

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)
```

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)

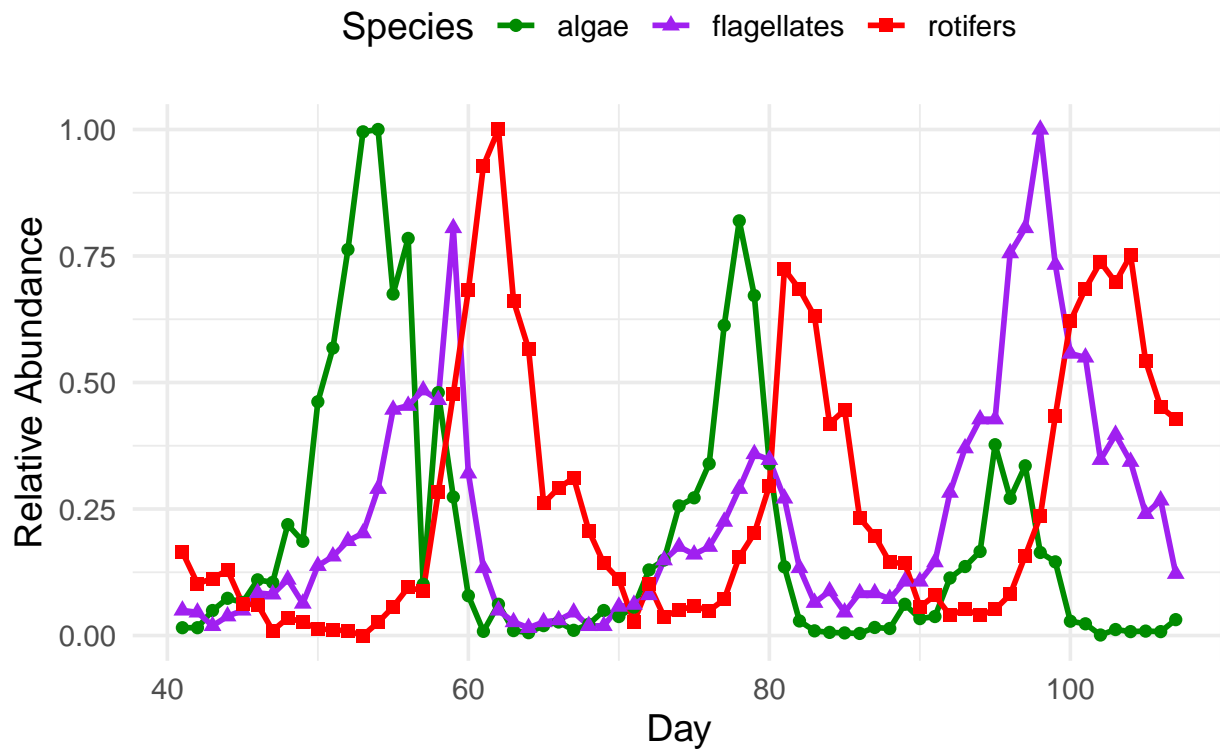
orange_rel <- orange_data
orange_rel[, 2:4] <- scale(orange_data[, 2:4], center = FALSE,
                          scale = c(max_algae, max_flag, max_roti))

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 = 0.25,
  kR = 0.59,
  r = 0.5,
  g = 0.3,

```

```

h = 2,
alphaA = 0.3,
alphaF = 1,
eta = 2.5,
IF = 0.002,
S_0 = 80,
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")

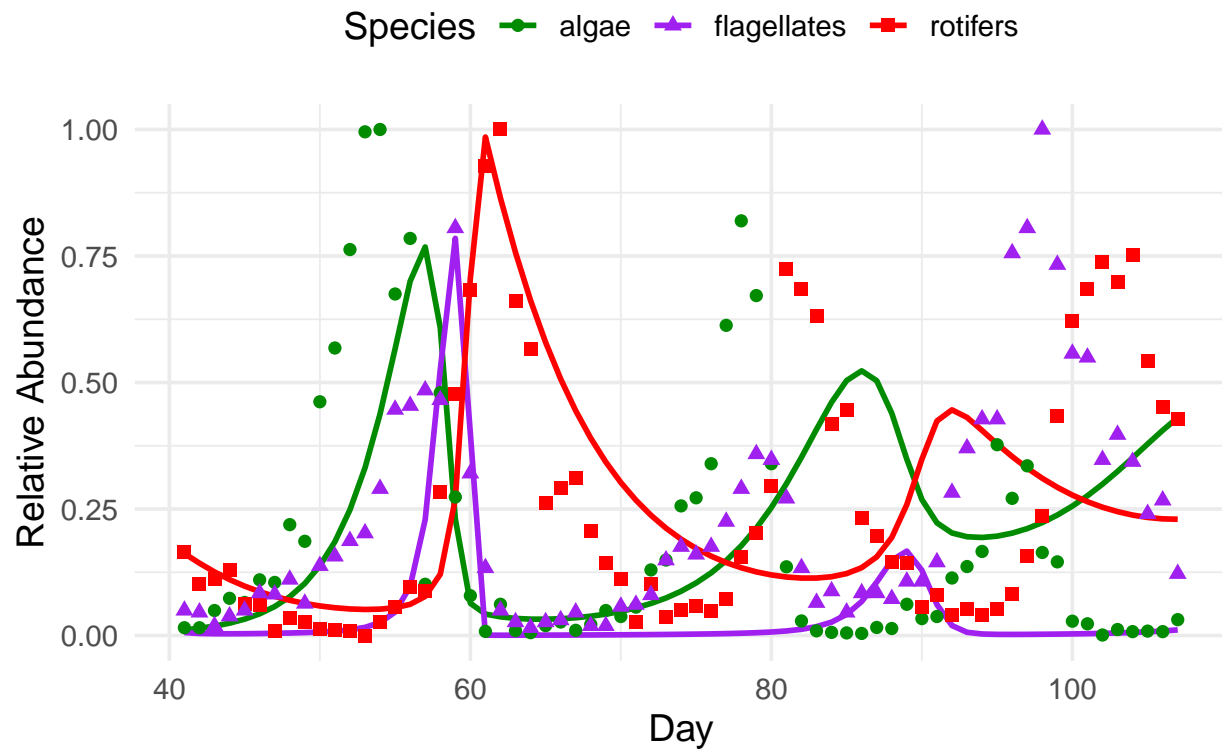
sim_long <- sim_df %>%
  mutate(time = day) %>%
  pivot_longer(cols = c(A, R, F),
               names_to = "species",
               values_to = "sim") %>%
  mutate(species = recode(species,
                          A = "algae",
                          R = "rotifers",
                          F = "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] -827.5218
```

```
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
)
```

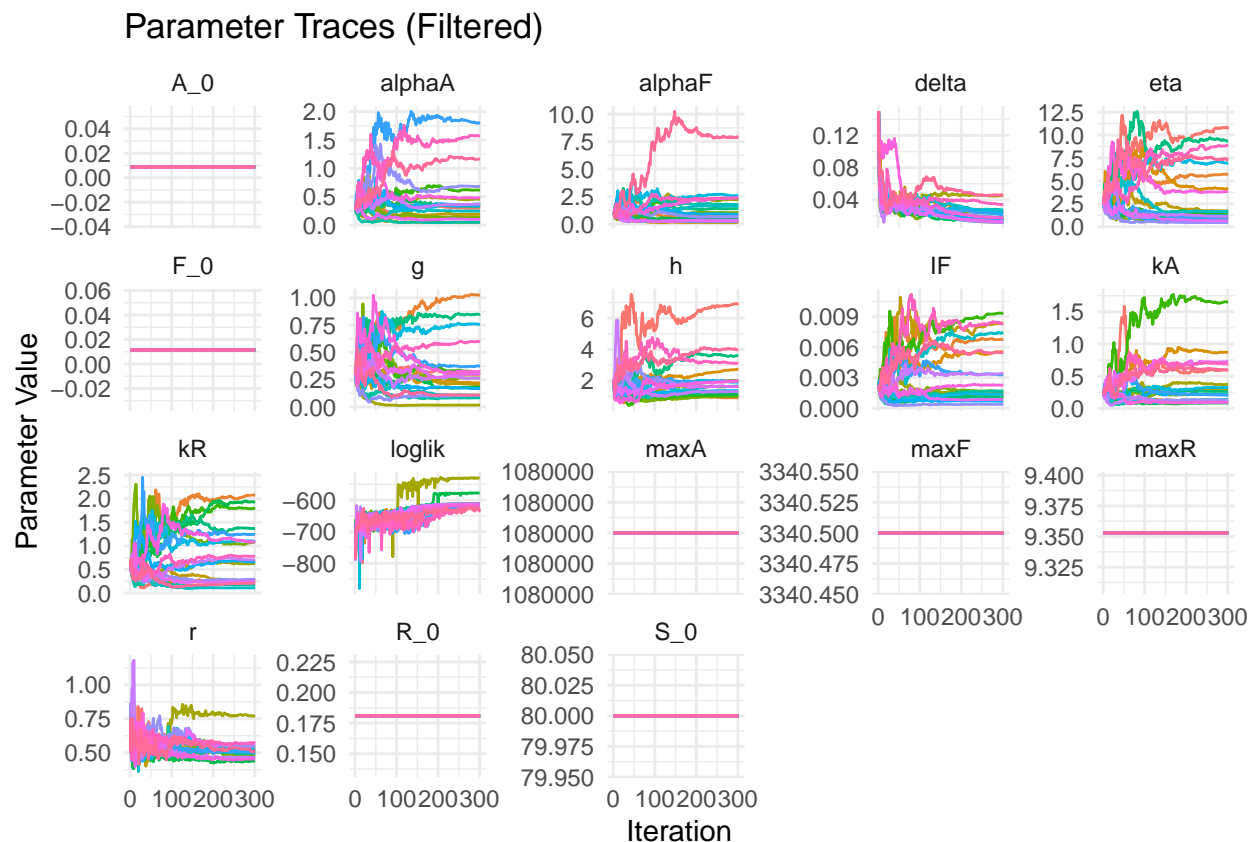
```
mifs_local <- foreach(i = 1:20, .combine = c) %dorng% {
```

```

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"
  )

```




```

