`matrixpop`: a matrix population model for plankton size classes

- model the size distribution of a plankton population

setup:

m	number of size classes
v_{min}	minimum size
Δ_v	determines size class spacing
	Δ_v^{-1} must be an integer

model:

$w(t) \in \mathbb{R}^m$	population
$A(t, \Delta_t) \in \mathbb{R}^{m \times m}$	time-propagation matrix

$w_i(t)$ proportion of cells at time t with size between v_i and v_{i+1} $(\sum_{i=1}^{n} w_i(t) = 1)$

$$w(t + \Delta_t) = \frac{A(t, \Delta_t) w(t)}{\|A(t, \Delta_t) w(t)\|_1}$$

size classes:

$$v_i = v_{min}\, 2^{(i-1)\Delta_v} \text{ for } i = 1, \dots, m+1$$

integer $j = 1 + \Delta_v^{-1}$ with

$$\begin{aligned} v_{i+j-1} &= v_{min}\, 2^{(i+j-1-1)\Delta_v} \\ &= v_{min}\, 2^{(i+\Delta_v^{-1}-1)\Delta_v} \\ &= v_{min}\, 2^{(i-1)\Delta_v} \cdot 2 \\ &= 2 \cdot v_i \end{aligned}$$

structure of A

growth: with size-independent growth rate $\gamma(t,\Delta_t) = \Delta_t \, \gamma_{max} \left(1 - \exp\left(-\frac{E(t)}{E^*}\right)\right)$ (2 parameters)

division: with size-dependent division rate $\delta_i(\Delta_t) = \Delta_t \, \delta_{max} \, \frac{v_i^b}{1+v_i^b}$ (2 parameters)

stasis: the rest of the cells remain at their size class

$$A(t,\Delta_t) = \begin{array}{c} \begin{array}{cccccccc} 1 & & & j-1 & j & & m-1 & m \\ \downarrow & & & \downarrow & \downarrow & & \downarrow & \downarrow \end{array} \\ \left(\begin{array}{cccccccc} v_1 \to v_1 & & & & v_j \to v_1 = \frac{v_j}{2} & & & \\ v_1 \to v_2 & \ddots & & & & \ddots & & \\ & \ddots & \ddots & & & & \ddots & \\ & & \ddots & \ddots & & & & v_m \to v_{m-j+1} \\ & & & \ddots & v_j \to v_j & & & \\ & & & & v_j \to v_{j+1} & \ddots & & \\ & & & & & \ddots & \ddots & \\ & & & & & & v_{m-1} \to v_m & v_m \to v_m \end{array}\right) \end{array} \begin{array}{l} \leftarrow 1 \\ \leftarrow 2 \\ \\ \\ \\ \\ \leftarrow m-1 \\ \leftarrow m \end{array}$$

structure of A

growth: with size-independent growth rate $\gamma(t,\Delta_t) = \Delta_t\, \gamma_{max} \left(1 - \exp\left(-\frac{E(t)}{E^*}\right)\right)$ (2 parameters)

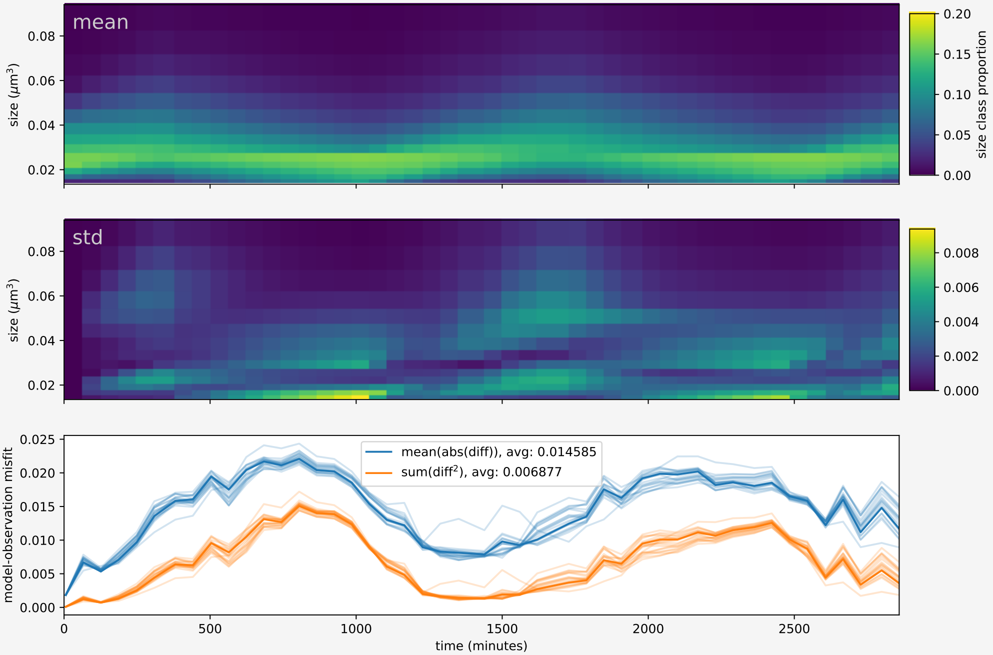
division: with size-dependent division rate $\delta_i(\Delta_t) = \Delta_t\, \delta_{max}\, \frac{v_i^b}{1+v_i^b}$ (2 parameters)

