POMP Model Analysis

Sijia Wang

2025-06-11

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

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

POMP Model Fitting

We use the **orange panel data between days 41 and 107** to build and analyze a Partially Observed Markov Process (POMP) model for the three-species food web system.

```r
# Load the data
X123 <- read.csv("NoEvoData=SE=Feb14 2012.csv")

# Prepare orange panel data (log-transformed and mean-centered)
orange_data <- data.frame(
  day = X123$day,
  algae = log(X123$Algae.orange1),
  flagellates = log(X123$Flag.orange1),
  rotifers = log(pmax(X123$Rot.orange1, 1e-3))
)

# Subset data between day 41 and 107
orange_data <- orange_data[orange_data$day >= 41 & orange_data$day <= 107, ]

# Simple mean-centering
orange_data <- orange_data %>%
  mutate(
    algae = algae,
    flagellates = flagellates,
    rotifers = rotifers
  )
```

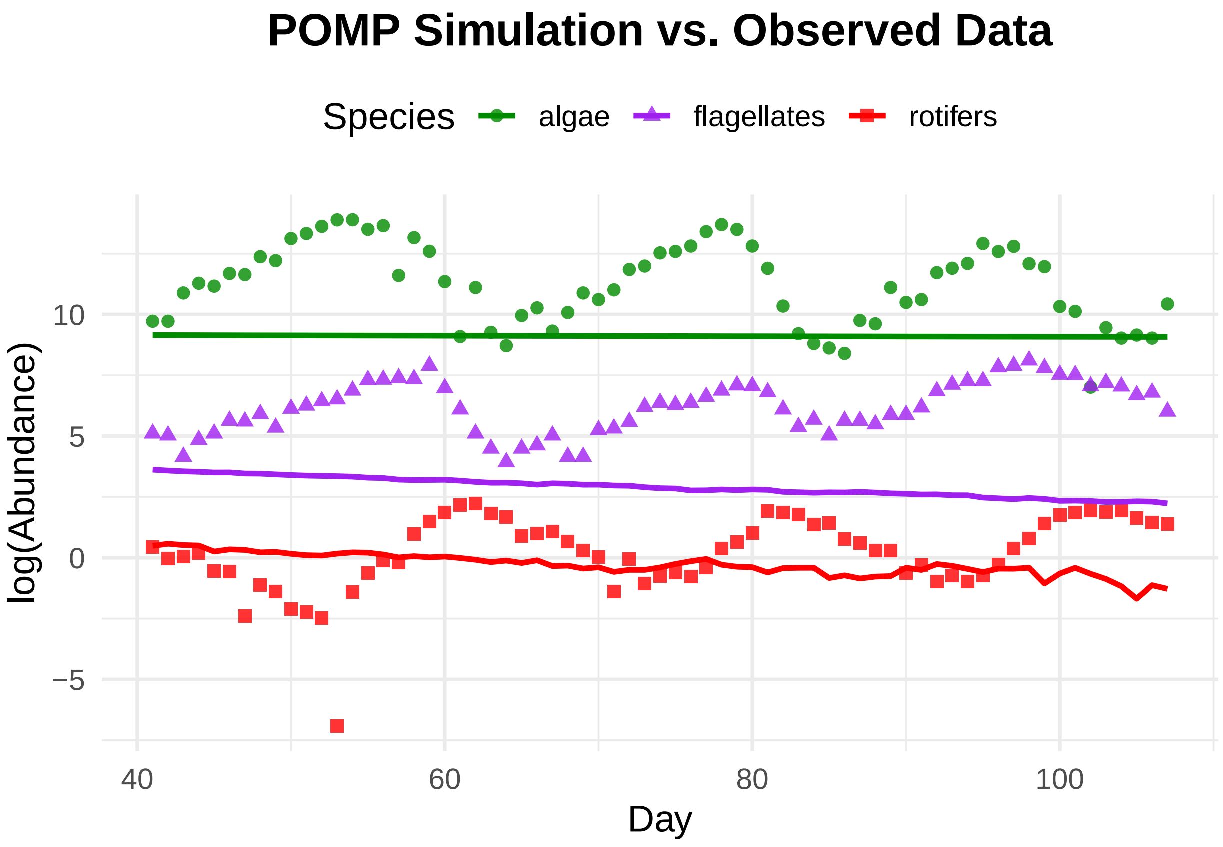
```r
rmeasure <- Csnippet("
  algae = rnorm(log(A), sigma_A);
  flagellates = rnorm(log(F), sigma_F);
  rotifers = rnorm(log(R), sigma_R);
")

dmeasure <- Csnippet("
  // Convert states to mean-centered log scale
  lik = dnorm(algae, log(A), sigma_A, 1) +
        dnorm(flagellates, log(F), sigma_F, 1) +
        dnorm(rotifers, log(R), sigma_R, 1);

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

# Updated parameter names to include the means
paramnames <- c(
  "delta", "kA", "kR", "r", "g", "h",
  "alphaA", "alphaF", "eta", "IF",
  "S_0", "A_0", "R_0", "F_0",
  "sigma_A", "sigma_F", "sigma_R",
  "sigma_proS", "sigma_proA",
  "sigma_proF", "sigma_proR"
)

# Process model (same as before)
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;

    double eS = rnorm(0, sigma_proS * sqrt(dt));
    double eA = rnorm(0, sigma_proA * sqrt(dt));
    double eR = rnorm(0, sigma_proR * sqrt(dt));
    double eF = rnorm(0, sigma_proF * sqrt(dt));

