnames(X)<-c("day","algae","flagellates","rotifers");

```

```

X=X[X$day>=24,]; X=X[X$day<=90,]; rownames(X)<-NULL;

adjust=1; npeaks=3;
#smoothed algae
acv=adjust*pluginBw(X$day,sqrt(X$algae),deg=3,kernel=gaussK);
fita=locpol(sqrt(algae)~day,data=X,deg=3,bw=acv,kernel=gaussK,xevalLen=1500);
pxa=fita$xeval; pya=fitted(fita)^2; maxa=max(pya);

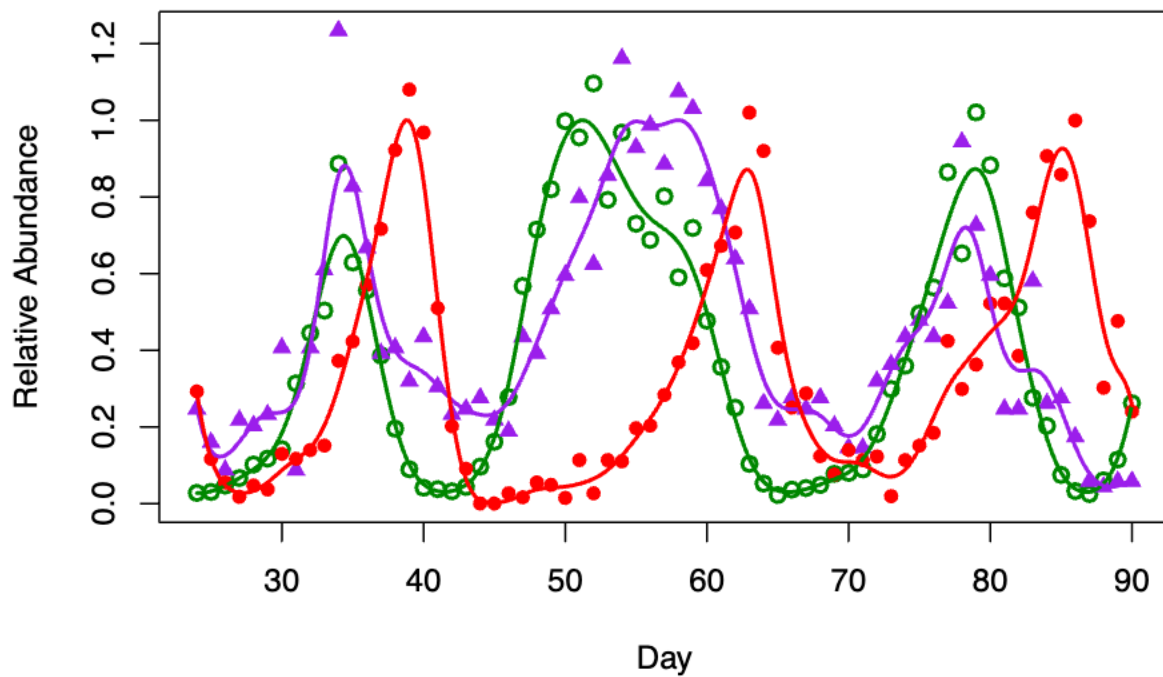
#smoothed flagellates
fcv=adjust*pluginBw(X$day,sqrt(X$flagellates),deg=3,kernel=gaussK);
fitf=locpol(sqrt(flagellates)~day,data=X,deg=3,bw=fcv,kernel=gaussK,xevalLen=1500);
pxf=fitf$xeval; pyf=fitted(fitf)^2; maxf=max(pyf);

#smoothed rotifers
rcv=adjust*pluginBw(X$day,sqrt(X$rotifers),deg=3,kernel=gaussK);
fitr=locpol(sqrt(rotifers)~day,data=X,deg=3,bw=rcv,kernel=gaussK,xevalLen=1500);
pxr=fitr$xeval; pyr=fitted(fitr)^2; maxr=max(pyr);

Xs=X; Xs[,2:4]=scale(Xs[,2:4],center=FALSE,scale=c(maxa,maxf,maxr));
matplot(Xs$day,Xs[,2:4],type="p",lty=1,col=c("green4","purple","red"),
        pch=c(1,17,16),lwd=2,ylab="Relative Abundance",xlab="Day",main="Brown (Panel 2)");
matpoints(pxa,cbind(pya/maxa,pyf/maxf,pyr/maxr),type="l",lty=1,lwd=2,col=c("green4","purple","red"));

```

Brown (Panel 2)



The **brown** panel shows data from run 2, covering days 24 to 90.