# 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 0.0457 0.373 0.620 0.769 0.0166 1.63 0.456 2.23 1.06 0.00167 80
## 2 0.0170 0.0822 1.93 0.439 0.181 1.14 0.334 0.514 1.69 0.00113 80.0
## 3 0.0115 0.0920 0.702 0.467 0.263 1.32 0.369 0.348 1.13 0.00340 80.0
## 4 0.0126 0.137 0.173 0.496 0.0840 1.36 0.0477 1.78 1.23 0.00167 80.0
## 5 0.0118 0.215 1.08 0.483 0.265 0.998 0.139 1.04 0.941 0.00164 80.0
## 6 0.0167 0.303 0.256 0.523 0.203 2.03 0.193 0.147 1.71 0.00839 80.0
## # 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.
## -628.3 -616.8 -614.2 -609.7 -611.9 -529.0

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

orange_obs_long <- orange_obs_long %>%
  rename(abundance = obs)

simulated_data <- simulate(
  pomp_model,
  params = unlist(best_searches[1,]),
  nsim = 1,
  format = "data.frame",
  include.data = TRUE
)

sim_df <- as.data.frame(simulated_data, include.data = TRUE)

```

```

sim_long <- sim_df %>%
  mutate(
    A = A / maxA,
    R = R / maxR,
    F = F / maxF
  ) %>%
  pivot_longer(cols = c(A, R, F),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(
    abundance = pmax(log_abundance, 1e-6),
    species = recode(species,
                     A = "algae",
                     R = "rotifers",
