

experiment 1

Owen Petchey

6/25/2021

This experiment supersedes all previous ones. It is a factorial manipulation of diversity of the three groups. It takes about 50 hours to run while using 12 cores.

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")) {
  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 3.1.6      v dplyr 1.0.7
## v tidyr 1.1.4       v stringr 1.4.0
## v readr 2.1.0       v forcats 0.5.1

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

library(patchwork)
library(here)

## here() starts at /Users/rainerkrug/Documents_Local/git/diversity_envresp1
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

```

# default_dynamic_model <- bushplus_dynamic_model
# default_event_definition <- event_definition_1
# default_event_interval <- 100
# default_noise_sigma <- 0
# default_minimum_abundances <- rep(1, 3)
# names(default_minimum_abundances) <- c("CB", "PB", "SB")
# default_sim_duration <- 80000
# default_sim_sample_interval <- 100
# initial_pars_from <- "bush_ssfig3"
## note that next line (log10a_series is over-ridden with getting stable states)
#default_log10a_series <- c(-2, -2, -2, -2, -10, -10, -10, -10, -10)

num_CB_strains <- 9
num_SB_strains <- 9
num_PB_strains <- 9

sp <- new_strain_parameter(
  n_CB = 9,
  n_PB = 9,
  n_SB = 9,

  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$a_0),
  log10(sp$a_0)
)
)
names(parameter$minimum_abundances) <- c("CB", "PB", "SB")
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 <- 20

## num_div_treatment_levels <- 3 ## FOR TEST

```

Create diversity

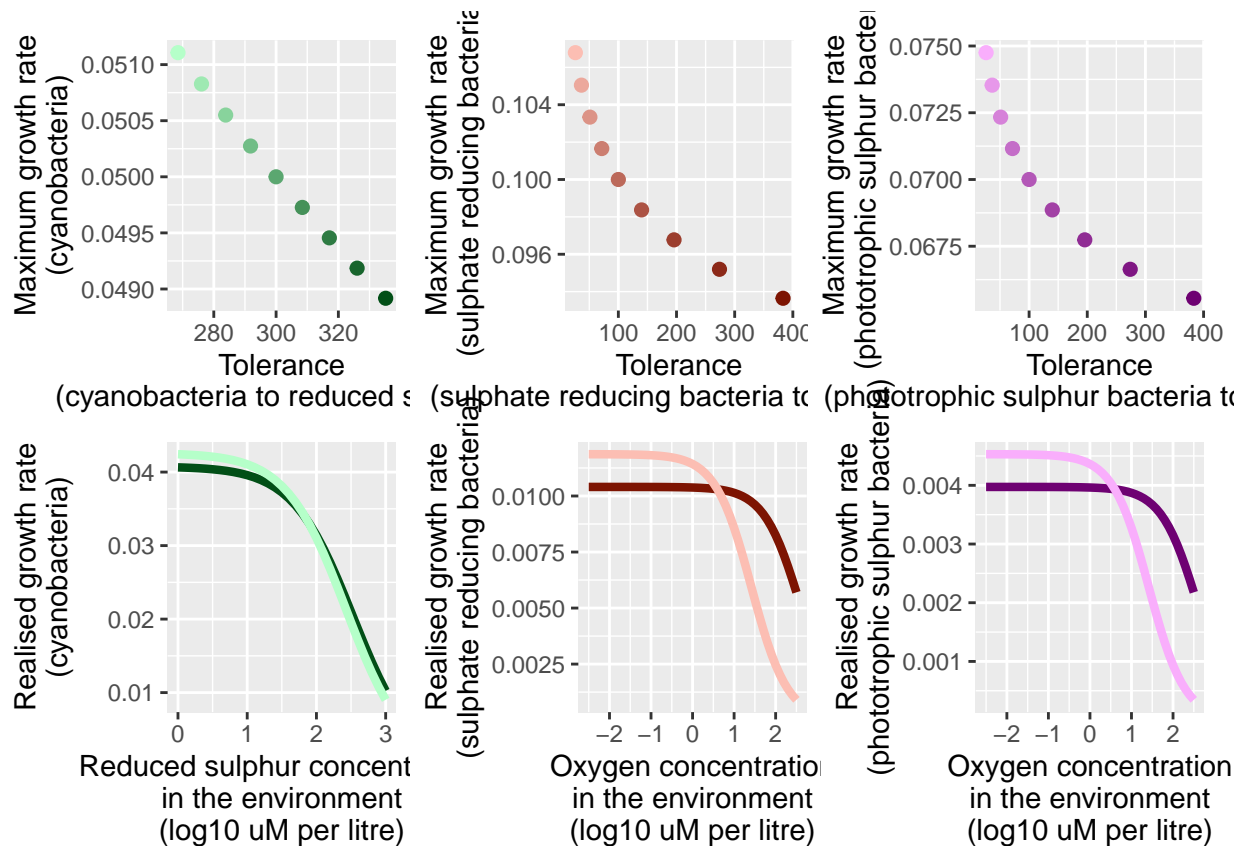
```

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(
  400,
  var_expt = var_expt,
  num_CB_strains = num_CB_strains,
  num_SB_strains = num_SB_strains,
  num_PB_strains = num_PB_strains
)
```



Temporal switching

```
var_expt_levels <- var_expt[, 1:6]

no_diversity <- which(rowSums(abs(var_expt_levels)) == 0)
max_diversity_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,]

```

Oxic to anoxic

No diversity

```

# default_sim_duration <- 80000
parameter$sim_duration <- 80000

```

```

# default_log10a_series <- c(-1, -7, -7)
parameter$log10a_series <- c(-1, -7, -7)
parameter$strain_parameter <- var_expt$pars[[no_diversity]]
parameter$strain_parameter$initial_state <- new_initial_state(
  num_CB_strains,
  num_PB_strains,
  num_SB_strains,
  values = "bush_ssf3"
)

```

```

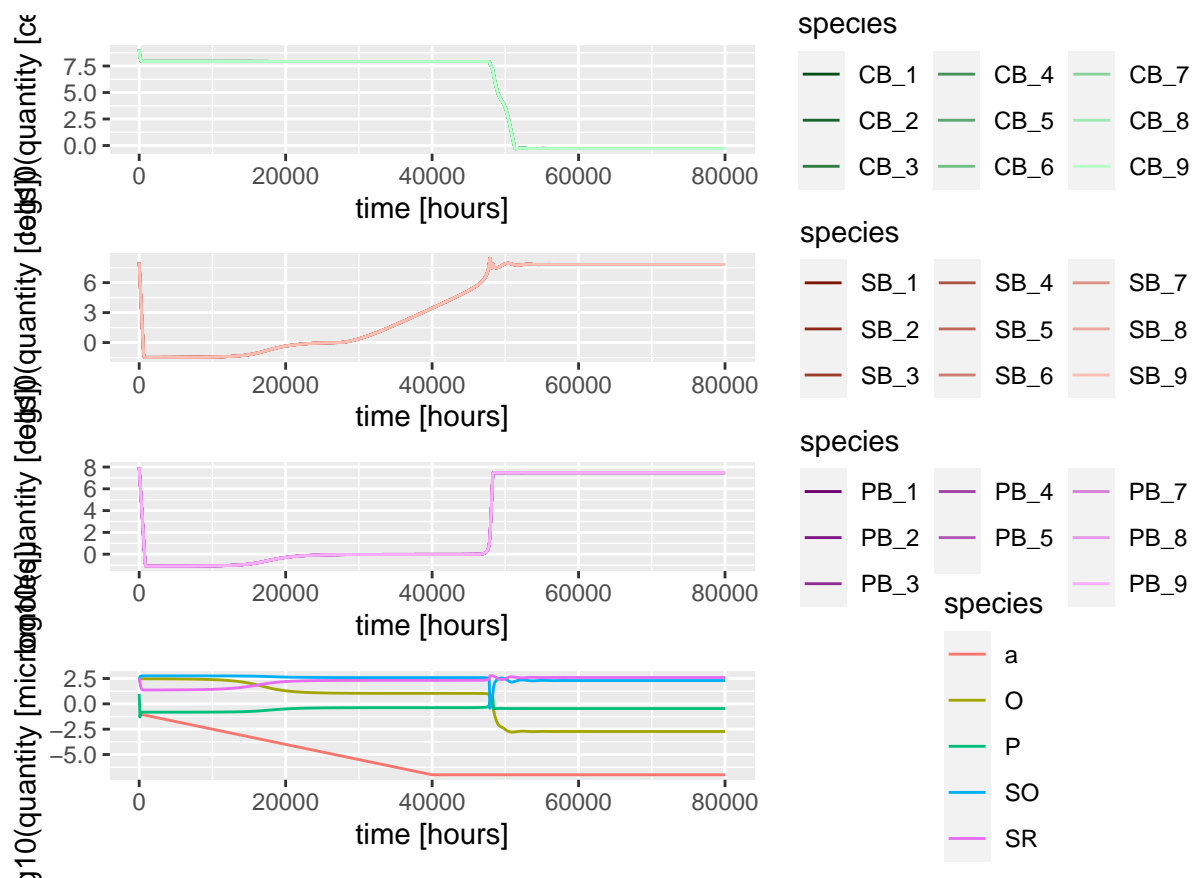
parameter$strain_parameter$initial_state[grepl("CB_", names(parameter$strain_parameter$initial_state))]
sim_res_novar1 <- run_simulation(parameter)
saveRDS(sim_res_novar1, here("experiments/experiment 1/data/sim_res_novar1.RDS"))