```

# Pink
X=cbind(X123$day,X123$Algae.pink2,X123$Flag.pink2,X123$Rot.pink2);
X=data.frame(X); colnames(X)<-c("day","algae","flagellates","rotifers");
X=X[X$day>=29,]; X=X[X$day<=69,]; rownames(X)<-NULL;

adjust=1; npeaks=3;
#smoothed algae
acv=adjust*pluginBw(X$day,sqrt(X$algae),deg=3,kernel=gaussK);
fita=locpol(sqrt(algae)~day,data=X,deg=3,bw=acv,kernel=gaussK,xevalLen=1500);
pxa=fita$xeval; pya=fitted(fita)^2; maxa=max(pya);

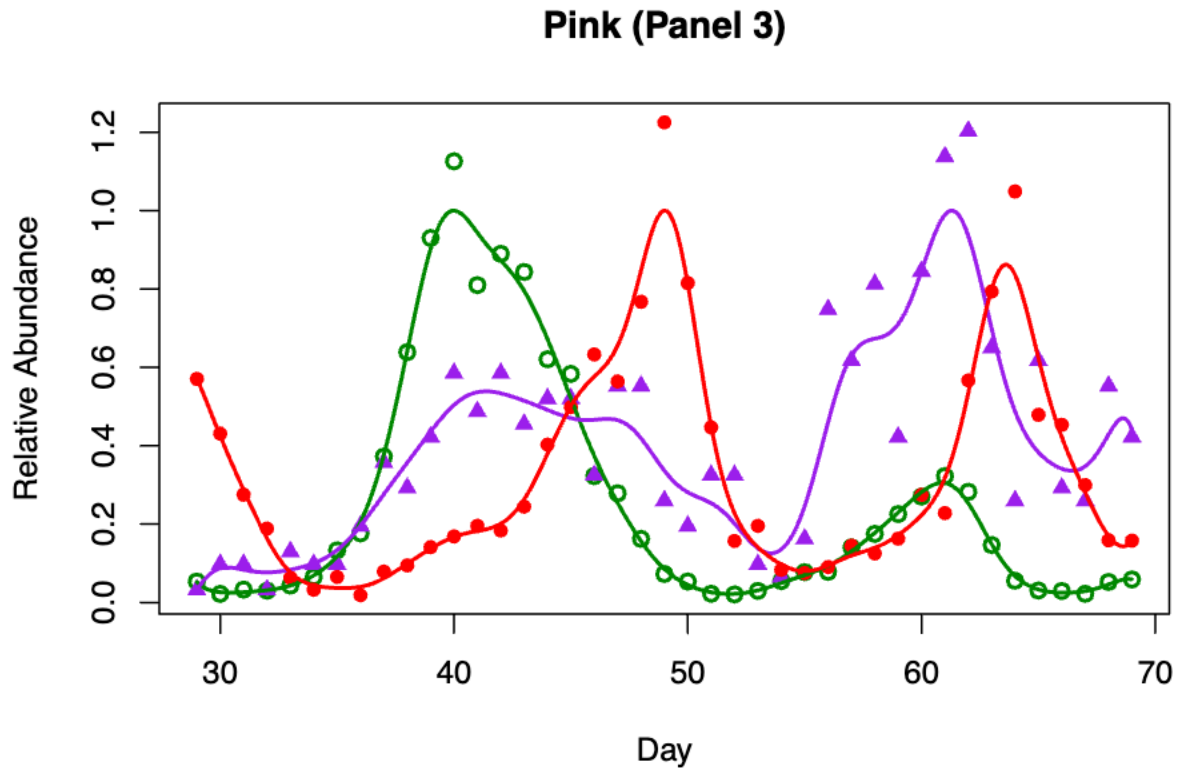
#smoothed flagellates
fcv=adjust*pluginBw(X$day,sqrt(X$flagellates),deg=3,kernel=gaussK);
fitf=locpol(sqrt(flagellates)~day,data=X,deg=3,bw=fcv,kernel=gaussK,xevalLen=1500);
pxf=fitf$xeval; pyf=fitted(fitf)^2; maxf=max(pyf);