    S += dS * dt + eS;
    A += dA * dt + eA;
    R += dR * dt + eR;
    F += dF * dt + eF;

    if (S < 80) S = 80;
    if (A < 0) A = 1e-3;
    if (R < 0) R = 1e-3;
    if (F < 0) F = 1e-3;

    if (S > 1e6) S = 1e6;
    if (A > 1e8) A = 1e8;
    if (R > 1e6) R = 1e6;
    if (F > 1e6) F = 1e6;
  "),
  delta.t = 0.25
)

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

# Parameter transformation
pt <- parameter_trans(
  log = c("delta", "kA", "kR", "r", "g", "h",
          "alphaA", "alphaF", "eta", "IF",
          "S_0", "A_0", "R_0", "F_0",
          "sigma_A", "sigma_F", "sigma_R",
          "sigma_proS", "sigma_proA",
          "sigma_proF", "sigma_proR")
)

# Create POMP model
pomp_model <- pomp(
  data = orange_data,
  times = "day",
  t0 = 40,
  rprocess = rproc,
  rmeasure = rmeasure,
  dmeasure = dmeasure,
  rinit = rinit,
  statenames = c("S", "A", "R", "F"),
  paramnames = paramnames,
  obsnames = c("algae", "flagellates", "rotifers"),
  partrans = pt
)

# Get initial conditions
init_row <- X123[X123$day == 40, ]

# Parameter vector including the means
params <- c(
```

```r
  delta = 0.25,
  kA = 0.2, # smaller kA, algae can grow faster even at low nutrient levels
  r = 0.25, # algae growth rate
  kR = 1.6, # lower kR, predators (F and R) can respond strongly even when prey is scarce
  g = 0.2, # higher g, rotifers grow faster from consuming algae.
  h = 0.5, # higher h, algae are grazed faster by flagellates, and flagellates grow faster
  alphaA = 2.33, # Higher  A, flagellates saturate faster
  alphaF = 0.4, # rotifer's feeding rate on flagellates
  eta = 3.6, # rotifer feeding on flagellates
  IF = 0.3, # rotifer's feeding saturation constant
  S_0 = 80,
  A_0 = init_row$Algae.orange1,
  R_0 = init_row$Rot.orange1,
  F_0 = init_row$Flag.orange1,
  sigma_A = 0.1,
  sigma_F = 0.1,
  sigma_R = 0.1,
  sigma_proS = 0.1,
  sigma_proA = 2,
  sigma_proF = 0.5,