```

```

sim_res_novar1 <- readRDS(here("experiments/experiment 1/data/sim_res_novar1.RDS"))
plot_dynamics(sim_res_novar1)

```

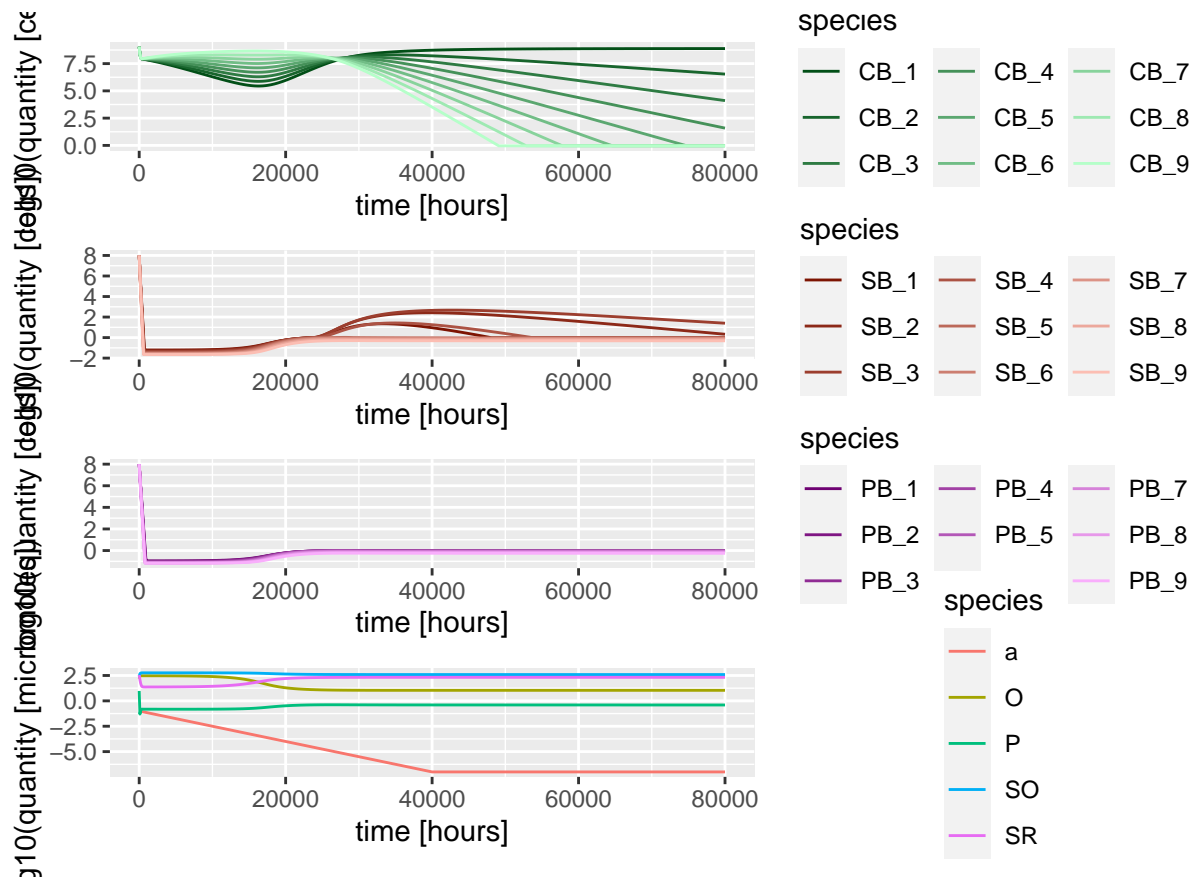


```
#ggsave(here("simulations/expt2/figures/switching_novar.pdf"), width = 10)
```

With diversity

```
# Is this actually needed?
# sim_number <- num_div_treatment_levels
try_me <- 311
#sim_number2 <- num_div_treatment_levels
parameter$strain_parameter <- var_expt$pars[[try_me]]
#parameter$strain_parameter <- var_expt$pars[[max_diversty_all]]
parameter$strain_parameter$initial_state <- sim_res_novar1$strain_parameter$initial_state
sim_res_highvar1 <- run_simulation(parameter)
saveRDS(sim_res_highvar1, here("experiments/experiment 1/data/sim_res_highvar1.RDS"))

sim_res_highvar1 <- readRDS(here("experiments/experiment 1/data/sim_res_highvar1.RDS"))
plot_dynamics(sim_res_highvar1)
```



```
#ggsave(here("simulationsexpt2/figures/switching_highvar.pdf"), width = 10)
```

Anoxic to oxic

No diversity

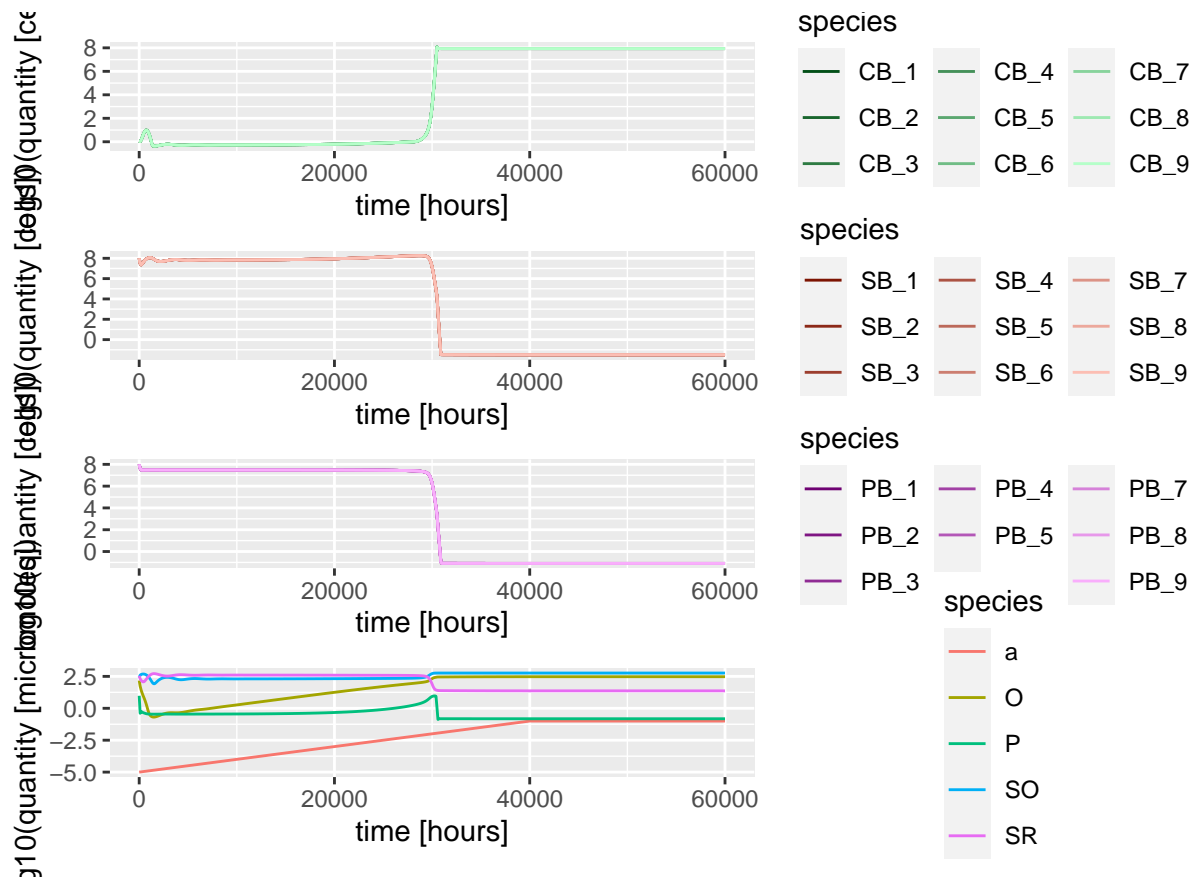
```
parameter$sim_duration <- 60000
```

```

# sim_number <- 1
parameter$log10a_series <- c(-5, -3, -1, -1)
parameter$strain_parameter <- var_expt$pars[[no_diversity]]
parameter$strain_parameter$initial_state <- new_initial_state(
  num_CB_strains,
  num_PB_strains,
  num_SB_strains,
  values = "bush_ssf3"
)
parameter$strain_parameter$initial_state[grepl("CB_", names(parameter$strain_parameter$initial_state))]
sim_res_novar2 <- run_simulation(parameter)
saveRDS(sim_res_novar2, here("experiments/experiment 1/data/sim_res_novar2.RDS"))

sim_res_novar2 <- readRDS(here("experiments/experiment 1/data/sim_res_novar2.RDS"))
plot_dynamics(sim_res_novar2)