#smoothed rotifers
rcv=adjust*pluginBw(X$day,sqrt(X$rotifers),deg=3,kernel=gaussK);
fitr=locpol(sqrt(rotifers)~day,data=X,deg=3,bw=rcv,kernel=gaussK,xevalLen=1500);
pxr=fitr$xeval; pyr=fitted(fitr)^2; maxr=max(pyr);

Xs=X; Xs[,2:4]=scale(Xs[,2:4],center=FALSE,scale=c(maxa,maxf,maxr));

matplot(Xs$day,Xs[,2:4],type="p",lty=1,col=c("green4","purple","red"),
        pch=c(1,17,16),lwd=2,ylab="Relative Abundance",xlab="Day",main="Pink (Panel 3)");
matpoints(pxa,cbind(pya/maxa,pyf/maxf,pyr/maxr),type="l",lty=1,lwd=2,col=c("green4","purple","red"));

```



The **pink** panel represents run 3, with observations taken between days 29 and 69.

POMP Model Fitting

We now use the **orange** panel data between days 41 and 107 to build and analyze a Partially Observed Markov Process (POMP) model.

```
X123=read.csv("NoEvoData=SE=Feb14 2012.csv");
orange_data <- data.frame(
  day = X123$day,
  algae = X123$Algae.orange1,
  flagellates = X123$Flag.orange1,
  rotifers = X123$Rot.orange1
)

orange_data <- orange_data[orange_data$day >= 41 & orange_data$day <= 107, ]
rownames(orange_data) <- NULL

summary(orange_data)
```

##	day	algae	flagellates	rotifers
##	Min. : 41.0	Min. : 1111	Min. : 51.0	Min. : 0.0000
##	1st Qu.: 57.5	1st Qu.: 19167	1st Qu.: 214.1	1st Qu.: 0.5294
##	Median : 74.0	Median : 79444	Median : 497.2	Median : 1.3514
##	Mean : 74.0	Mean : 211156	Mean : 786.1	Mean : 2.4763

```
## 3rd Qu.: 90.5    3rd Qu.: 293333    3rd Qu.:1160.2    3rd Qu.:4.1135
## Max.      :107.0    Max.      :1080000    Max.      :3340.5    Max.      :9.3529
```

```
xrange <- range(orange_data$day)
yrange <- range(
  orange_data$algae,
  orange_data$flagellates,
  orange_data$rotifers,
  na.rm = TRUE
)

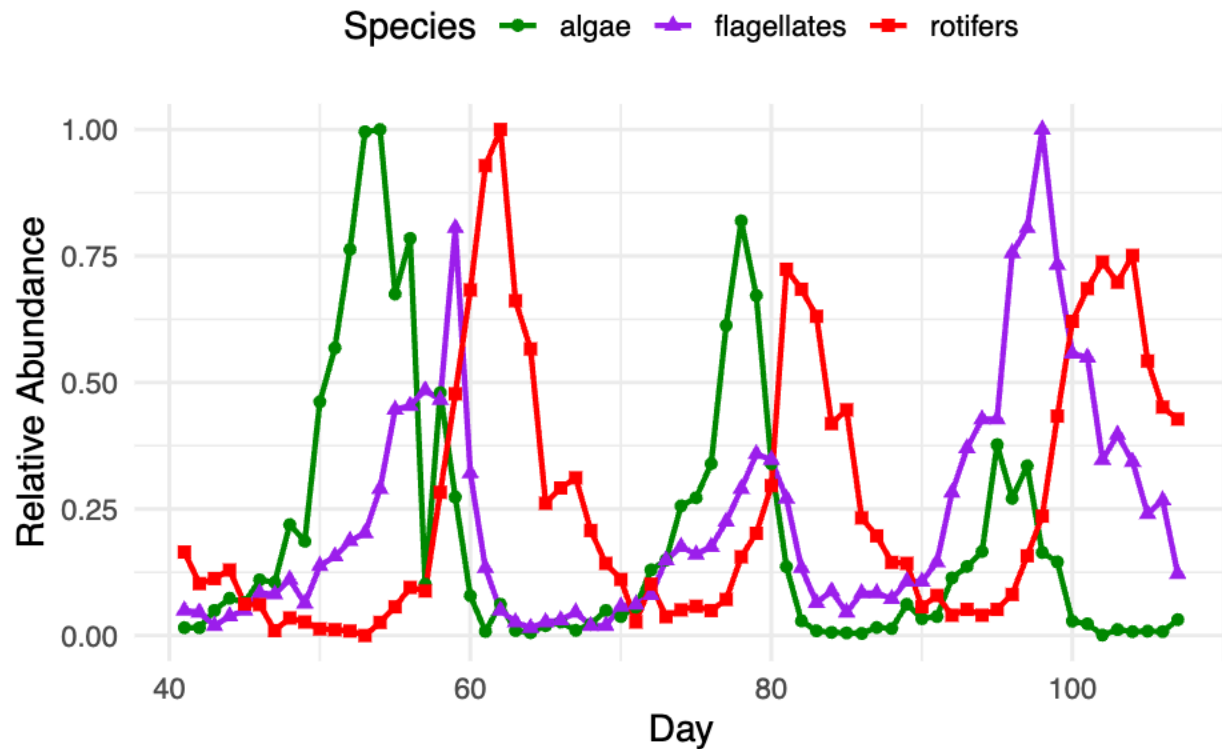
max_algae <- max(orange_data$algae, na.rm = TRUE)
max_flag <- max(orange_data$flagellates, na.rm = TRUE)
max_roti <- max(orange_data$rotifers, na.rm = TRUE)