  sigma_proR = 0.1
)
```

```r
# Simulate and plot
sim <- simulate(pomp_model, params = params, nsim = 1)
sim_df <- as.data.frame(sim)
summary(sim_df)
```

```
##       day            algae         flagellates       rotifers
##  Min.   : 41.0   Min.   :8.851   Min.   :2.150   Min.   :-1.66936
##  1st Qu.: 57.5   1st Qu.:9.033   1st Qu.:2.607   1st Qu.:-0.54541
##  Median : 74.0   Median :9.088   Median :2.940   Median :-0.33298
##  Mean   : 74.0   Mean   :9.096   Mean   :2.918   Mean   :-0.29119
##  3rd Qu.: 90.5   3rd Qu.:9.158   3rd Qu.:3.184   3rd Qu.: 0.07498
##  Max.   :107.0   Max.   :9.323   Max.   :3.666   Max.   : 0.65749
##        S            A               R                F
##  Min.   :80   Min.   :8757   Min.   :0.1858   Min.   : 9.333
##  1st Qu.:80   1st Qu.:8887   1st Qu.:0.6067   1st Qu.:13.710
##  Median :80   Median :9055   Median :0.7117   Median :17.440
##  Mean   :80   Mean   :9065   Mean   :0.8236   Mean   :19.776
##  3rd Qu.:80   3rd Qu.:9227   3rd Qu.:1.0581   3rd Qu.:24.718
##  Max.   :80   Max.   :9433   Max.   :1.7782   Max.   :37.362
```

```r
# Prepare observed data for plotting
obs_long <- orange_data %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Observed")

# Prepare simulation data - convert to log scale to match observations
sim_long <- sim_df %>%
  mutate(
    algae = log(pmax(A, 1e-6)),
    flagellates = log(pmax(F, 1e-6)),
    rotifers = log(pmax(R, 1e-6))
  ) %>%
  select(day, algae, flagellates, rotifers) %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Simulated")

# Combine and plot
combined_data <- bind_rows(obs_long, sim_long)

ggplot(combined_data, aes(x = day, y = log_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 = "POMP Simulation vs. Observed Data",
    x = "Day",
    y = "log(Abundance)",
    color = "Species",
    shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

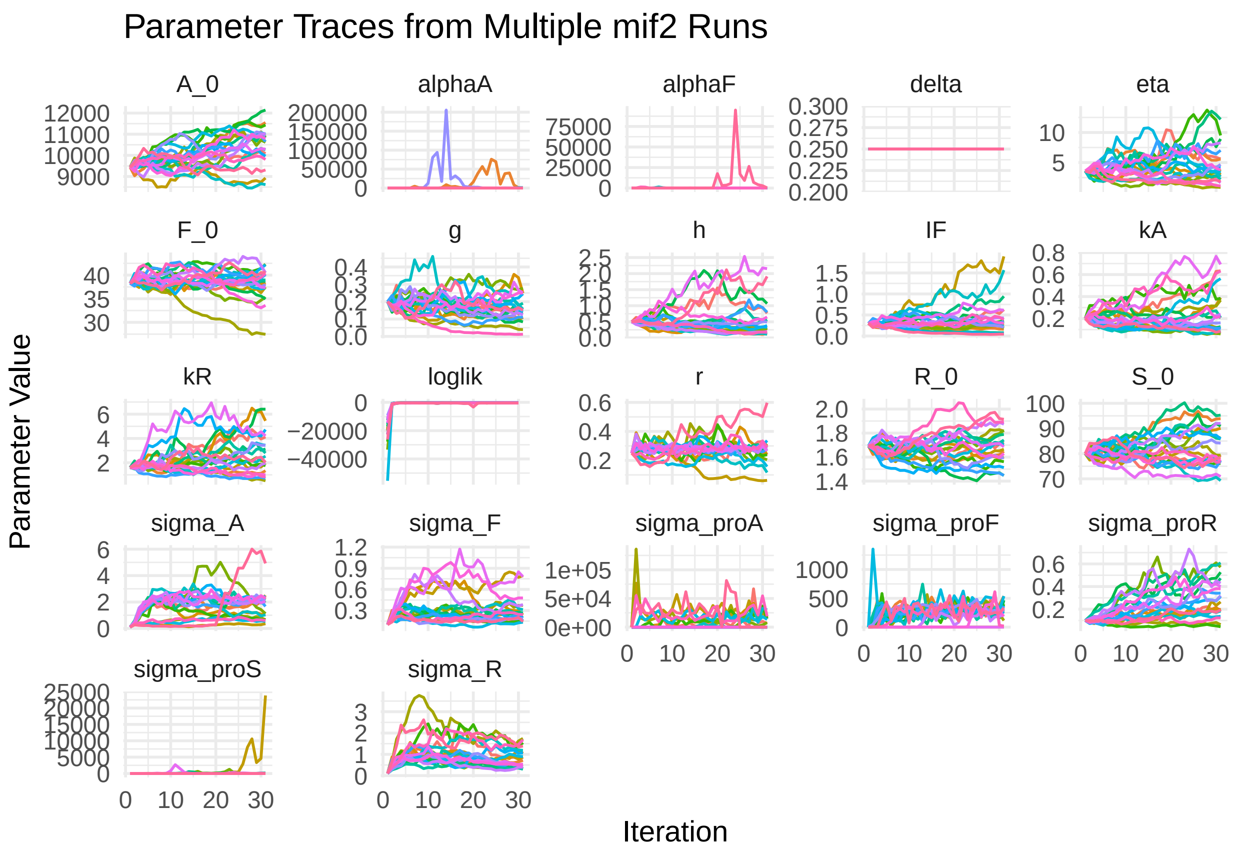


