Checks and puzzles

Owen Petchey

6/25/2021

Setup

R

```
# rm(list = ls())

knitr::opts_knit$set(
  progress = TRUE,
  verbose = FALSE,
  cache = TRUE
)

microxanox_release <- "0.3.0"

#tmplib <- tempfile()
#dir.create(tmplib)


### From '?remotes::install_github`:
# auth_token
#   To install from a private repo, generate a personal access token (PAT) in
#   "https://github.com/settings/tokens" and supply to this argument. This is
#   safer than using a password because you can easily delete a PAT without
#   affecting any others. Defaults to the GITHUB_PAT environment variable.

# remotes::install_github(
#   "opetchey/microxanox",
#   ref = microxanox_release,
#   # auth_token = "ENTER YOUR TOKEN or PROVED AS ENVIRONMENT VARIABLE",
#   build_vignettes = FALSE,
#   force = TRUE,
#   upgrade = FALSE,
#   lib = tmplib
# )

#library(microxanox, lib.loc = tmplib)

library(microxanox)
if (packageVersion("microxanox") < package_version("0.3.0")) {
```

```r
  stop("microxanox version needs to be at least 0.3.0!")
}
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(patchwork)
library(here)
```

```
## here() starts at /Users/owenpetchey/Desktop/microxanox/diversity_envresp1
```

```r
source(here("R/various_useful_functions.r"))
zero <- 0 ## don't change
unity <- 1 ## don't change!!!
options(mc.cores = 7)
eval_dynamics_flag <- FALSE
```

Version of `microxanox` package used: 0.3.0

General simulation conditions

```r
num_CB_strains <- 9
num_SB_strains <- 9
num_PB_strains <- 9

sp <- new_strain_parameter(
  n_CB = num_CB_strains,
  n_PB = num_SB_strains,
  n_SB = num_PB_strains,
  values_initial_state = "bush_ssfig3"
)

parameter <- new_runsim_parameter(
  dynamic_model = bushplus_dynamic_model,
  event_definition = event_definition_1,
  event_interval = 100,
  noise_sigma = 0,
  minimum_abundances = rep(1, 3),
  sim_duration = 2000,
  sim_sample_interval = 100,
  strain_parameter = sp,
  log10a_series = c(
    log10(sp$a_O),
    log10(sp$a_O)
  )
)
names(parameter$minimum_abundances) <- c("CB", "PB", "SB")
parameter_master <- parameter
rm(sp)
```