```



```

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

```

With diversity

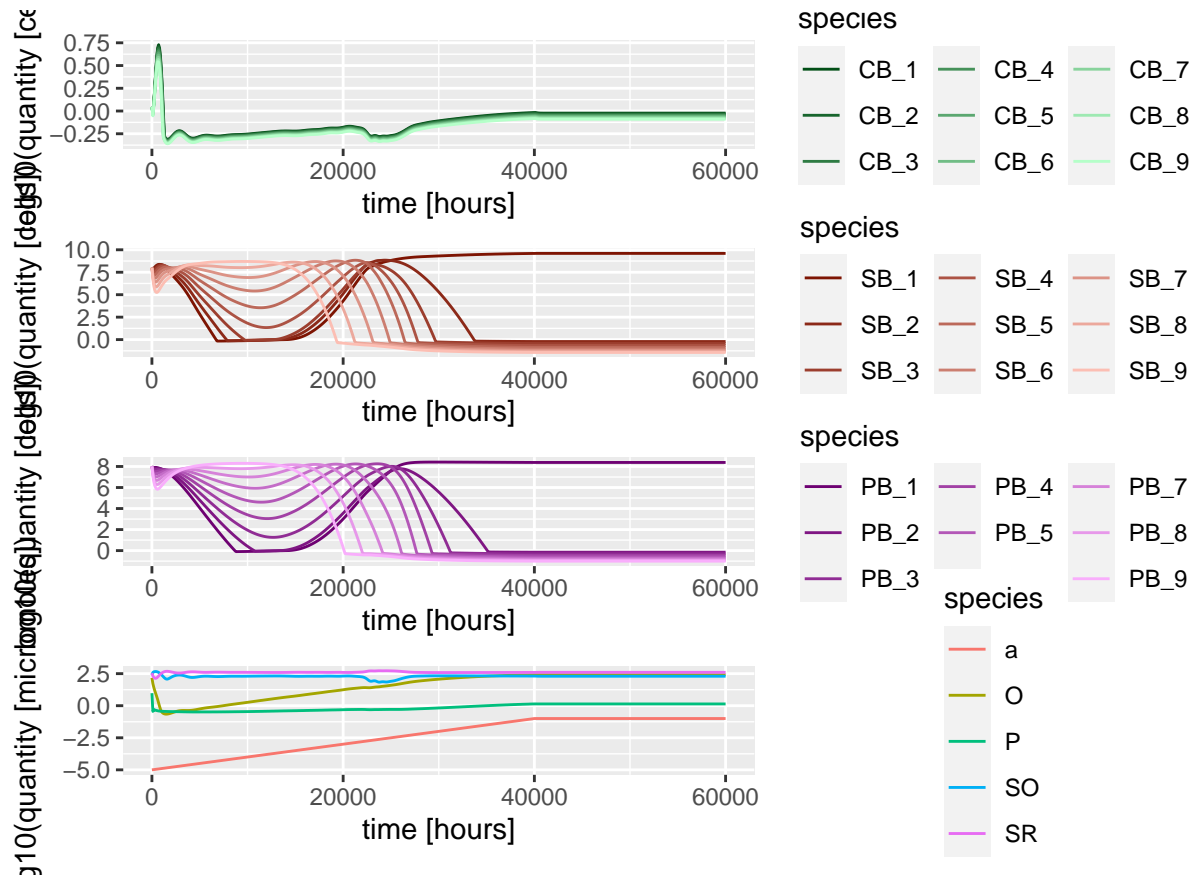
```

#sim_number <- num_div_treatment_levels
try_me <- 311
#sim_number2 <- num_div_treatment_levels
parameter$strain_parameter <- var_expt$pars[[try_me]]
#parameter$strain_parameter <- var_expt$pars[[max_diversity_all]]
parameter$strain_parameter$initial_state <- sim_res_novar2$strain_parameter$initial_state

```

```
sim_res_highvar2 <- run_simulation(parameter)
saveRDS(sim_res_highvar2, here("experiments/experiment 1/data/sim_res_highvar2.RDS"))

sim_res_highvar2 <- readRDS(here("experiments/experiment 1/data/sim_res_highvar2.RDS"))
plot_dynamics(sim_res_highvar2)
```



```
#ggsave(here("simulationsext2/figures/switching_highvar.pdf"), width = 10)
```

Anoxic to oxitic to anoxic

No diversity

```
parameter$sim_duration <- 1000000
```

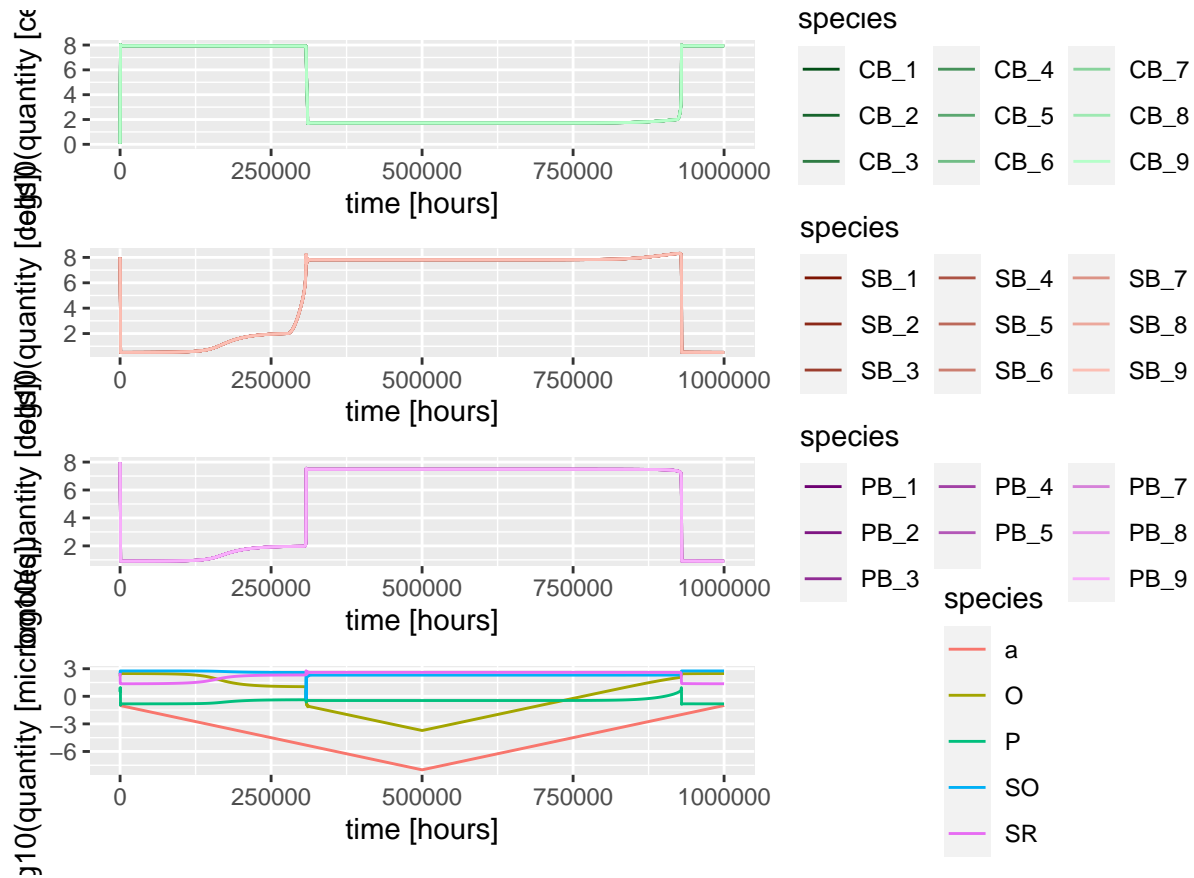
```
parameter$minimum_abundances <- rep(100, 3)
names(parameter$minimum_abundances) <- c("CB", "PB", "SB")
```

```
# sim_number1 <- 1
parameter$log10a_series <- c(-1, -8, -1)
parameter$strain_parameter <- var_expt$pars[[no_diversity]]
parameter$strain_parameter$initial_state <- new_initial_state(
  num_CB_strains,
  num_PB_strains,
  num_SB_strains,
  values = "bush_ssfig3"
)
parameter$strain_parameter$initial_state[grep("CB_", names(parameter$strain_parameter$initial_state))]
```



```
sim_res_novar3 <- run_simulation(parameter)
saveRDS(sim_res_novar3, here("experiments/experiment 1/data/sim_res_novar3.RDS"))
```

```
sim_res_novar3 <- readRDS(here("experiments/experiment 1/data/sim_res_novar3.RDS"))
plot_dynamics(sim_res_novar3)
```