orange_rel <- orange_data
orange_rel[, 2:4] <- scale(orange_data[, 2:4], center = FALSE,
                          scale = c(max_algae, max_flag, max_roti))

orange_rel_long <- orange_rel %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "rel_abundance")

ggplot(orange_rel_long, aes(x = day, y = rel_abundance, color = species)) +
  geom_line(linewidth = 1) +
  geom_point(size = 2, aes(shape = species)) +
  scale_color_manual(values = c(
    algae = "green4",
    flagellates = "purple",
    rotifers = "red"
  )) +
  scale_shape_manual(values = c(
    algae = 16, flagellates = 17, rotifers = 15
  )) +
  labs(
    title = "Standardized Population Dynamics of Orange Panel",
    x = "Day", y = "Relative Abundance",
    color = "Species", shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

Standardized Population Dynamics of Orange Panel



```

statenames <- c("S", # limiting substrate
               "A", # algae
               "R", # rotifers
               "F") # flagellates

maxA <- max(orange_data$algae, na.rm = TRUE)
maxF <- max(orange_data$flagellates, na.rm = TRUE)
maxR <- max(orange_data$rotifers, na.rm = TRUE)

# all parameters are positive
paramnames <- c(
  "delta", # dilution rate
  "kA", # half-saturation constant
  "kR", # half-saturation constant
  "r",
  "g",
  "h",
  "alphaA", # handling-time parameter
  "alphaF", # handling-time parameter
  "eta", # maximum per capitarate parameter for flagellates grazing on algae
  "IF",
  "S_0",
  "A_0",
  "R_0",
  "F_0",

```



```

"maxA",
"maxF",
"maxR"
)

```

```

rproc <- euler(
  step.fun = Csnippet("
    double dS = delta * (1 - S) - S * r * A / (kA + S);
    double dA = A * (
      r * S / (kA + S)
      - g * R / (kR + A + alphaF * F)
      - h * F / (1 + alphaA * A)
      - delta
    );
    double dR = R * (
      g * A / (kR + A + alphaF * F)
      + eta * F / (kR + A + alphaF * F)
      - delta
    );
    double dF = F * (
      h * A / (1 + alphaA * A)
      - eta * R / (kR + A + alphaF * F)
      - delta
    ) + IF;

    S += dS * dt;
    A += dA * dt;
    R += dR * dt;
    F += dF * dt;

    if (A < 0) A = 1e-6;
    if (F < 0) F = 1e-6;
    if (R < 0) R = 1e-6;
    if (S < 0) S = 1e-6;

    if (A > 1e3) A = 1e3;
    if (F > 1e3) F = 1e3;
    if (R > 1e3) R = 1e3;

  "),
  delta.t = 0.3
)