                     F = "flagellates"),
    type = "Simulated"
  )

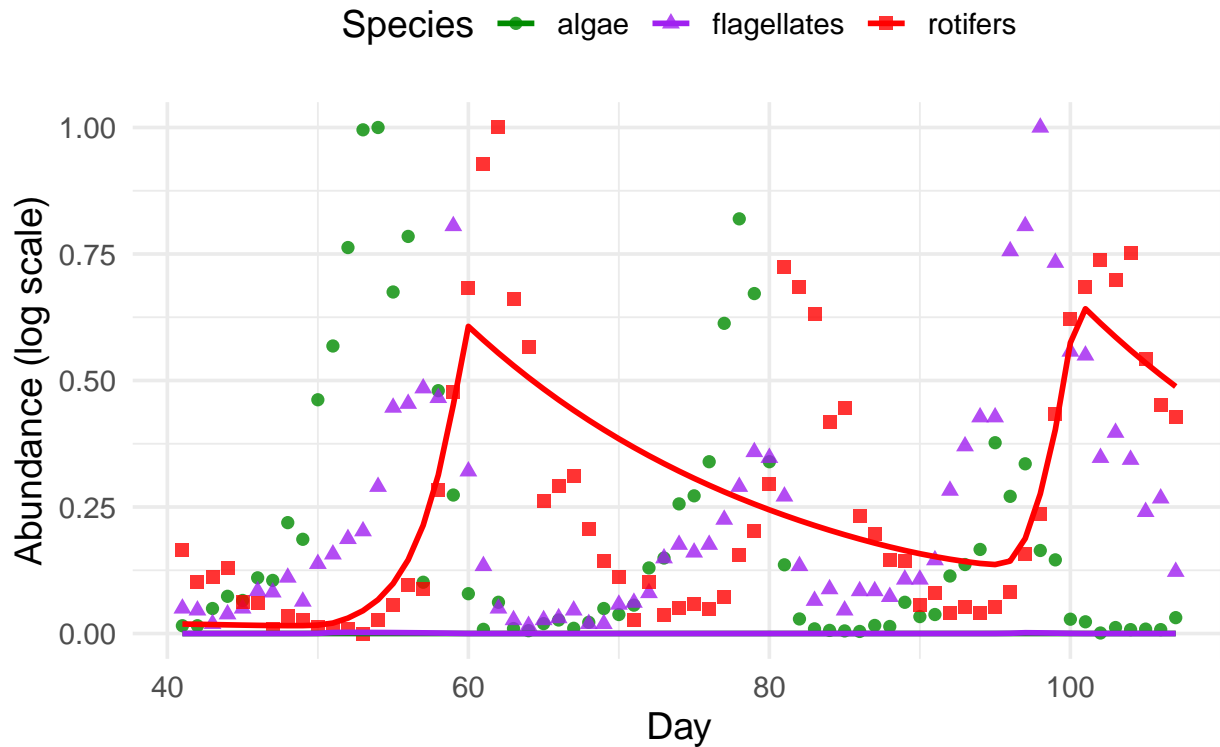
sim_long <- sim_long %>%
  filter(!is.na(abundance), is.finite(abundance), abundance >= 0, abundance <= 1)

combined_data <- bind_rows(orange_obs_long, sim_long)

ggplot(combined_data, aes(x = day, y = abundance, color = species, shape = species)) +
  geom_point(data = filter(combined_data, type == "Observed"), size = 2, alpha = 0.8) +
  geom_line(data = filter(combined_data, type == "Simulated"), linewidth = 1) +
  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",
    x = "Day",
    y = "Abundance (log scale)",
    color = "Species",
    shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )

```

POMP Simulation vs. Observed Data



```
library(doFuture)
library(iterators)
plan(multisession)
library(purrr)

# Full parameter list including scaling factors
paramnames <- c(
  "delta", "kA", "kR", "r", "g", "h",
  "alphaA", "alphaF", "eta", "IF",
  "S_0", "A_0", "R_0", "F_0",
  "maxA", "maxF", "maxR"
)

# Step 1: define narrowed random parameter ranges
best_param <- coef(mifs_local[[1]])
param_range <- runif_design(
  lower = best_param * 0.5,
  upper = best_param * 1.5,
  nseq = 100
)