```r
library(future)
library(doParallel)
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```r
library(foreach)
plan(multisession)

# Step 1: initial guess
init_params <- params

# Step 2: define random walk standard deviations for each parameter
rw_sd_values <- rw_sd(
  kA        = 0.02,
  kR        = 0.02,
  r         = 0.02,
  g         = 0.02,
  h         = 0.02,
  alphaA    = 0.2,
  alphaF    = 0.2,
  eta       = 0.02,
  IF        = 0.02,
  sigma_A = 0.02,
  sigma_F = 0.02,
  sigma_R = 0.02,
  sigma_proS = 0.2,
  sigma_proA = 0.5,
  sigma_proF = 0.2,
  sigma_proR = 0.02,
  A_0 = ivp(0.02),
  F_0 = ivp(0.02),
  R_0 = ivp(0.02),
  S_0 = ivp(0.02)
)

cores <-  as.numeric(Sys.getenv('SLURM_NTASKS_PER_NODE', unset=NA))
if(is.na(cores)) cores <- detectCores()
registerDoParallel(cores)
ggplot2::theme_set(ggplot2::theme_bw())

stopifnot(packageVersion("pomp")>="5.0")

# Perform multiple mif2 fits in parallel to obtain multiple parameter estimates
mifs_local <- foreach(
  i = 1:20,
  .combine = c,
  .options.future = list(seed = 89898975),
  .packages = "pomp"
) %dopar% {
  mif2(
    pomp_model,
    params = init_params,
    Np = 1000,                # Number of particles
    Nmif = 30,                # Number of mif iterations
    cooling.fraction.50 = 0.5,  # Cooling fraction
    rw.sd = rw_sd_values      # Random walk standard deviations
  )
}

#parameter traces
mifs_local |>
  traces() |>
  melt() |>
  ggplot(aes(x = iteration, y = value, group = .L1, color = factor(.L1))) +
  geom_line() +
  guides(color = "none") +
  facet_wrap(~ name, scales = "free_y") +
  theme_minimal() +
  labs(
    title = "Parameter Traces from Multiple mif2 Runs",
    x = "Iteration",
    y = "Parameter Value"
  )
```