Define diversity

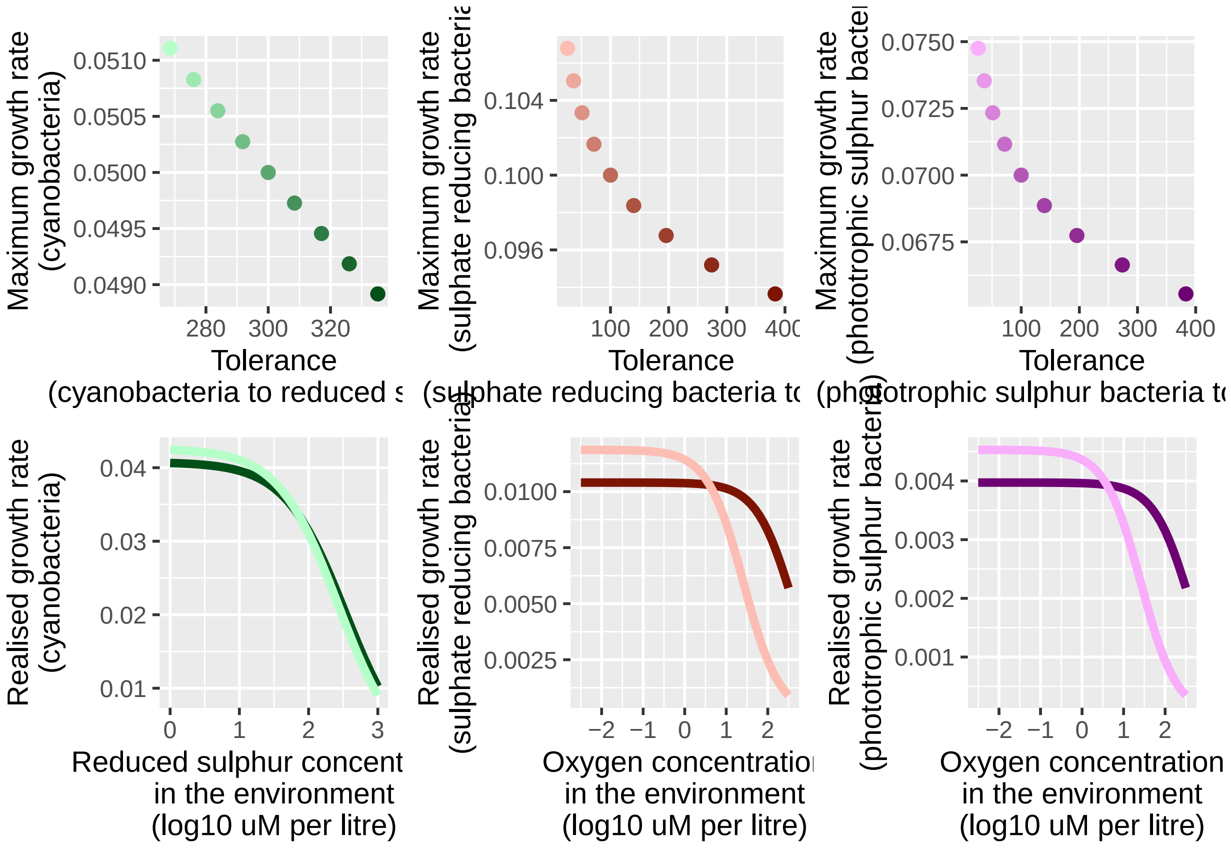
```
## multiplier of SBPB variation
CB_var_multiplier <- 2
SBPB_var_multiplier <- 6

CB_gmax_div <- 0.015789474 * CB_var_multiplier
CB_h_div <- -0.08 * CB_var_multiplier
SB_gmax_div <- 0.015789474 * SBPB_var_multiplier
SB_h_div <- -0.323  * SBPB_var_multiplier
PB_gmax_div <- 0.015789474  * SBPB_var_multiplier
PB_h_div <- -0.323  * SBPB_var_multiplier

num_div_treatment_levels <- 5
```

Create diversity

```
var_expt <- create_diversity_factorial(
  zero = zero, unity = unity,
  num_div_treatment_levels = num_div_treatment_levels,
  CB_gmax_div = CB_gmax_div, CB_h_div = CB_h_div,
  SB_gmax_div = SB_gmax_div, SB_h_div = SB_h_div,
  PB_gmax_div = PB_gmax_div, PB_h_div = PB_h_div,
  default_9strain = new_strain_parameter(
    n_CB = num_CB_strains,
    n_SB = num_SB_strains,
    n_PB = num_PB_strains,
    values_initial_state =  "bush_ssfig3"
  )
)
```

Display diversity

```
display_diversity(
  nrow(var_expt),
  var_expt = var_expt,
  num_CB_strains = num_CB_strains,
  num_SB_strains = num_SB_strains,
  num_PB_strains = num_PB_strains
)
```



```
## Here we get some rows of var_expt that correspond with particular diversity levels

var_expt_levels <- var_expt[, 1:6]

no_diversity <- which(rowSums(abs(var_expt_levels)) == 0)
max_diversty_all <- which(
  max(rowSums(abs(var_expt_levels))) == rowSums(abs(var_expt_levels))
)
max_only_CB_diversity <- which(
  max(rowSums(abs(var_expt_levels[, 1:2]))) == rowSums(abs(var_expt_levels[, 1:2])) &
  rowSums(abs(var_expt_levels[,3:6])) == 0
)
# var_expt_levels[381,]

max_only_SBPB_diversity <- which(
  max(rowSums(abs(var_expt_levels[, 3:6]))) == rowSums(abs(var_expt_levels[, 3:6])) &
  rowSums(abs(var_expt_levels[, 1:2])) == 0
)
#var_expt_levels[20,]

#medium_diverity_varexp_row <- 311
```