# Step 2: pick a starting mif2 object from local search
mf_start <- mifs_local[[1]]
```

```

# Step 3: Run mif2 for each random parameter set
bake(file = "mif2_global.rds", seed = 888888, {
  foreach(p = iter(param_range, "row"), .combine = c,
    .options.future = list(seed = TRUE)) %dofuture% {
    mif2(
      mf_start,
      Nmif = 200,
      params = unlist(p),
      Np = 1000,
      cooling.fraction.50 = 0.5,
      rw.sd = rw_sd_values
    )
  }
}) -> mif2_results

# Step 4: Evaluate log-likelihoods
bake(file = "pfilter_ll.rds", seed = 999999, {
  foreach(m = mif2_results, .combine = rbind,
    .options.future = list(seed = TRUE)) %dofuture% {
    replicate(20, logLik(pfilter(m, Np = 5000))) |>
      logmeanexp(se = TRUE)
  }
}) -> loglik_results

# Step 5: Combine and sort
mif2_params <- map_dfr(mif2_results, coef)

mif2_global_summary <- mif2_params %>%
  bind_cols(
    loglik = loglik_results[, 1],
    loglik.se = loglik_results[, 2]
  ) %>%
  arrange(desc(loglik))

# Step 6: Filter good results
best_searches <- mif2_global_summary %>%
  filter(is.finite(loglik), loglik.se < 0.5) %>%
  arrange(-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 0.0968 1.57   0.171 0.571 0.0324 23.0 0.602 0.341 69.5 0.00514 92.5