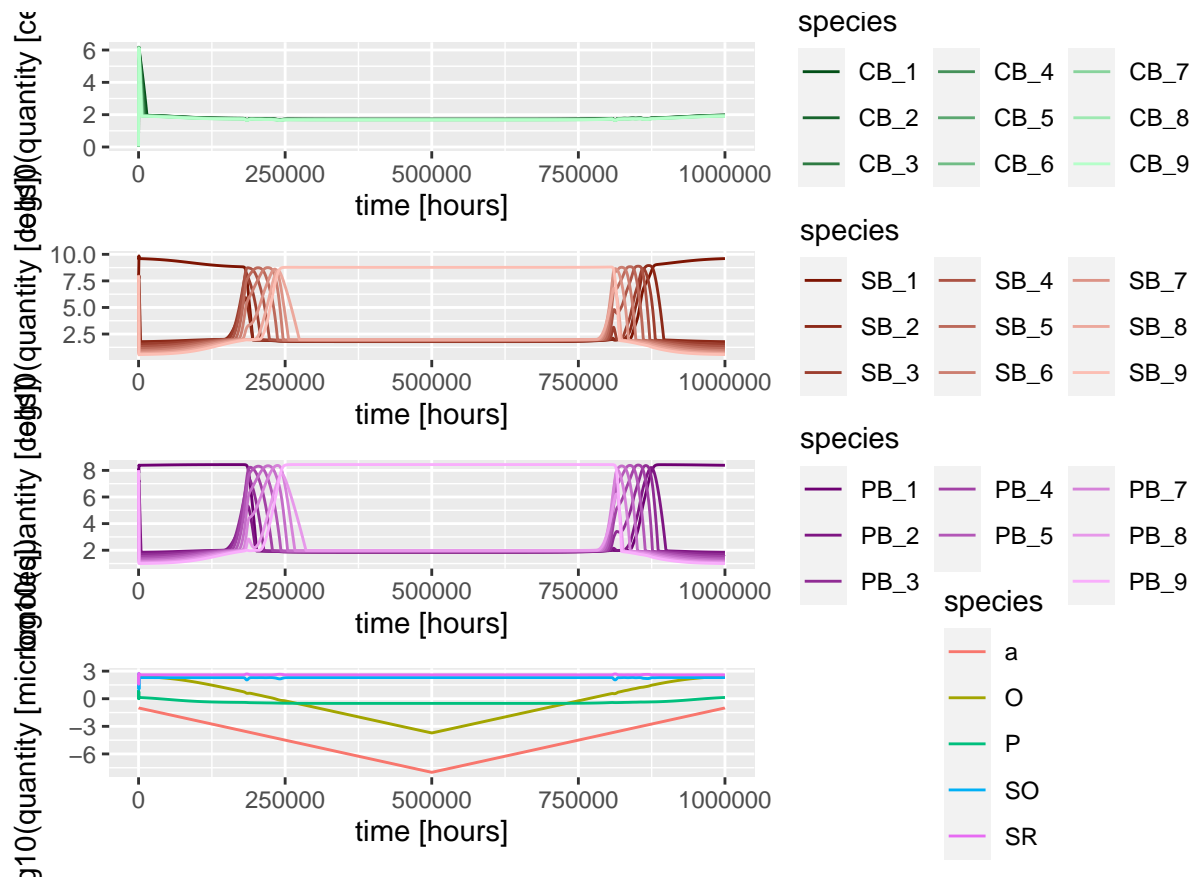


```
#ggsave(here("simulations/expt2/figures/switching_novar.pdf"), width = 10)
```

With diversity

```
try_me <- 311
#sim_number2 <- num_div_treatment_levels
parameter$strain_parameter <- var_expt$pars[[try_me]]
parameter$strain_parameter$initial_state <- sim_res_novar3$strain_parameter$initial_state
sim_res_highvar3 <- run_simulation(parameter)
saveRDS(sim_res_highvar3, here("experiments/experiment 1/data/sim_res_highvar3.RDS"))
```

```
sim_res_highvar3 <- readRDS(here("experiments/experiment 1/data/sim_res_highvar3.RDS"))
plot_dynamics(sim_res_highvar3)
```



```
#ggsave(here("simulationsext2/figures/switching_highvar.pdf"), width = 10)
```

Visualise

Stable state finding

Finding

Setup parameter

```
options(mc.cores = 7)

# default_sim_duration <- 1000000
# ssfind_minimum_abundances <- rep(0, 3)
# names(ssfind_minimum_abundances) <- c("CB", "PB", "SB")
# sfind_simulation_duration <- default_sim_duration
# ssfind_simulation_sampling_interval <- ssfind_simulation_duration
# ssfind_event_interval <- ssfind_simulation_duration

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 <- 100 ## 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 = 1000000,
  noise_sigma = parameter$noise_sigma,
  minimum_abundances = minimum_abundances,
  sim_duration = 1000000,
  sim_sample_interval = 1000000,
  log10a_series = parameter$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/experiment 1/data/parameter_1e6_x2x6_factorial.RDS"))
saveRDS(var_expt, here("experiments/experiment 1/data/var_expt_1e6_x2x6_factorial.RDS"))

```

Run stable state finding

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

```

run_ss_var_experiment(
  parameter = readRDS(here("experiments/experiment 1/data/parameter_1e6_x2x6_factorial.RDS")),
  var_expt = readRDS(here("experiments/experiment 1/data/var_expt_1e6_x2x6_factorial.RDS"))) %>%
saveRDS(here("experiments/experiment 1/data/ss_data_1e6_x2x6_factorial.RDS"))

```

Process the stable state data

Bring in various stable state datasets

```

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

## sim length 80'000, 20 x 20 factorial, reference maximum diversity
# readRDS(here("experiments/experiment 1/data/ss_data_80000.RDS")) %>%
#   mutate(sim_length = 80000) %>%
#   multidplyr::partition(cluster) %>%
#   mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
#   collect() %>%
#   unnest(cols = c(stability_measures)) %>%
#   saveRDS(here("experiments/experiment 1/data/stab_data_80000.RDS"))

## sim length 1'000'000, 20 x 20 factorial, reference maximum diversity
# readRDS(here("experiments/experiment 1/data/ss_data_1000000_20factorial.RDS")) %>%

```

```

# mutate(sim_length = 1000000) %>%
# multidplyr::partition(cluster) %>%
# mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
# collect() %>%
# unnest(cols = c(stability_measures)) %>%
# saveRDS(here("experiments/experiment 1/data/stab_data_1000000_20factorial.RDS"))

## sim length 1'000'000, 20 SBPBgrad, 5x maximum diversity
# readRDS(here("experiments/experiment 1/data/ss_data_1e6_noCB_5xSBPB_.RDS")) %>%
# mutate(sim_length = 1000000) %>%
# multidplyr::partition(cluster) %>%
# mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
# collect() %>%
# unnest(cols = c(stability_measures)) %>%
# saveRDS(here("experiments/experiment 1/data/stab_data_1e6_noCB_5xSBPB_.RDS"))

## sim length 300'000, 20 SBPBgrad, reference maximum diversity
# readRDS(here("experiments/experiment 1/data/ss_data_300000_small.RDS")) %>%
# mutate(sim_length = 300000) %>%
# multidplyr::partition(cluster) %>%
# mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
# collect() %>%
# unnest(cols = c(stability_measures)) %>%
# saveRDS(here("experiments/experiment 1/data/stab_data_300000.RDS"))

## sim length 1'000'000, 20 SBPBgrad, 2xCB variation, 6xSBPB variation
readRDS(here("experiments/experiment 1/data/ss_data_1e6_x2x6_factorial.RDS")) %>%
  mutate(sim_length = 1000000) %>%
  multidplyr::partition(cluster) %>%
  mutate(stability_measures = list(get_stability_measures(ss_res))) %>%
  collect() %>%
  unnest(cols = c(stability_measures)) %>%
  saveRDS(here("experiments/experiment 1/data/stab_data_1e6_x2x6_factorial.RDS"))

```

SS, no diversity, all diversity, CB only, and SBPB only

No Experimental data

Look at stability measures

No experimental data.

Calculations

Plot raw

Plot log

Extra SBPB diversity

No experimental data.

raw

log

2x CB, 6xSBPB diversity

```
## find various combinations of diversity
var_expt <- readRDS(here("experiments/experiment 1/data/ss_data_1e6_x2x6_factorial.RDS"))
stab_data <- readRDS(here("experiments/experiment 1/data/stab_data_1e6_x2x6_factorial.RDS"))

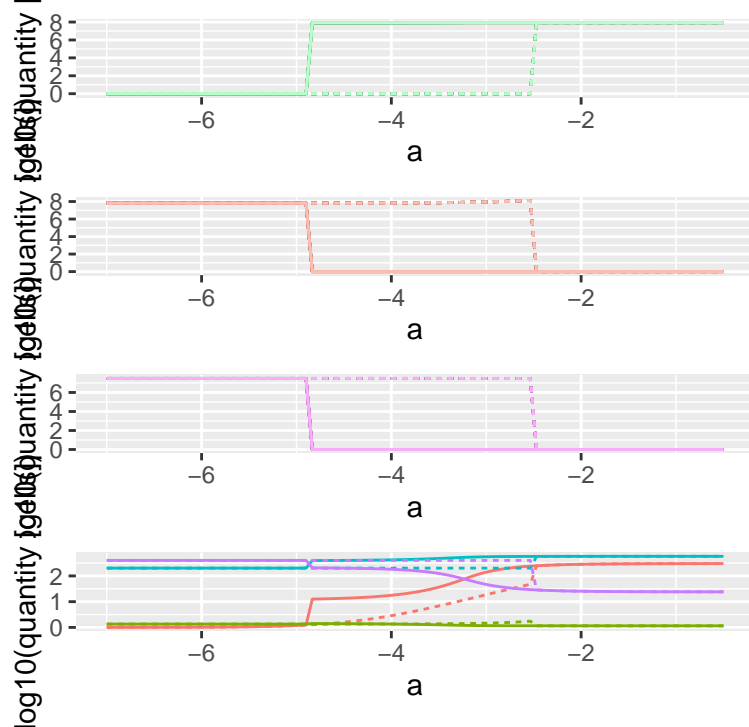
var_expt_levels <- var_expt[,1:6]

no_diversity <- which(rowSums(abs(var_expt_levels))==0)
max_diversity_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,]

p1 <- plot_ss_result1(var_expt,
                      result_index = no_diversity,
                      filename_prefix = NULL,
                      save_image_file = FALSE)
p1
```