```

```

rmeasure <- Csnippet("
  algae = rnorm(A / maxA, 0.1);
  flagellates = rnorm(F / maxF, 0.1);
  rotifers = rnorm(R / maxR, 0.1);
")

dmeasure <- Csnippet("
  lik = dnorm(algae, A / maxA, 0.1, 1) +
        dnorm(flagellates, F / maxF, 0.1, 1) +
        dnorm(rotifers, R / maxR, 0.1, 1);

```



```

    if (!give_log) lik = exp(lik);
  })

```

```

rinit <- Csnippet("
  S = S_0;
  A = A_0;
  R = R_0;
  F = F_0;
")

```

```

pt <- parameter_trans(
  log = c(
    "delta", # dilution rate
    "kA", # half-saturation constant
    "kR", # half-saturation constant
    "r",
    "g",
    "h",
    "alphaA", # handling-time parameter
    "alphaF", # handling-time parameter
    "eta", # maximum per capitarate parameter for flagellates grazing on algae
    "IF",
    "S_0",
    "A_0",
    "R_0",
    "F_0",
    "maxA",
    "maxF",
    "maxR")
)

```

```

pomp_model <- pomp(
  data = orange_rel,
  times = "day",
  t0 = 40,
  rprocess = rproc,
  rmeasure = rmeasure,
  dmeasure = dmeasure,
  rinit = rinit,
  statenames = statenames,
  paramnames = paramnames,
  obsnames = c("algae", "flagellates", "rotifers"),
  partrans = pt
)

```

```

init_row <- X123[X123$day == 40, ]

```

```

params <- c(
  delta = 0.15,
  kA = 0.15,
  kR = 0.25,
  r = 0.7,
  g = 0.7,

```

```

h = 7,
alphaA = 2,
alphaF = 1,
eta = 1.5,
IF = 0.001,
S_0 = 1000,
A_0 = init_row$Algae.orange1 / max(orange_data$algae, na.rm = TRUE),
R_0 = init_row$Rot.orange1 / max(orange_data$rotifers, na.rm = TRUE),
F_0 = init_row$Flag.orange1 / max(orange_data$flagellates, na.rm = TRUE),
maxA = max(orange_data$algae, na.rm = TRUE),
maxR = max(orange_data$rotifers, na.rm = TRUE),
maxF = max(orange_data$flagellates, na.rm = TRUE)
)

sim <- simulate(pomp_model, params = params, nsim = 1)
sim_df <- as.data.frame(sim, include.data = FALSE)

orange_obs_long <- orange_rel %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "obs")

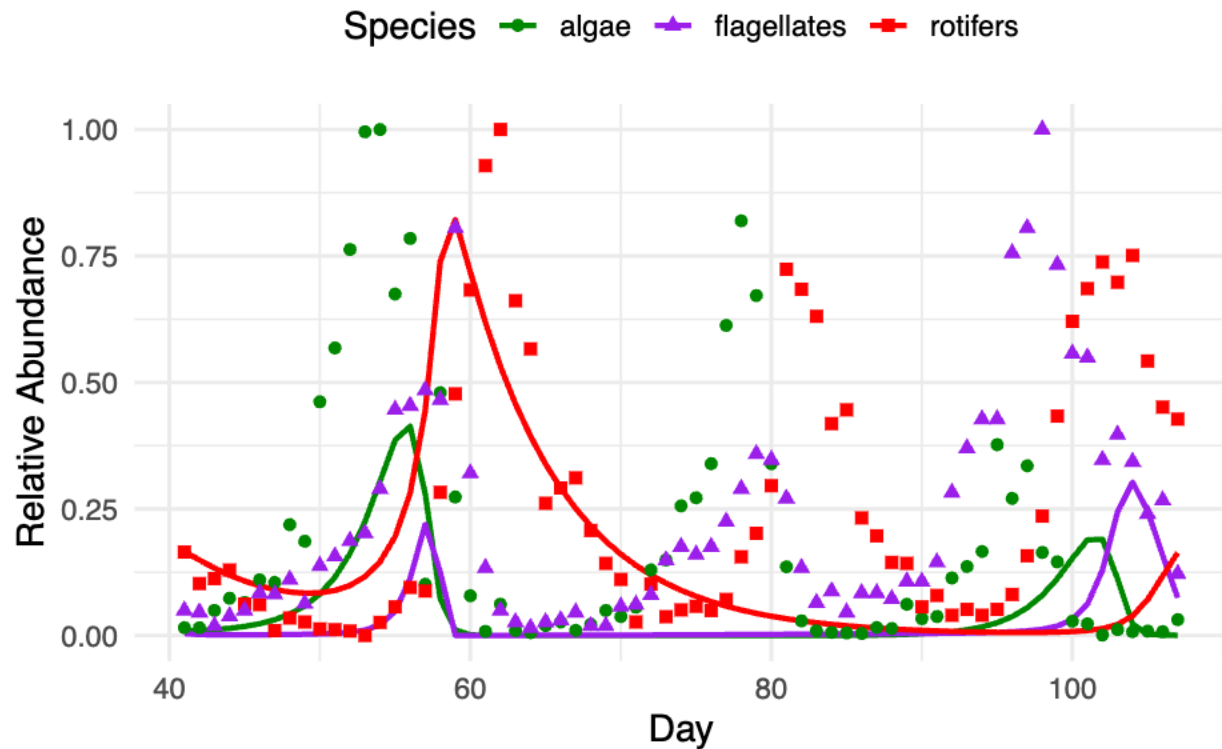
sim_long <- sim_df %>%
  mutate(time = day) %>%
  pivot_longer(cols = c(A, R, F),
               names_to = "species",
               values_to = "sim") %>%
  mutate(species = recode(species,
                          A = "algae",
                          R = "rotifers",
                          F = "flagellates"))

plot_data <- left_join(sim_long, orange_obs_long, by = c("time" = "day", "species"))

ggplot(plot_data, aes(x = time)) +
  geom_line(aes(y = sim, color = species), linewidth = 1) +
  geom_point(aes(y = obs, color = species, shape = species), size = 2) +
  scale_color_manual(values = c(
    algae = "green4",
    flagellates = "purple",
    rotifers = "red"
  )) +
  scale_shape_manual(values = c(
    algae = 16, flagellates = 17, rotifers = 15
  )) +
  labs(
    title = "POMP Simulation vs. Observed Data (Relative Abundance)",
    y = "Relative Abundance",
    x = "Day", color = "Species", shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "top")