## 2 0.127  1.18   0.0512 0.630 0.0374 2.24 0.0968 0.305 22.2 0.00421 71.5
## 3 0.138  0.969 0.0689 0.639 0.0410 2.87 0.0896 0.331 32.2 0.00473 118.
## 4 0.111  0.0494 0.0677 0.570 0.0380 11.5 0.397 0.637 53.5 0.00340 84.1
## 5 0.121  0.864 0.0452 0.601 0.0319 18.7 0.481 1.93 43.9 0.00233 85.1
## 6 0.162  0.844 0.0796 1.00 0.163 11.3 0.449 0.0393 69.9 0.00314 96.2
## # 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.
## -4723.9 -628.7  -620.2  -725.4  -613.8  -556.9
```

```
# Simulate best
```

```
params <- best_searches[1, paramnames] |> unlist()
```

```
simulated_data <- simulate(
  pomp_model,
  params = params,
  nsim = 1,
  format = "data.frame",
  include.data = TRUE
)
```

```
summary(simulated_data)
```

```
##      day      .id      algae      flagellates
## Min.   : 41.00  data:67  Min.   :-0.30798  Min.   :-0.249371
## 1st Qu.: 57.25   1 :67   1st Qu.: -0.01901  1st Qu.: -0.008518
## Median : 74.00           Median : 0.02855  Median : 0.054112
## Mean   : 74.00           Mean   : 0.08730  Mean   : 0.114788
## 3rd Qu.: 90.75           3rd Qu.: 0.13426  3rd Qu.: 0.175572
## Max.   :107.00           Max.   : 1.00000  Max.   : 1.000000
##
##      rotifers      S      A      R
## Min.   :-0.2111  Min.   : 0.07915  Min.   :0.00001  Min.   :0.4694
## 1st Qu.: 0.0898  1st Qu.: 0.28068  1st Qu.:0.06036  1st Qu.:1.1756
## Median : 0.2279  Median : 0.69871  Median :0.46865  Median :2.5231
## Mean   : 0.2629  Mean   :11.95062  Mean   :1.02779  Mean   :2.8744
## 3rd Qu.: 0.3908  3rd Qu.:14.36371  3rd Qu.:1.65733  3rd Qu.:4.0068
## Max.   : 1.0000  Max.   :83.96676  Max.   :4.10832  Max.   :6.4811
##           NA's :67           NA's :67           NA's :67
##
##      F
## Min.   :0.00069
## 1st Qu.:0.00119
## Median :0.00123
## Mean   :0.00725
## 3rd Qu.:0.00126
## Max.   :0.40345
## NA's   :67
```

```
sim_df <- as.data.frame(simulated_data, include.data = FALSE)
```

```
# Standardize simulation to match measurement model
```

```
sim_long <- sim_df %>%
```

```
  mutate(
    A = A / maxA,
    R = R / maxR,
    F = F / maxF
  ) %>%
```

```

pivot_longer(cols = c(A, R, F),
             names_to = "species",
             values_to = "rel_abundance") %>%
mutate(
  species = recode(species,
                   A = "algae",
                   R = "rotifers",
                   F = "flagellates"),
  type = "Simulated"
)

# Combine and plot
obs_long <- orange_rel %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "rel_abundance") %>%
  mutate(type = "Observed")

combined_data <- bind_rows(obs_long, sim_long)

ggplot(combined_data, aes(x = day, y = rel_abundance, color = species)) +
  geom_point(data = filter(combined_data, type == "Observed"),
            aes(shape = species), size = 2, alpha = 0.8) +
  geom_line(data = filter(combined_data, type == "Simulated"),
           linewidth = 1) +
  scale_color_manual(values = c(
    algae = "green4",
    flagellates = "purple",
    rotifers = "red"
  )) +
  scale_shape_manual(values = c(
    algae = 16, flagellates = 17, rotifers = 15
  )) +
  labs(
    title = "Global Search: POMP Simulation vs Observed",
    x = "Day",
    y = "Relative Abundance",
    color = "Species",
    shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "top")

```

```

## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

Global Search: POMP Simulation vs Observed