CB_gmax_var = 0 CB_h_var = 0
 SB_gmax_var = 0 SB_h_var = 0
 PB_gmax_var = 0 PB_h_var = 0



direction

— down
 ---- up

species

SB_1	SB_4	SB_7
SB_2	SB_5	SB_8
SB_3	SB_6	SB_9

species

PB_1	PB_4	PB_7
PB_2	PB_5	PB_8
PB_3		PB_9

direction

— down

species

O
P
SO
SR

direction

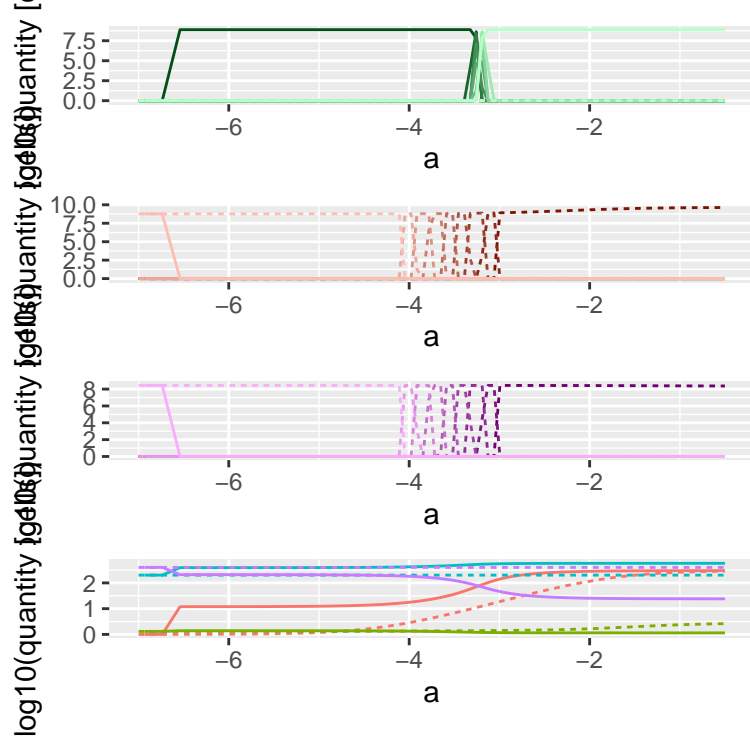
— down

```
#junk1 <- var_expt[no_diversity,]$ss_res[[1]]
```

```
p2 <- plot_ss_result1(var_expt,
  result_index = max_diversty_all,
  filename_prefix = NULL,
  save_image_file = FALSE)
```

p2

CB_gmax_var = 0.031578948 CB_h_var = -0.16
 SB_gmax_var = 0.094736844 SB_h_var = -1.938
 PB_gmax_var = 0.094736844 PB_h_var = -1.938



direction

— down
 up

species

SB_1	SB_4	SB_7
SB_2	SB_5	SB_8
SB_3	SB_6	SB_9

species

PB_1	PB_4	PB_7
PB_2	PB_5	PB_8
PB_3		PB_9

direction

— down

species

O
P
SO
SR

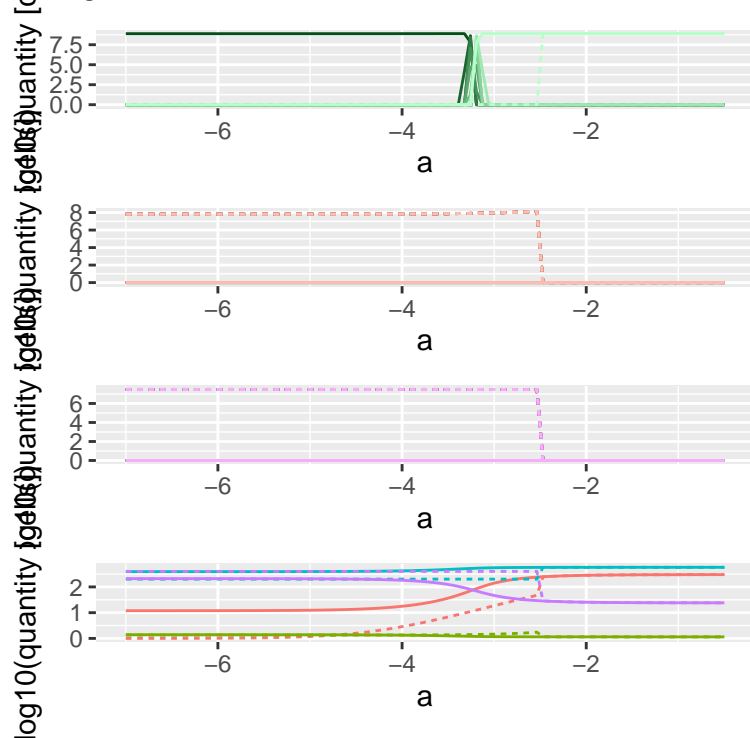
direction

— down

```
p3 <- plot_ss_result1(var_expt,
  result_index = max_only_CB_diversity,
  filename_prefix = NULL,
  save_image_file = FALSE)
```

p3

CB_gmax_var = 0.031578948 CB_h_var = -0.16
 SB_gmax_var = 0 SB_h_var = 0
 PB_gmax_var = 0 PB_h_var = 0



direction

— down
 ---- up

species

SB_1	SB_4	SB_7
SB_2	SB_5	SB_8
SB_3	SB_6	SB_9

species

PB_1	PB_4	PB_7
PB_2	PB_5	PB_8
PB_3		PB_9

direction

— down

O
 P
 SO
 SR

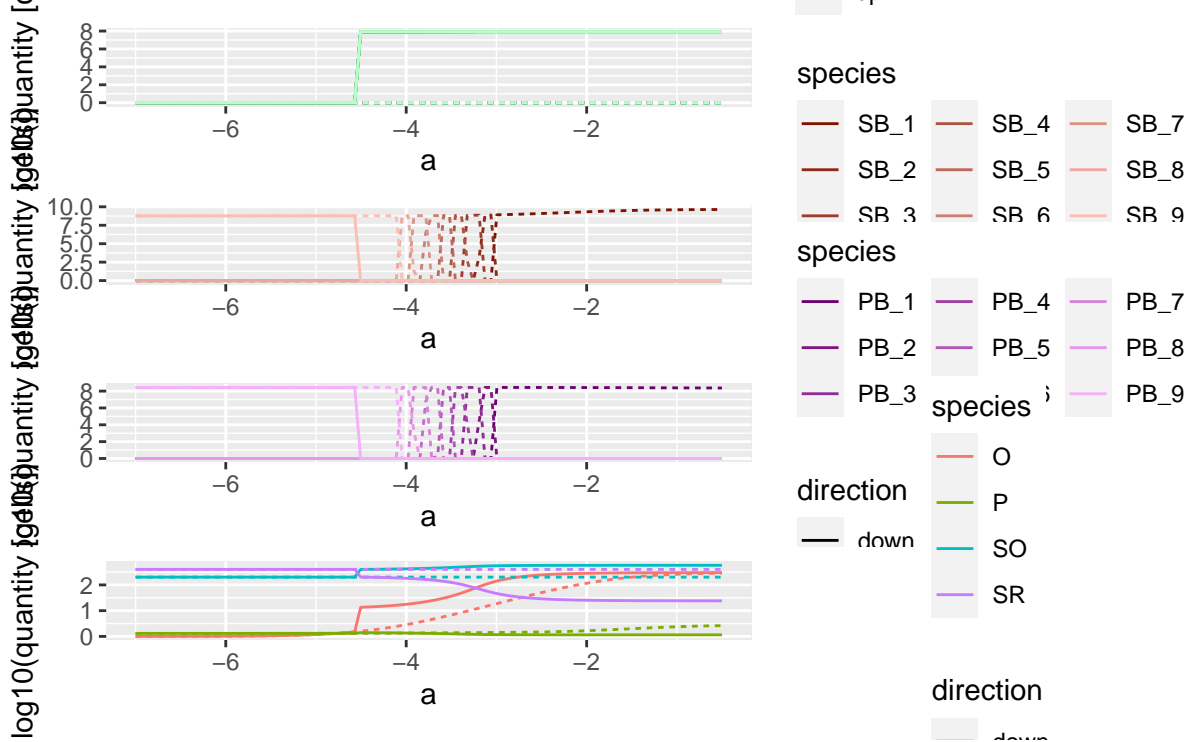
direction

— down

```
p4 <- plot_ss_result1(var_expt,
  result_index = max_only_SBPB_diversity,
  filename_prefix = NULL,
  save_image_file = FALSE)
```