```

POMP Simulation vs. Observed Data (Relative Abundance)



```
pf_result <- pfilter(pomp_model, params = params, Np = 1000)
logLik(pf_result)
```

```
## [1] -853.7633
```

```
library(doFuture)
library(doRNG)
library(foreach)
plan(multisession)
registerDoFuture()

rw_sd_values <- rw_sd(
  delta = 0.02,
  kA     = 0.02,
  kR     = 0.02,
  r      = 0.02,
  g      = 0.02,
  h      = 0.02,
  alphaA = 0.02,
  alphaF = 0.02,
  eta    = 0.02,
  IF     = 0.02,
  S_0    = ivp(0.02),
  A_0    = ivp(0.02),
  R_0    = ivp(0.02),
```

```

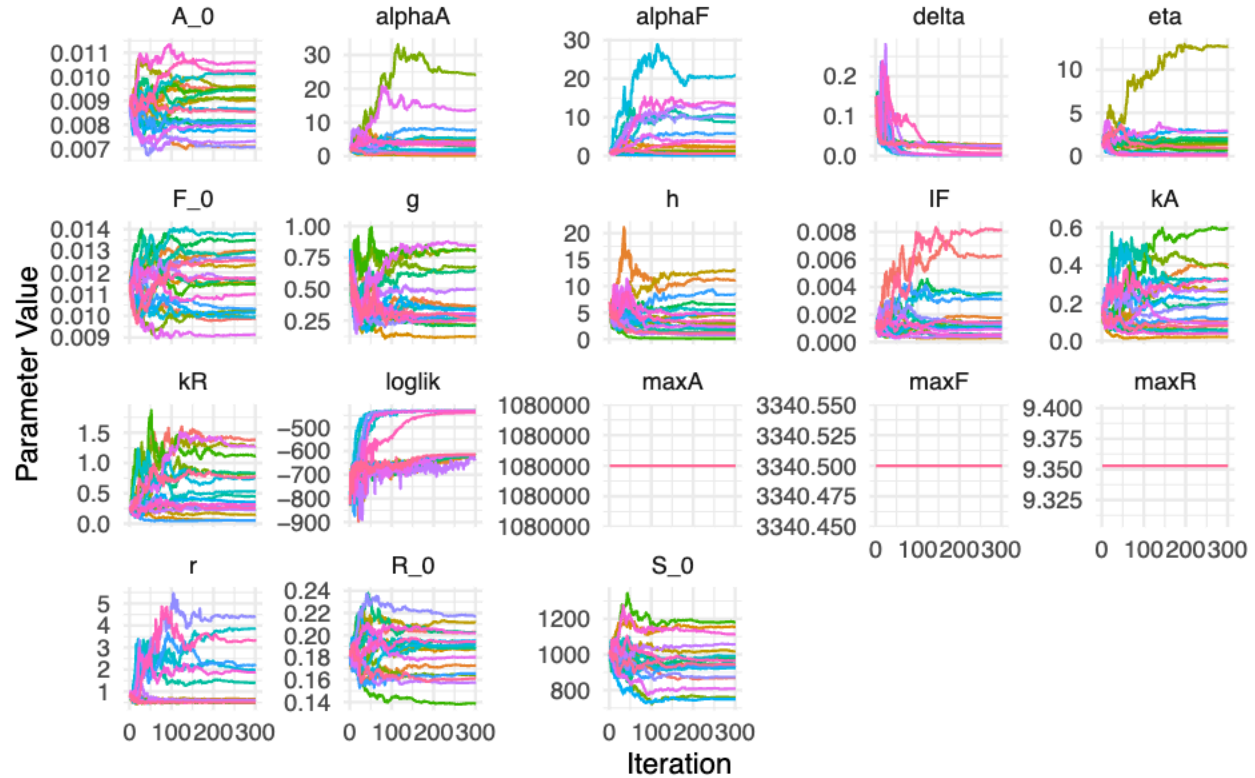
    F_0      = ivp(0.02)
  )

mifs_local <- foreach(i = 1:20, .combine = c) %dorange% {
  bake(file = tempfile(), {
    mif2(
      pomp_model,
      params = params,
      Np = 500,
      Nmif = 300,
      cooling.fraction.50 = 0.5,
      rw.sd = rw_sd_values
    )
  })
}

mifs_local |>
  traces() |>
  melt() |>
  filter(name != "loglik" | (value > -1e3)) |>
  ggplot(aes(x = iteration, y = value, group = .L1, color = factor(.L1))) +
  geom_line() +
  facet_wrap(~ name, scales = "free_y") +
  theme_minimal() +
  guides(color = "none") +
  labs(
    title = "Parameter Traces (Filtered)",
    x = "Iteration", y = "Parameter Value"
  )

```

Parameter Traces (Filtered)



```