```
local_search <- foreach(
  mf = mifs_local,
  .combine = rbind
) %dopar% {
  evals <- replicate(10, logLik(pfilter(mf, Np = 5000)))
  ll <- logmeanexp(evals, se = TRUE)
  mf %>% coef() %>% bind_rows() %>%
    bind_cols(loglik = ll[1], loglik.se = ll[2])
}

bind_rows(local_search) %>%
  filter(is.finite(loglik)) %>%
  filter(loglik.se < .5) %>%
  arrange(-loglik) -> best_searches

head(best_searches)
```

```
## # A tibble: 6 x 23
##   delta    kA     r    kR      g     h alphaA   alphaF   eta     IF   S_0    A_0
##   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl>    <dbl> <dbl>  <dbl> <dbl>  <dbl>
## 1 0.250 0.210 0.287 6.41  0.0837 1.07     5.26 0.00515   8.92 0.340   90.9 12144.
## 2 0.25  0.127 0.296 3.13  0.155  0.513    3.56 0.171    12.2  0.945   95.3 10820.
## 3 0.250 0.208 0.282 2.79  0.191  0.566    2.87 0.0849    2.21 0.0783  85.9 10260.
## 4 0.250 0.240 0.282 1.96  0.124  0.197    1.25 0.274     8.26 0.427   92.0 11048.
## 5 0.250 0.111 0.278 0.709 0.110  0.798    4.75 2.50      7.03 0.349   77.2 10651.
## 6 0.250 0.634 0.300 4.12  0.113  0.765   3.80 0.000519  5.60 0.629   79.5 10083.
## # i 11 more variables: R_0 <dbl>, F_0 <dbl>, sigma_A <dbl>, sigma_F <dbl>,
## #   sigma_R <dbl>, sigma_proS <dbl>, sigma_proA <dbl>, sigma_proF <dbl>,
## #   sigma_proR <dbl>, loglik <dbl>, loglik.se <dbl>
```

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

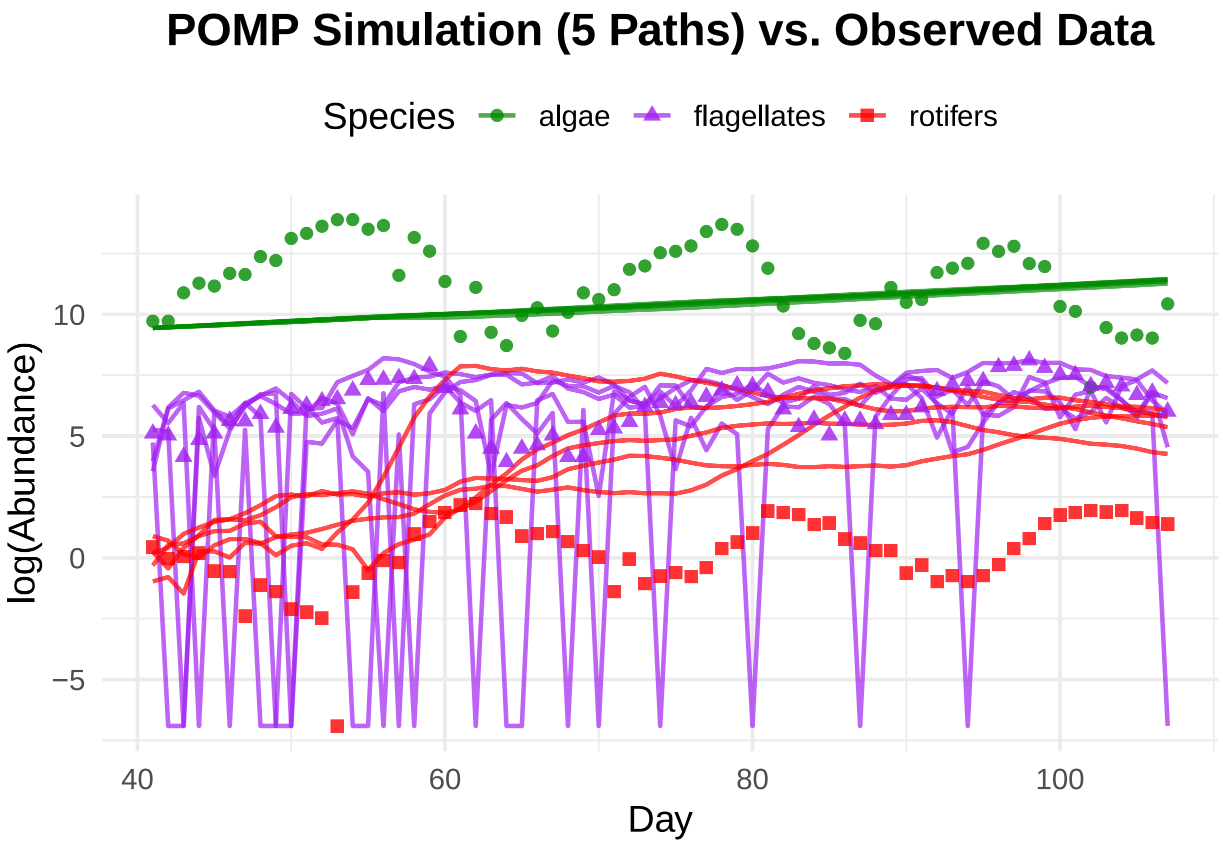
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -983.6  -345.8  -293.0  -398.1  -275.6  -273.4
```

```
params_best <- best_searches[1, ] %>% unlist() %>% as.numeric()
names(params_best) <- names(best_searches)
simulated_data <- simulate(
  pomp_model,
  params = params_best,
  nsim = 5,
  format = "data.frame",
  include.data = TRUE
)

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