p4

CB_gmax_var = 0 CB_h_var = 0
 SB_gmax_var = 0.094736844 SB_h_var = -1.938
 PB_gmax_var = 0.094736844 PB_h_var = -1.938



```
p_overlay1 <- plot_ss_result2(var_expt[no_diversity,]$ss_res[[1]],
                              var_expt[max_diversity_all,]$ss_res[[1]],
                              xlims = c(-7, -1))
```

```
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument.
```

```
p_overlay1
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

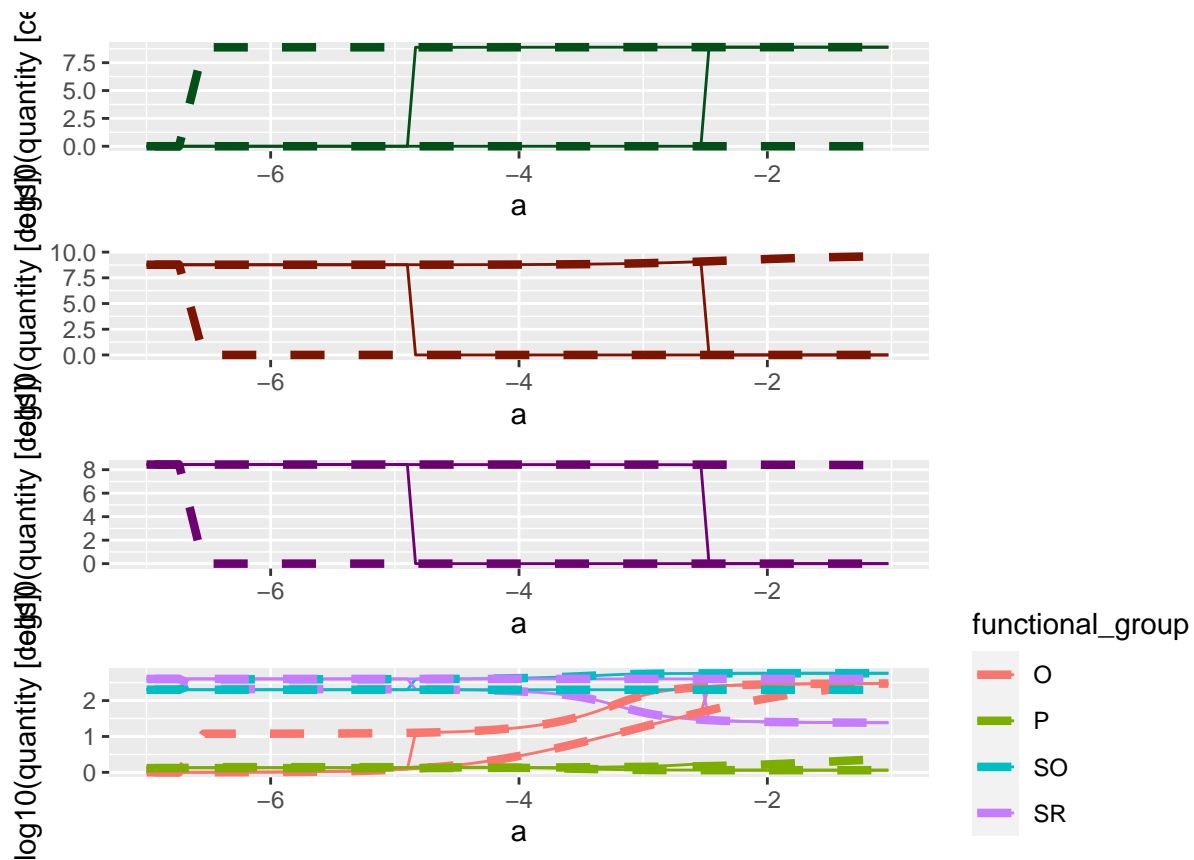
```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```



```
p_overlay2 <- plot_ss_result2(var_expt[no_diversity,]$ss_res[[1]],
                             var_expt[max_only_CB_diversity,]$ss_res[[1]],
                             xlims = c(-7, -1))
```

```
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument
```

```
p_overlay2
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

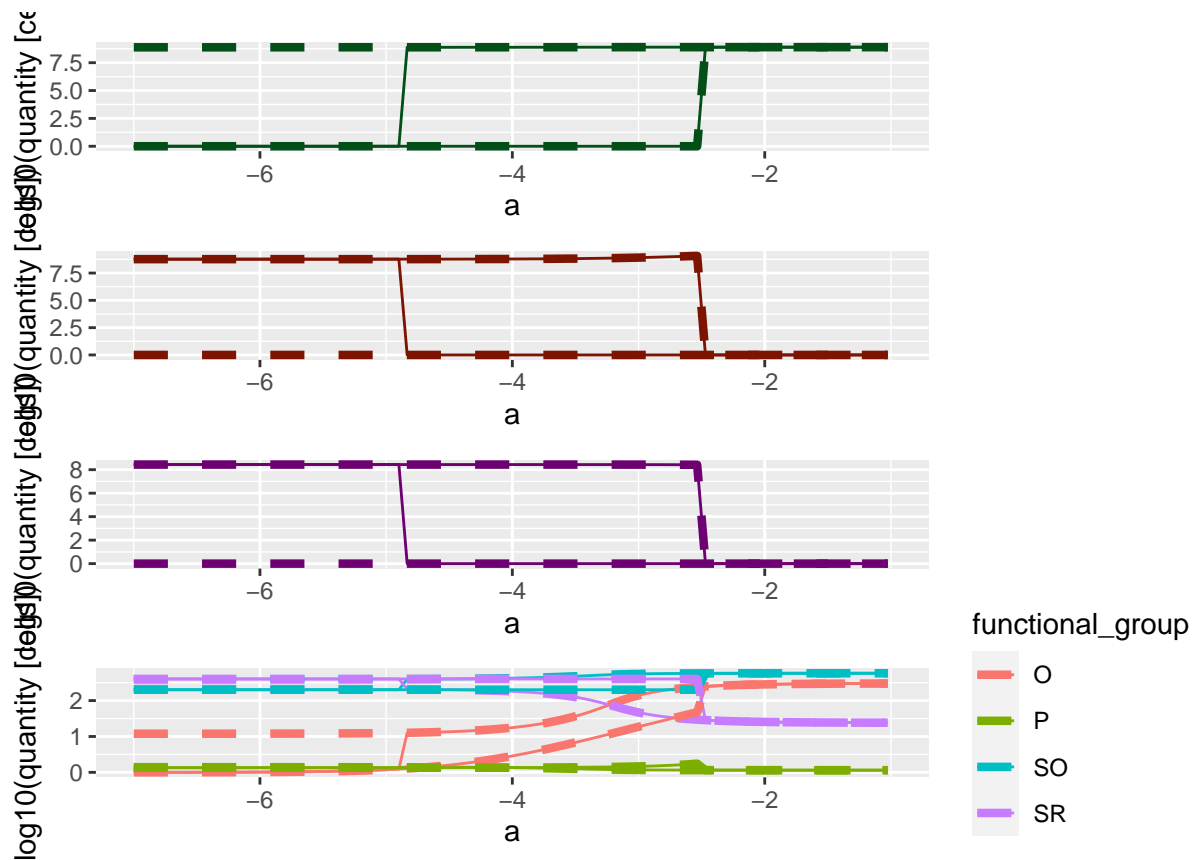
```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```



```
#ss_result1 <- var_expt[no_diversity,]$ss_res[[1]]
#ss_result2 <- var_expt[max_only_CB_diversity,]$ss_res[[1]]
```

```
p_overlay3 <- plot_ss_result2(var_expt[no_diversity,]$ss_res[[1]],
                              var_expt[max_only_SBPB_diversity,]$ss_res[[1]],
                              xlims = c(-7, -1))
```

```
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'a', 'direction', 'var_type'. You can override using the `.groups` argument.
p_overlay3
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

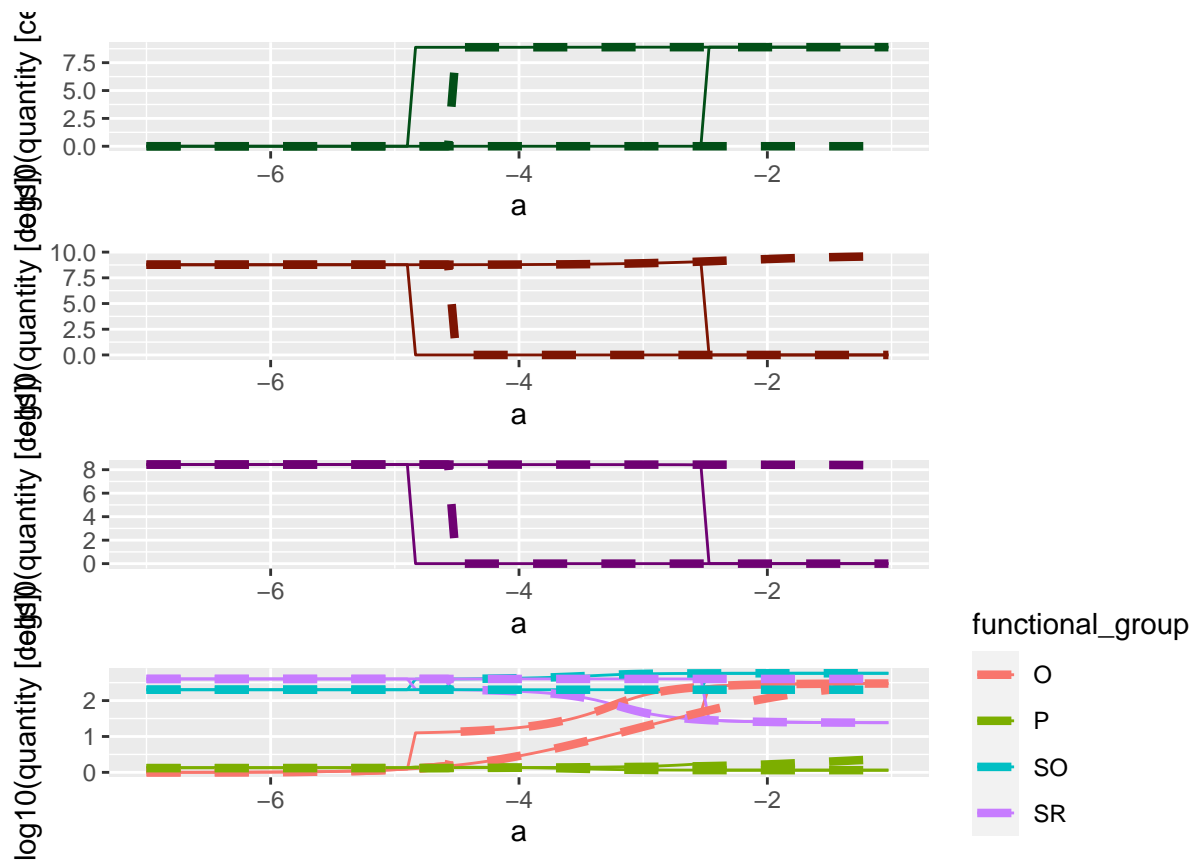
```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```