# Compute log-likelihood via pfilter
local_search <- foreach(mf = mifs_local, .combine = rbind) %dorn% {
  ll_evals <- replicate(50, logLik(pfilter(mf, Np = 5000)))
  ll <- logmeanexp(ll_evals, se = TRUE)
  coef(mf) %>% bind_rows() %>%
    bind_cols(loglik = ll[1], loglik.se = ll[2])
}

# Filter and summarize
best_searches <- local_search %>%
  filter(is.finite(loglik)) %>%
  filter(loglik.se < 0.5) %>%
  arrange(desc(loglik))

head(best_searches)

## # A tibble: 6 x 19
##   delta    kA    kR    r    g    h alphaA alphaF    eta    IF    S_0
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0.00109 0.327 0.446 1.40 0.350 1.89 1.34 8.81 0.120 0.000572 947.
## 2 0.00190 0.195 0.270 4.41 0.252 1.08 0.936 9.98 0.148 0.00129 874.
## 3 0.00224 0.117 0.0510 2.22 0.293 8.42 7.58 5.87 0.0799 0.00312 938.
## 4 0.00208 0.0555 0.530 3.88 0.265 5.52 5.41 10.6 0.167 0.00147 991.
## 5 0.00211 0.0399 0.319 1.88 0.301 4.80 4.07 13.6 0.210 0.000545 1114.
## 6 0.00233 0.320 0.757 1.97 0.345 2.32 2.01 20.8 0.330 0.00345 924.
## # i 8 more variables: A_0 <dbl>, R_0 <dbl>, F_0 <dbl>, maxA <dbl>, maxR <dbl>,
```

```
## #   maxF <dbl>, loglik <dbl>, loglik.se <dbl>
```

```
summary(best_searches$loglik)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -628.7  -621.4  -616.3  -553.9  -430.3  -428.4
```