stasis: the rest of the cells remain at their size class

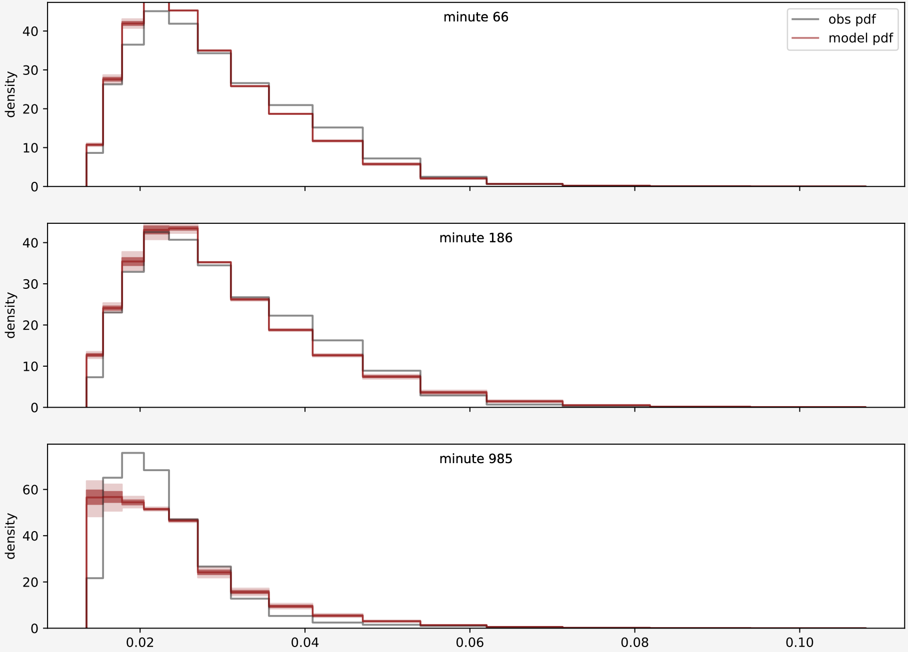
$$
\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & & & j-1 & j & & m-1 & m \\ \downarrow & & & \downarrow & \downarrow & & \downarrow & \downarrow \end{array} \\
A(t,\Delta_t) = & \left(\begin{array}{cccccccc}
(1-\gamma) & & & & 2\delta_j & & & \\
\gamma & \ddots & & & & \ddots & & \\
 & \ddots & \ddots & & & & \ddots & \\
 & & \ddots & (1-\gamma) & & & & 2\delta_m \\
 & & & \gamma & (1-\delta_j)(1-\gamma) & & & \\
 & & & & (1-\delta_j)\gamma & \ddots & & \\
 & & & & & \ddots & (1-\delta_{m-1})(1-\gamma) & \\
 & & & & & & (1-\delta_{m-1})\gamma & 1-\delta_m
\end{array}\right)
\begin{array}{l} \leftarrow 1 \\ \leftarrow 2 \\ \\ \\ \\ \\ \leftarrow m-1 \\ \leftarrow m \end{array}
\end{array}
$$

- So far the plots only show the posterior mean, add some plots that emphasize that it is a posterior distribution.

exercise


mean
std
size (μm^3)
0.08
0.06
0.04
0.02
size class proportion
0.20
0.15
0.10
0.05
0.00
0.008
0.006
0.004
0.002
0.000
model-observation misfit
0.025
0.020
0.015
0.010
0.005
0.000
mean(abs(diff)), avg: 0.014585
sum(diff^2), avg: 0.006877
0
500
1000
1500
2000
2500
time (minutes)

exercise



minute 66
obs pdf
model pdf
density
0
10
20
30
40
minute 186
density
0
10
20
30
40
minute 985
density
0
20
40
60
0.02
0.04
0.06
0.08
0.10

some ideas

1. adding respiration
2. making growth size-dependent
3. size-dependent respiration
4. a better way to compare model an observation
5. multiple populations