What effect of changing the length of the simulations

Setup and processing

Run a shorter sim:

```r
sim_length <- 100000

options(mc.cores = 7)

minimum_abundances <- rep(0, 3)
names(minimum_abundances) <- c("CB", "PB", "SB")

grid_num_a <- 1000 #usually 1000 ## number of a_0 values
#grid_num_a <- 10 ## FOR TEST
a_0s <- 10^seq(-7, -0.5, length=grid_num_a) ## sequence of a_0 values
grid_num_N <- 2 ## number of N values
initial_CBs <- 10^seq(0, 10, length=grid_num_N) ## sequence of N values
initial_PBs <- 1e8 ## not varied
initial_SBs <- 1e8 ## not varied
# next line creates all possible combinations
ss_expt <- expand.grid(N_CB = initial_CBs,
                       N_PB = initial_PBs,
                       N_SB = initial_SBs,
                       a_0 = a_0s)

parameter <- new_ss_by_a_N_parameter(
  dynamic_model = parameter$dynamic_model,
  event_definition = parameter$event_definition,
  event_interval = sim_length,
  noise_sigma = parameter$noise_sigma,
  minimum_abundances = minimum_abundances,
  sim_duration = sim_length,  ##    Was 1e6
  sim_sample_interval = sim_length,
  log10a_series = parameter$log10a_series,
  solver_method = parameter$solver_method,
  ss_expt = ss_expt
)
rm(minimum_abundances, grid_num_a, a_0s, grid_num_N, initial_CBs, initial_PBs, initial_SBs, ss_expt)

saveRDS(parameter, here("experiments/checks_puzzles/data/parameter_1e5_x2x6_factorial.RDS"))
saveRDS(var_expt, here("experiments/checks_puzzles/data/var_expt_1e5_x2x6_factorial.RDS"))
```

*Careful, this simulation takes about 600 hours on a single core

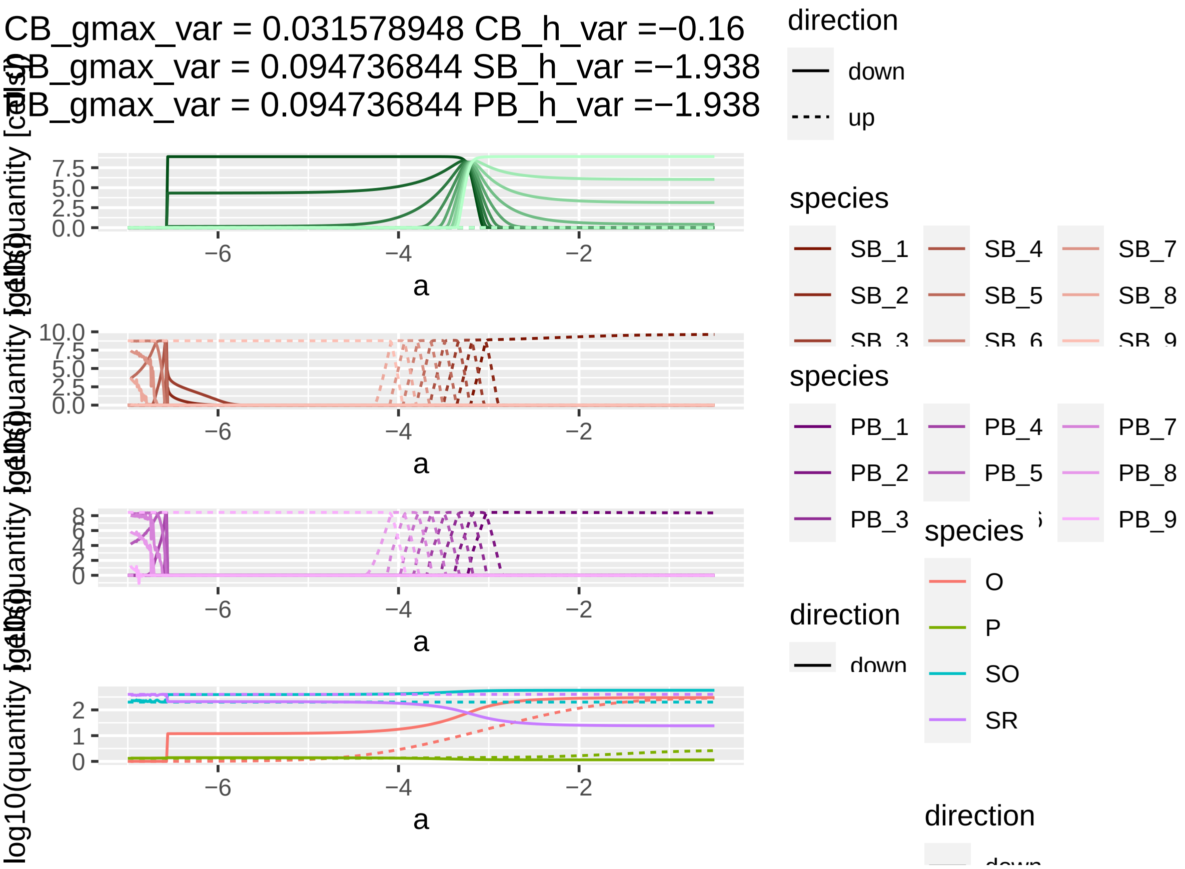
```r
run_ss_var_experiment(
  parameter = readRDS(here("experiments/checks_puzzles/data/parameter_1e5_x2x6_factorial.RDS")),
  var_expt = readRDS(here("experiments/checks_puzzles/data/var_expt_1e5_x2x6_factorial.RDS"))) %>%
  saveRDS(here("experiments/checks_puzzles/data/ss_data_1e5_x2x6_factorial.RDS"))
```