```
##       day         .id         algae          flagellates        rotifers
##  Min.   : 41   data:67   Min.   : 4.128   Min.   :-7.358   Min.   :-6.908
##  1st Qu.: 57   1   :67   1st Qu.: 8.854   1st Qu.: 5.853   1st Qu.: 1.411
##  Median : 74   2   :67   Median :10.534   Median : 6.500   Median : 3.646
##  Mean   : 74   3   :67   Mean   :10.348   Mean   : 5.650   Mean   : 3.503
##  3rd Qu.: 91   4   :67   3rd Qu.:11.859   3rd Qu.: 7.083   3rd Qu.: 5.920
##  Max.   :107   5   :67   Max.   :16.717   Max.   : 8.470   Max.   : 8.045
##
##        S             A                R                   F
##  Min.   :80   Min.   :12556   Min.   :   0.2309   Min.   :   0.001
##  1st Qu.:80   1st Qu.:20043   1st Qu.:  12.7748   1st Qu.: 369.392
##  Median :80   Median :31643   Median :  88.6258   Median : 694.015
##  Mean   :80   Mean   :38029   Mean   : 321.8617   Mean   : 884.632
##  3rd Qu.:80   3rd Qu.:52272   3rd Qu.: 469.4543   3rd Qu.:1196.150
##  Max.   :80   Max.   :94365   Max.   :2652.8085   Max.   :3642.942
##  NA's   :67   NA's   :67      NA's   :67          NA's   :67
```

```
# Prepare observed data for plotting
obs_long <- orange_data %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Observed")

# Prepare simulation data - convert to log scale to match observations
sim_long <- sim_df %>%
  filter(.id != "data") %>%
  mutate(
    algae = log(pmax(A, 1e-6)),
```

```r
    flagellates = log(pmax(F, 1e-6)),
    rotifers = log(pmax(R, 1e-6))
  ) %>%
  select(.id, day, algae, flagellates, rotifers) %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Simulated")


# Combine and plot
combined_data <- bind_rows(obs_long, sim_long)

ggplot() +
  geom_line(data = filter(sim_long, is.finite(log_abundance)),
            aes(x = day, y = log_abundance, color = species, group = interaction(.id, species)),
            linewidth = 0.8, alpha = 0.7) +
  geom_point(data = obs_long,
             aes(x = day, y = log_abundance, color = species, shape = species),
             size = 2, alpha = 0.8) +
  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 (5 Paths) vs. Observed Data",
    x = "Day",
    y = "log(Abundance)",
    color = "Species",
    shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```