idea: adding respiration

- let cells "shrink" (move down one size class) at a specific rate

$$
A = \begin{array}{c}
\begin{array}{cccccccc}
1 & 2 & & j-1 & j & & m-1 & m \\
\downarrow & \downarrow & & \downarrow & \downarrow & & \downarrow & \downarrow
\end{array} \\
\left(\begin{array}{cccccccc}
(1-\gamma_1) & \rho & & & 2\delta_j & & & \\
\gamma_1(1-\rho) & (1-\gamma_2)(1-\rho) & \ddots & & & \ddots & & \\
 & \ddots & \ddots & \rho & & & \ddots & \\
 & & \ddots & (1-\gamma_{j-1})(1-\rho) & (1-\delta_j)\rho & & & 2\delta_m \\
 & & & \gamma_{j-1}(1-\rho) & (1-\delta_j)(1-\gamma_j)(1-\rho) & \ddots & & \\
 & & & & (1-\delta_j)\gamma_j(1-\rho) & \ddots & \ddots & \\
 & & & & & \ddots & (1-\delta_{m-1})(1-\gamma_{m-1})(1-\rho) & (1-\delta_m)\rho \\
 & & & & & & (1-\delta_{m-1})\gamma_{m-1}(1-\rho) & 1-\delta_m(1-\rho)
\end{array}\right)
\begin{array}{l}
\leftarrow 1 \\ \leftarrow 2 \\ \\ \\ \\ \\ \leftarrow m-1 \\ \leftarrow m
\end{array}
\end{array}
$$

division: with size-dependent division rate δ_i

respiration: with size-independent respiration rate ρ

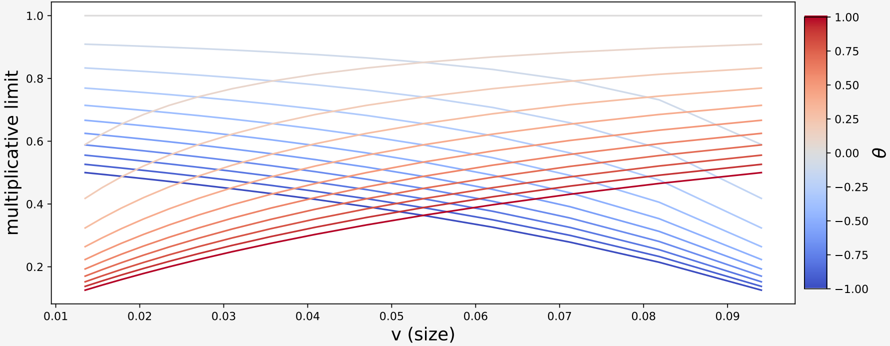
growth: with growth rate γ

stasis: the rest of the cells remain at their size class

idea: making growth size-dependent

- turn the γ parameter into γ_i
- an approach for data-driven size-dependence:

$$\text{“size limit”} = \begin{cases} \frac{v_i/v_m}{\theta + v_i/v_m} & \text{if } \theta > 0 \\ \frac{1+\frac{v_{\min}-v_i}{v_m}}{1+\frac{v_{\min}-v_i}{v_m}-\theta} & \text{otherwise} \end{cases} \qquad \text{for } \theta \in [-1, 1]$$



idea: size-dependent respiration

- combine the previous 2 ideas, to add size-dependence to the respiration rate

idea: a better way to compare model and observations

- find a better way to compare or model with observations

```
// fitting observations
for (it in 1:nt_obs){
    diff = 0.0;
    for (iv in 1:m){
        diff += fabs(mod_obspos[iv,it] - obs[iv,it]);
    }
    diff = diff/sigma;
    diff ~ normal(0.0, 1.0) T[0,];
}
```

- the standard, even more naive approach leads to significantly worse results

```
// fitting observations
for (it in 1:nt_obs){
    for (iv in 1:m){
        obs[iv,it] ~ normal(mod_obspos[iv,it], sigma);
    }
}
```

- in a follow-up paper, Hunter-Cevera et al. (2014) “Diel size distributions reveal seasonal growth dynamics of a coastal phytoplankter”, the authors use 2 subpopulations to model the data
- it is easy to add an `npop` parameter and use an iteration

turning

```
for (it in 1:nt){ // time-stepping loop
    // compute gamma
    gamma = gamma_max * dt_days * (1.0 - exp(-E[it]/E_star));
    ...
    for (i in 1:m){ // size-class loop
        ...
```

into

```
for (ipop in 1:npop){
    ...
    for (it in 1:nt){ // time-stepping loop
        // compute gamma
        gamma = gamma_max[ipop] * dt_days * (1.0 - exp(-E[it]/E_star[ipop]));
        ...
        for (i in 1:m){ // size-class loop
            ...
```