Bring in various stable state datasets

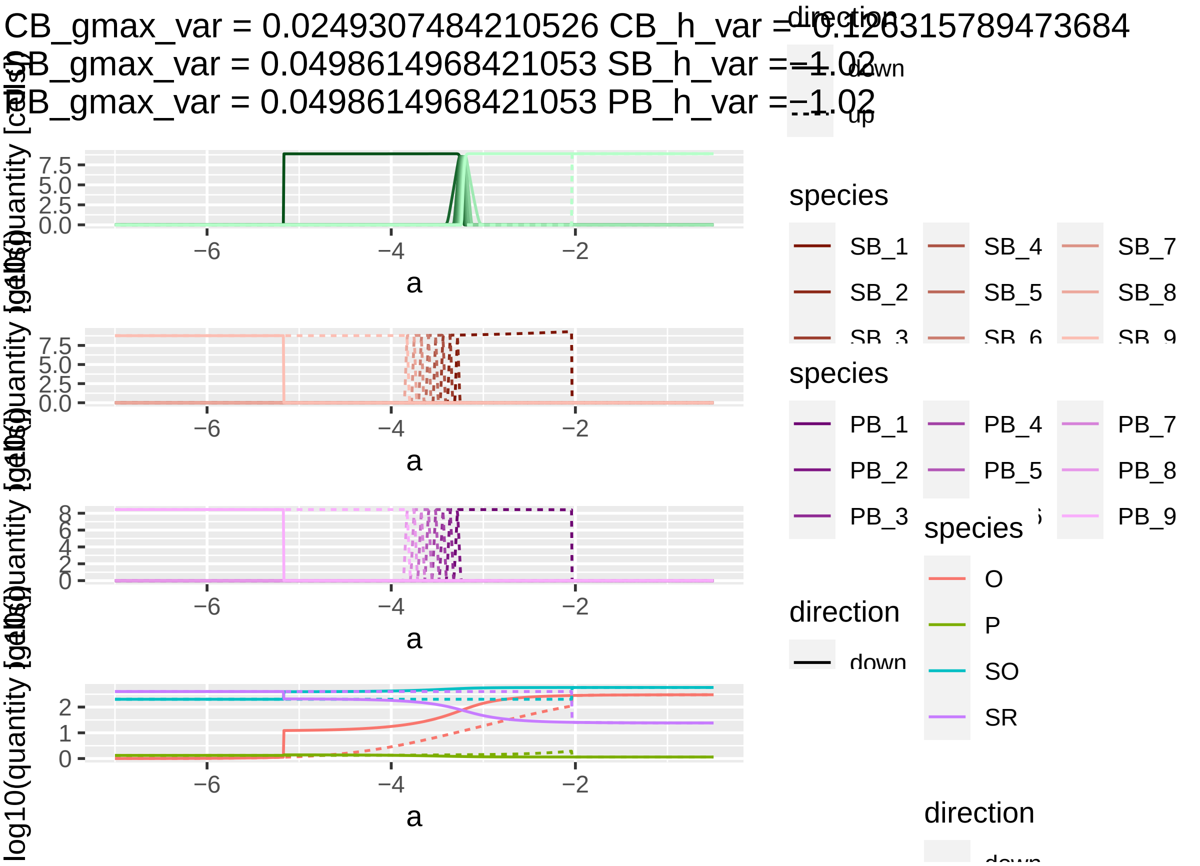
```
cluster <- multidplyr::new_cluster(7)
multidplyr::cluster_library(cluster, c("microxanox", "dplyr"))

## sim length 1'000'000, 20 SBPBgrad, 2xCB variation, 6xSBPB variation
readRDS(here("experiments/checks_puzzles/data/ss_data_1e5_x2x6_factorial.RDS")) %>%
  mutate(sim_length = 100000) %>%
  multidplyr::partition(cluster) %>%
  mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
  collect() %>%
  unnest(cols = c(stability_measures)) %>%
  saveRDS(here("experiments/checks_puzzles/data/stab_data_1e5_x2x6_factorial.RDS"))
```