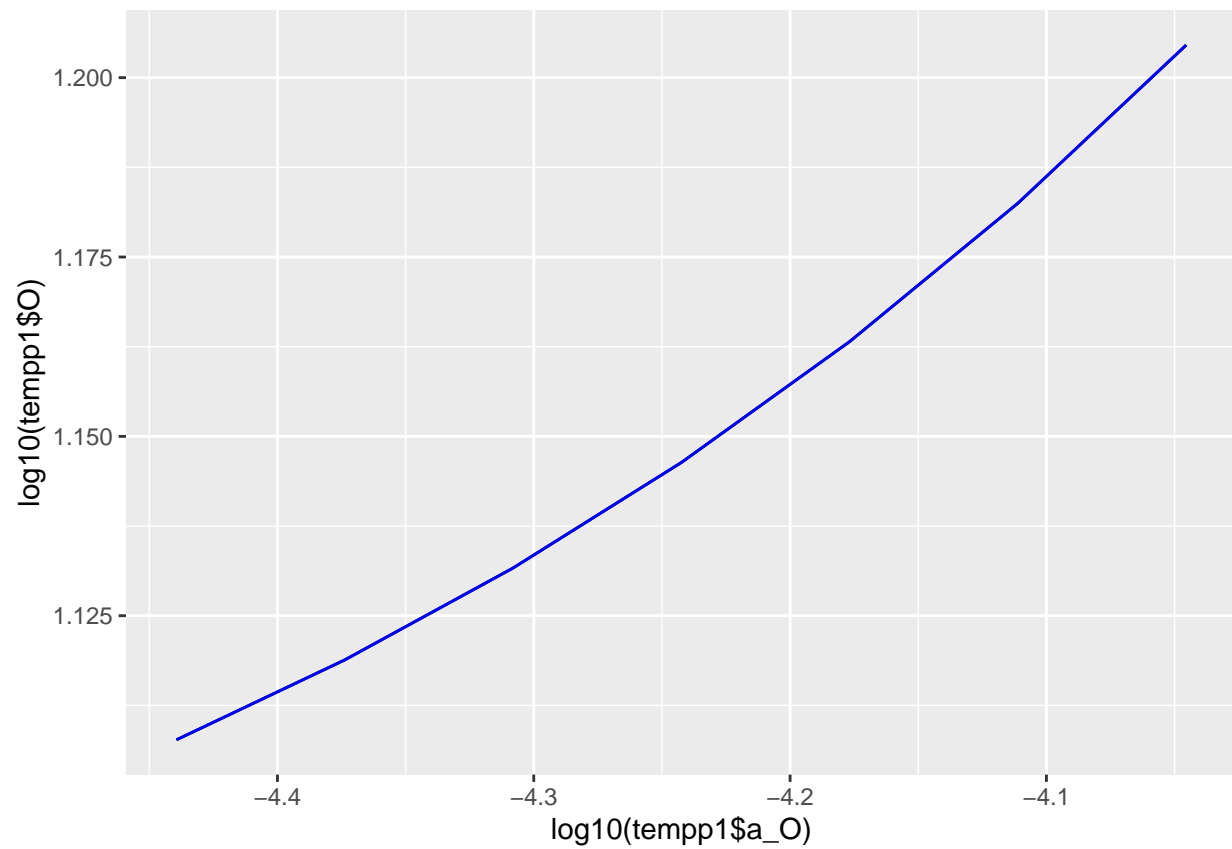
```
## Warning: Removed 64 row(s) containing missing values (geom_path).
```



```
#ss_result3 <- var_expt[max_only_SBPB_diversity,]$ss_res[[1]]
```

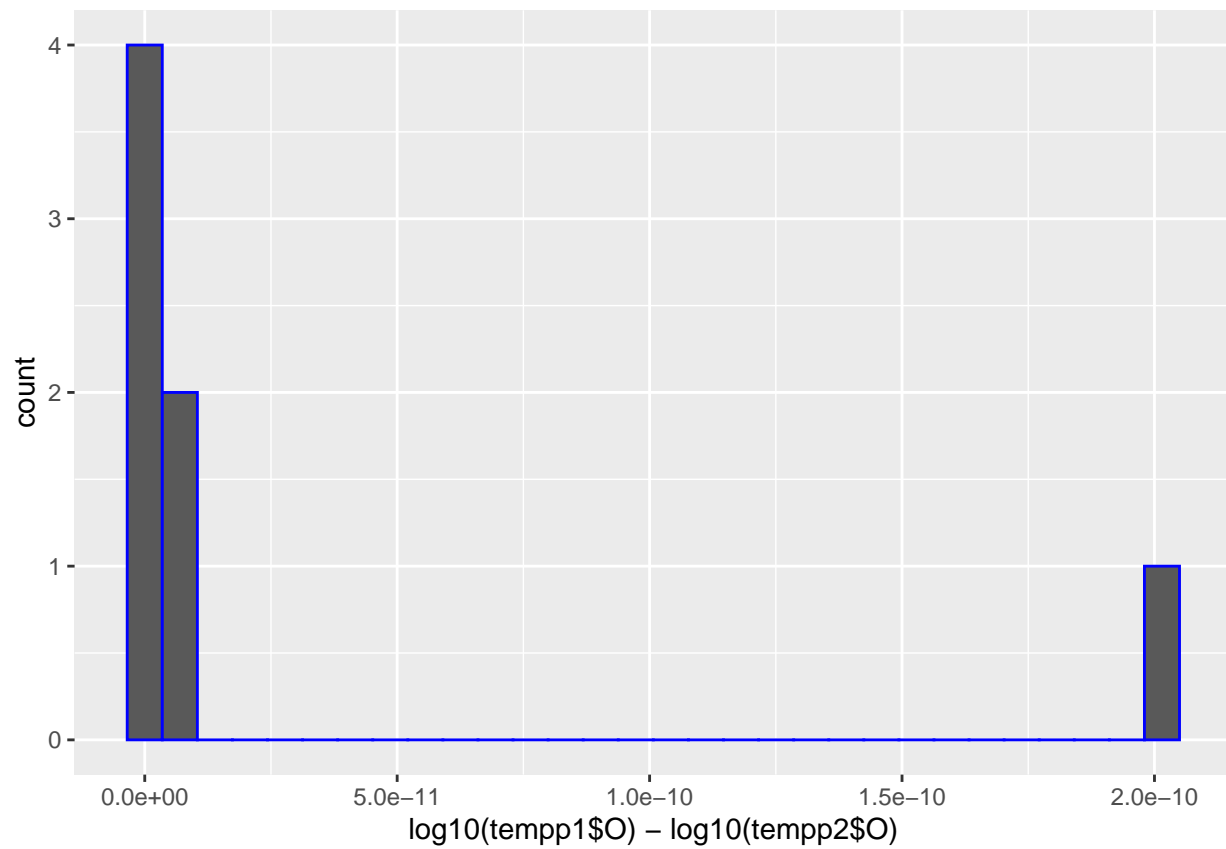
```
tempp1 <- var_expt[no_diversity,]$ss_res[[1]] %>%
  filter(initial_N_CB == 1e10,
         log10(a_0) > -4.5, log10(a_0) < -4)
tempp2 <- var_expt[max_only_SBPB_diversity,]$ss_res[[1]] %>%
  filter(initial_N_CB == 1e10,
         log10(a_0) > -4.5, log10(a_0) < -4)

ggplot() +
  geom_line(mapping = aes(x = log10(tempp1$a_0),
                        y = log10(tempp1$0))) +
  geom_line(mapping = aes(x = log10(tempp2$a_0),
                        y = log10(tempp2$0)),
            col = "blue")
```