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

# 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,
```

```r
      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 23
##   delta    kA     r     kR      g     h alphaA  alphaF    eta    IF   S_0    A_0
##   <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl>  <dbl>   <dbl>  <dbl> <dbl> <dbl>  <dbl>
## 1 0.262 1.10  0.246 1.16   0.218  2.90  12.6   2.97e+1  1.06  0.550  80.0  5144.
## 2 0.259 0.867 0.172 0.226  0.143  0.269  1.29  8.58e-1  0.380 1.88   97.1  6512.
## 3 0.263 0.890 0.288 0.155  0.121  0.502  2.60  8.99e-3  7.72  1.26   87.2 14250.
## 4 0.134 0.166 0.113 2.28   0.0331 0.761 44.8   9.01e-4  7.82  1.04  127.  11409.
## 5 0.205 0.264 0.159 0.322  0.175  0.118  0.653 1.22e+0  0.296 0.812  78.5  6617.
## 6 0.356 0.558 0.344 0.0817 0.188  2.85   9.74  1.25e-2 10.5   0.925  63.5 13664.
## # i 11 more variables: R_0 <dbl>, F_0 <dbl>, sigma_A <dbl>, sigma_F <dbl>,
## #   sigma_R <dbl>, sigma_proS <dbl>, sigma_proA <dbl>, sigma_proF <dbl>,
## #   sigma_proR <dbl>, loglik <dbl>, loglik.se <dbl>
```

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

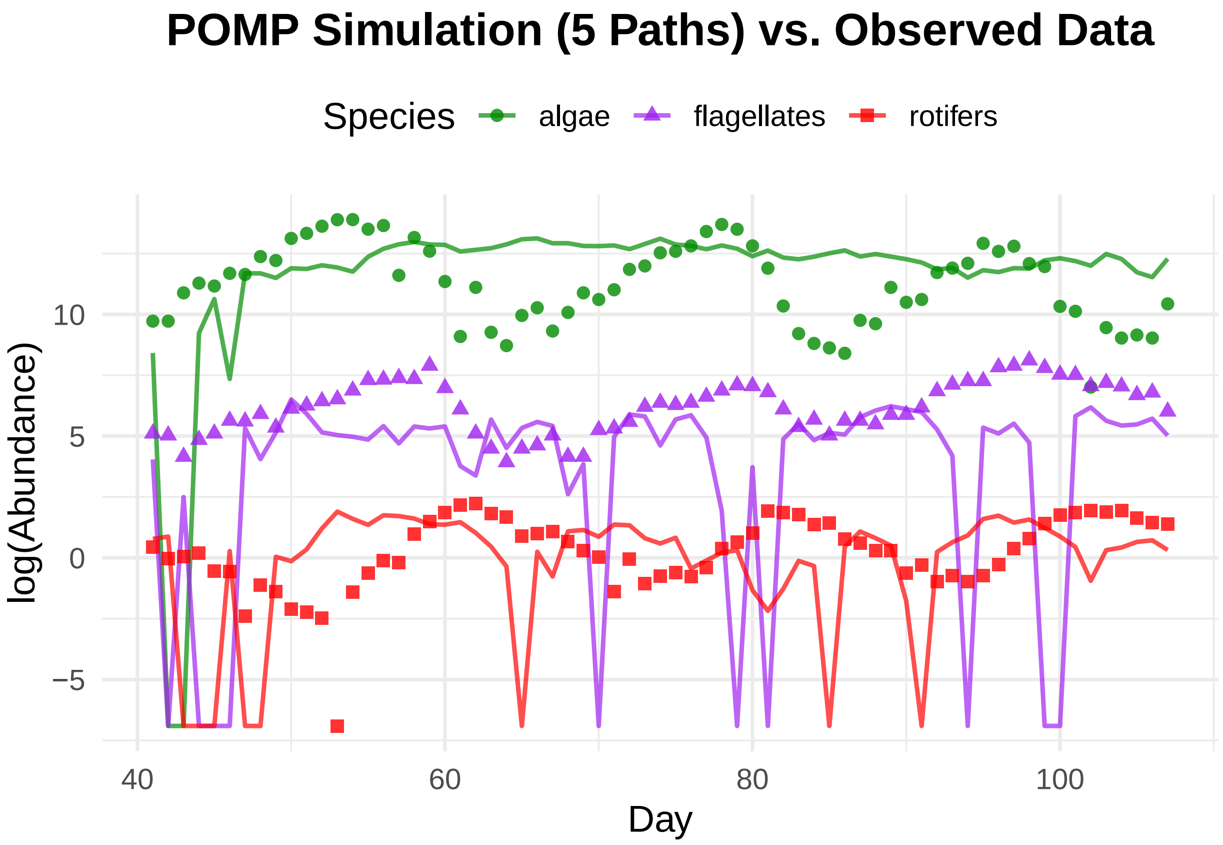
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -359.7  -290.1  -282.1  -282.6  -270.2  -254.6
```

```r
# Simulate best
params <- best_searches[1, paramnames] |> unlist()
```

```r
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.   :-9.143   Min.   :-7.591
##  1st Qu.: 57.25   1   :67   1st Qu.:10.726   1st Qu.: 4.637
##  Median : 74.00             Median :11.930   Median : 5.532
##  Mean   : 74.00             Mean   :11.356   Mean   : 4.700
##  3rd Qu.: 90.75             3rd Qu.:12.688   3rd Qu.: 6.309
##  Max.   :107.00             Max.   :14.687   Max.   : 8.114
##
##     rotifers              S                 A                R
##  Min.   :-7.03468   Min.   :     80   Min.   :     0   Min.   :0.0010
##  1st Qu.:-0.58056   1st Qu.:     80   1st Qu.:143326   1st Qu.:0.7922
##  Median : 0.42530   Median :1000000   Median :234618   Median :1.6570
##  Mean   :-0.05763   Mean   : 552275   Mean   :239837   Mean   :2.1220
##  3rd Qu.: 1.28994   3rd Qu.:1000000   3rd Qu.:348831   3rd Qu.:3.2400
##  Max.   : 2.23569   Max.   :1000000   Max.   :499083   Max.   :6.6582
##                     NA's   :67        NA's   :67       NA's   :67
##        F
##  Min.   :  0.001
##  1st Qu.: 51.821
##  Median :164.964
##  Mean   :179.914
##  3rd Qu.:256.592
##  Max.   :667.235
##  NA's   :67
```