Visualise

```
var_expt1 <- readRDS(here("experiments/9_strains/data/ss_data_1e6_x2x6_factorial.RDS"))
var_expt2 <- readRDS(here("experiments/checks_puzzles/data/ss_data_1e5_x2x6_factorial.RDS"))

p1_short  <- plot_ss_result1(var_expt2,
                result_index = nrow(var_expt2),
                filename_prefix = NULL,
                save_image_file = FALSE)
p1_short
```



```
p1_long  <- plot_ss_result1(var_expt1,
                result_index = nrow(var_expt1),
                filename_prefix = NULL,
                save_image_file = FALSE)
p1_long
```



Continute here to check the stability measures.

Hidden effect of sulphur bacteria diversity

When the environment ameliorates for the sulphur bacteria, there is no strain replacement (in final state) along the oxygen diffusivity gradient – either the system is oxic or the least tolerant strain SB9 dominates. And yet the switch to anoxic occurs earlier than when there is no diversity, which suggests there is some role of the more tolerant strains. Indeed Uriah showed that the presence of only the most tolerant strain is sufficient to give an earlier switch, even though it is not present in the final state when less tolerant strains are present. And he showed that the presence of only the least tolerant strain creates a later switch than when there are more tolerant strains.

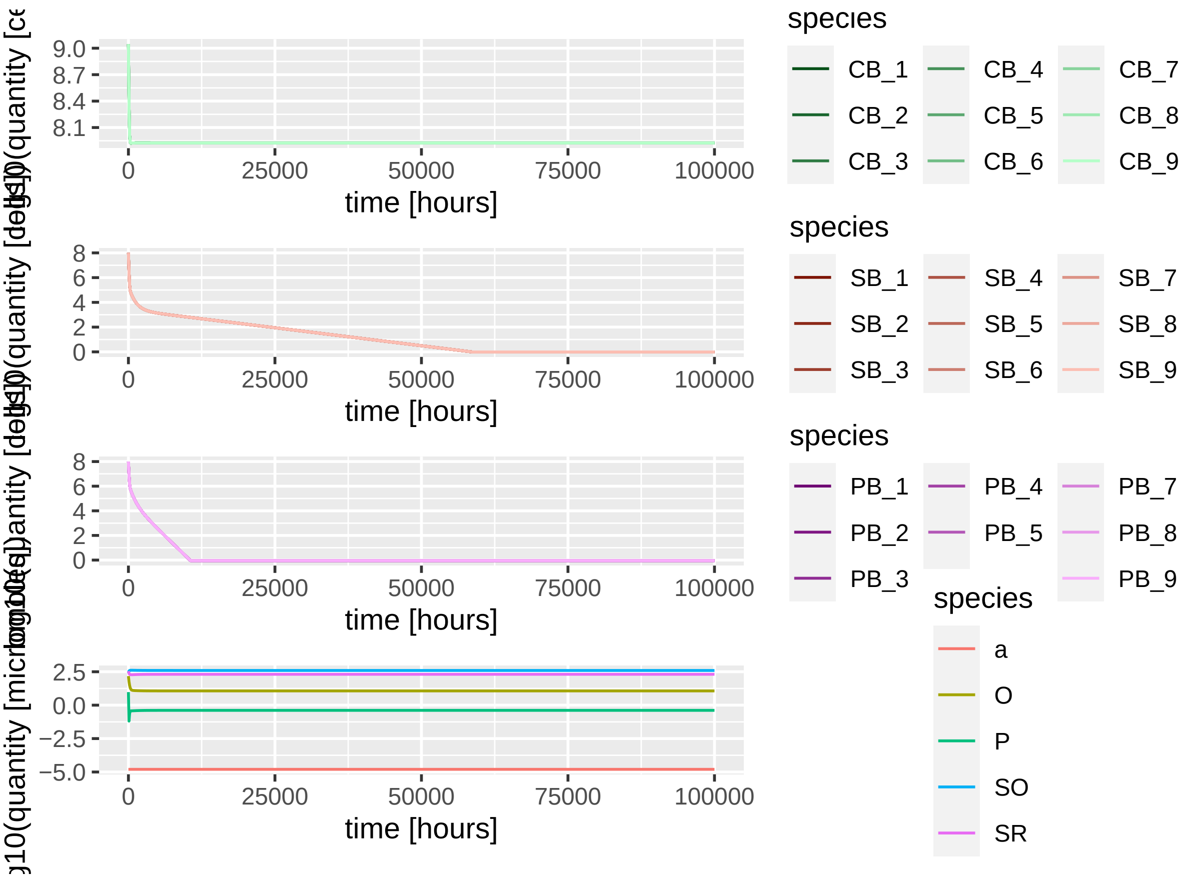
The explanation is that the most tolerant strain does play a role, but only a transient one. The following dynamics are for the system starting oxic, and with a value of oxygen diffusivity (log10(a_O) = -4.8) for which the system remains oxic if there is no diversity, but switches to anoxic if there is diversity. There is only diversity in the sulphur bacteria. The most tolerant strain does at first grow, but is then outcompeted by less tolerant strains as the environment ameliorates (temporally).

TODO CHECK THE PARAMETER

```
#var_expt <- readRDS(here("experiments/9_strains/data/ss_data_1e6_x2x6_factorial.RDS"))
var_expt_levels <- var_expt[,1:6]
no_diversity <- which(rowSums(abs(var_expt_levels))==0)
max_diversty_all <- which(max(rowSums(abs(var_expt_levels))) ==
                            rowSums(abs(var_expt_levels)))
max_only_CB_diversity <- which(max(rowSums(abs(var_expt_levels[,1:2]))) ==
                            rowSums(abs(var_expt_levels[,1:2])) &
                              rowSums(abs(var_expt_levels[,3:6]))==0)
max_only_SBPB_diversity <- which(max(rowSums(abs(var_expt_levels[,3:6]))) ==
                            rowSums(abs(var_expt_levels[,3:6])) &
                              rowSums(abs(var_expt_levels[,1:2]))==0)
```

```
parameter <- parameter_master
parameter$strain_parameter <- var_expt[no_diversity,]$pars[[1]]
parameter$sim_duration <- 100000
parameter$result <- NULL
parameter$log10a_series <- c(-4.8, -4.8)
parameter$strain_parameter$initial_state <- new_initial_state(
  nrow(parameter$strain_parameter$CB),
  nrow(parameter$strain_parameter$PB),
  nrow(parameter$strain_parameter$PB),
  values = "bush_ssfig3"
)
parameter$strain_parameter$initial_state[grep("CB_", names(parameter$strain_parameter$initial_state))] <


sim_res_novar1 <- run_simulation(parameter)
saveRDS(sim_res_novar1, here("experiments/checks_puzzles/data/puzzle1_1.RDS"))
```