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

# Prepare observed data for plotting
obs_long <- orange_data %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Observed")

# Prepare simulation data - convert to log scale to match observations
sim_long <- sim_df %>%
  filter(.id != "data") %>%
  mutate(
    algae = log(pmax(A, 1e-6)),
    flagellates = log(pmax(F, 1e-6)),
    rotifers = log(pmax(R, 1e-6))
  ) %>%
  select(.id, day, algae, flagellates, rotifers) %>%
  pivot_longer(cols = c(algae, flagellates, rotifers),
               names_to = "species",
               values_to = "log_abundance") %>%
  mutate(type = "Simulated")


# Combine and plot
combined_data <- bind_rows(obs_long, sim_long)

ggplot() +
  geom_line(data = filter(sim_long, is.finite(log_abundance)),
            aes(x = day, y = log_abundance, color = species, group = interaction(.id, species)),
            linewidth = 0.8, alpha = 0.7) +
  geom_point(data = obs_long,
             aes(x = day, y = log_abundance, color = species, shape = species),
             size = 2, alpha = 0.8) +
  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 (5 Paths) vs. Observed Data",
    x = "Day",
    y = "log(Abundance)",
    color = "Species",
    shape = "Species"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```


POMP Simulation (5 Paths) vs. Observed Data
Species
algae
flagellates
rotifers
log(Abundance)
10
5
0
−5
Day
40
60
80
100