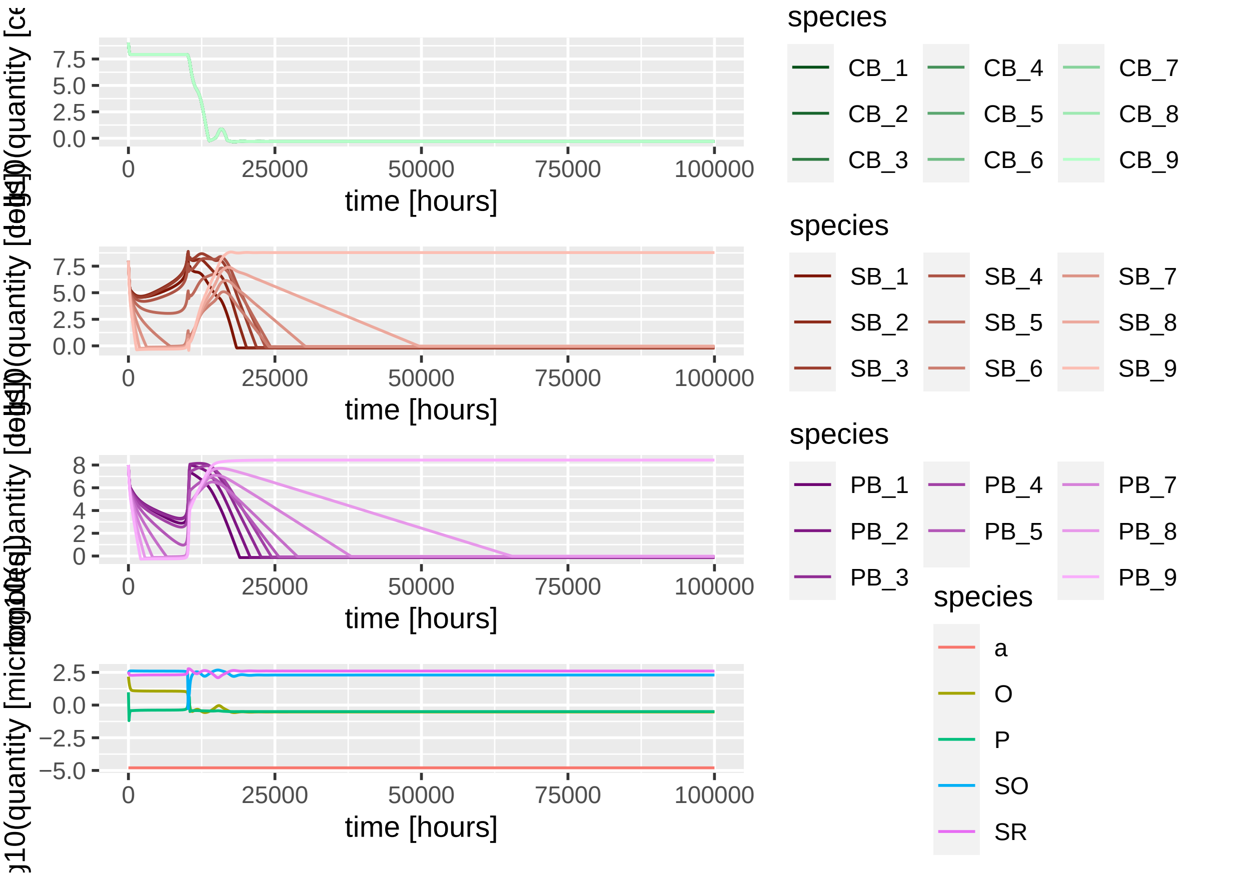
```
sim_res_novar1 <- readRDS(here("experiments/checks_puzzles/data/puzzle1_1.RDS"))
plot_dynamics(sim_res_novar1)
```



```
#ggsave(here("simulations/expt2/figures/switching_novar.pdf"), width = 10)
```

```
parameter$strain_parameter <- var_expt[max_only_SBPB_diversity,]$pars[[1]]
#parameter$sim_duration <- 100000
#parameter$result <- NULL
#parameter$log10a_series <- c(-5.2, -5.2)
#parameter$strain_parameter$initial_state <- new_initial_state(
#  nrow(parameter$strain_parameter$CB),
#  nrow(parameter$strain_parameter$PB),
#  nrow(parameter$strain_parameter$PB),
#  values = "bush_ssfig3"
#)
parameter$strain_parameter$initial_state[grep("CB_", names(parameter$strain_parameter$initial_state))]
sim_res_highvar1 <- run_simulation(parameter)
saveRDS(sim_res_highvar1, here("experiments/checks_puzzles/data/puzzle1_2.RDS"))
```

```
sim_res_highvar1 <- readRDS(here("experiments/checks_puzzles/data/puzzle1_2.RDS"))
plot_dynamics(sim_res_highvar1)
```



```
#ggsave(here("simulations/expt2/figures/switching_novar.pdf"), width = 10)
```

Zoom in on SS

Negative abundance investigation

I (Owen) found that the sampling interval had an effect on the stability of the simulation. If the sampling interval was long, then in some rare cases (see below) the odesolver failed, with negative abundances occuring. I think this is due to abundances becoming very small, and then the computer having trouble with precision. I guess that when a sample is taken, the abundance is somehow altered if it is very low, probably by some rounding.

Understand about relative and absolute variation in traits