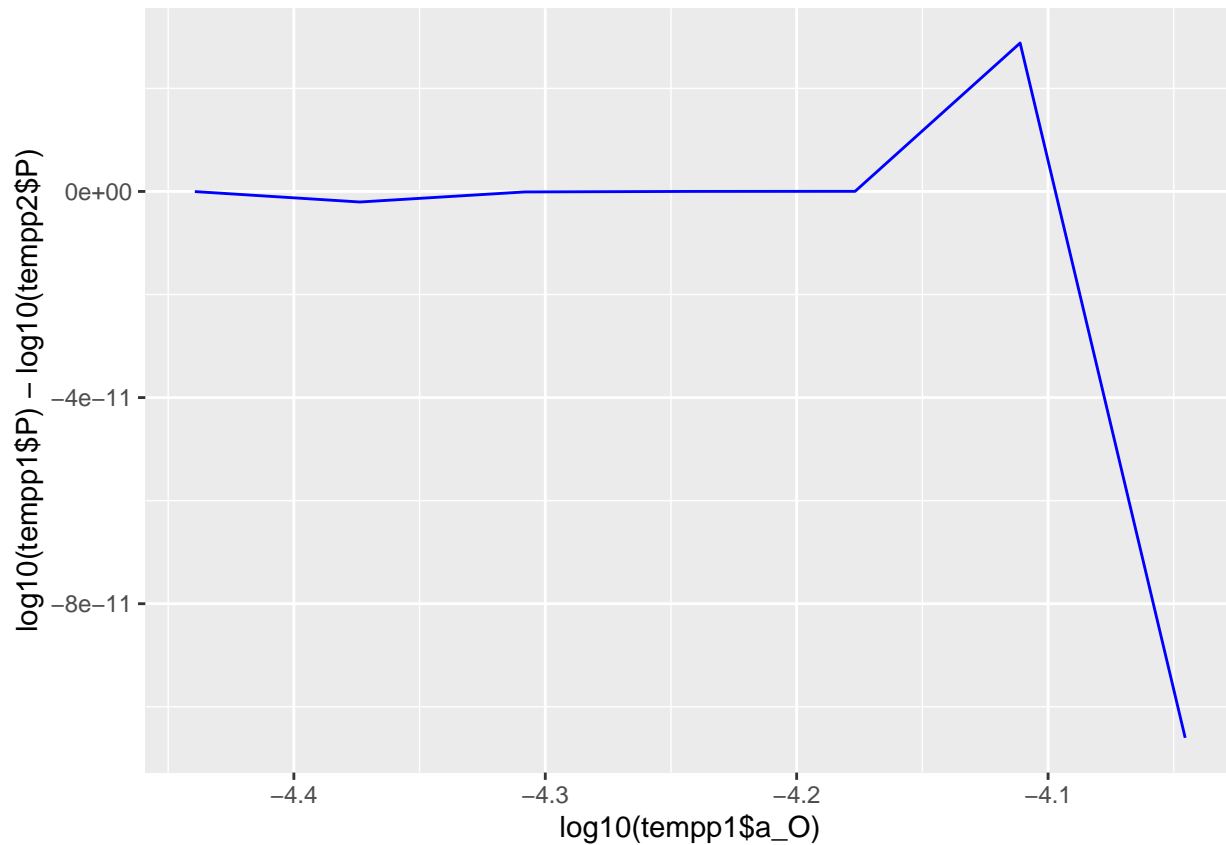


```
ggplot() +  
  geom_histogram(mapping = aes(x = log10(tempp1$O) - log10(tempp2$O)),  
                 col = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot() +  
  geom_line(mapping = aes(x = log10(temp1$a_0),  
    y = log10(temp1$P) - log10(temp2$P)),  
    col = "blue")
```



```

CB_vars <- unique(stab_data$CB_var_gmax_s)
SB_vars <- unique(stab_data$SB_var_gmax_s)

CB_stab_data <- stab_data %>%
  filter(SB_var_gmax_s == 0) %>%
  mutate(var_treat = "CB",
         var_gmax = CB_var_gmax_s)

SBPB_stab_data <- stab_data %>%
  filter(CB_var_gmax_s == 0) %>%
  mutate(var_treat = "SB-PB",
         var_gmax = SB_var_gmax_s)

for_join <- tibble(CB_var_gmax_s = CB_vars,
                  SB_var_gmax_s = SB_vars)
CBSBPB_stab_data <- stab_data %>%
  right_join(for_join) %>%
  mutate(var_treat = "CB-SB-PB",
         var_gmax = CB_var_gmax_s)

## Joining, by = c("CB_var_gmax_s", "SB_var_gmax_s")
all_stab_results <- CB_stab_data %>%
  bind_rows(SBPB_stab_data) %>%
  # bind_rows(results3) %>%
  # bind_rows(results4) %>%
  bind_rows(CBSBPB_stab_data)

```

```

all_stab_results <- all_stab_results %>%
  mutate(var_treat = forcats::fct_relevel(var_treat, levels = c("CB", "SB-PB", "CB-SB-PB")))

## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: CB, CB-SB-PB
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: CB, CB-SB-PB
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: SB-PB, CB-SB-PB
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: CB, SB-PB
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: SB-PB, CB-SB-PB
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
## Warning: Unknown levels in `f`: CB, SB-PB
#saveRDS(all_stab_results, here("experiments/experiment summary/all_stab.RDS"))

#all_stab_results <- readRDS(here("experiments/experiment summary/all_stab.RDS"))

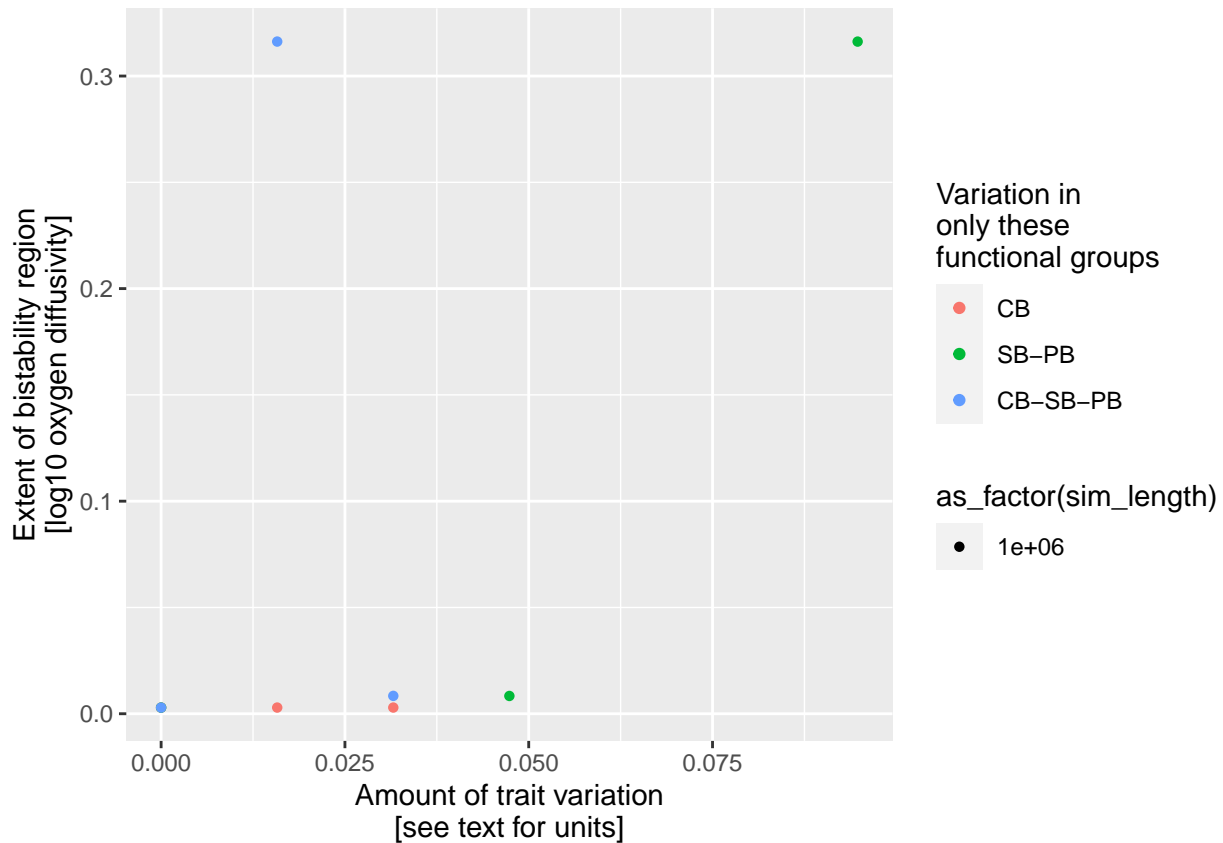
```

raw

```

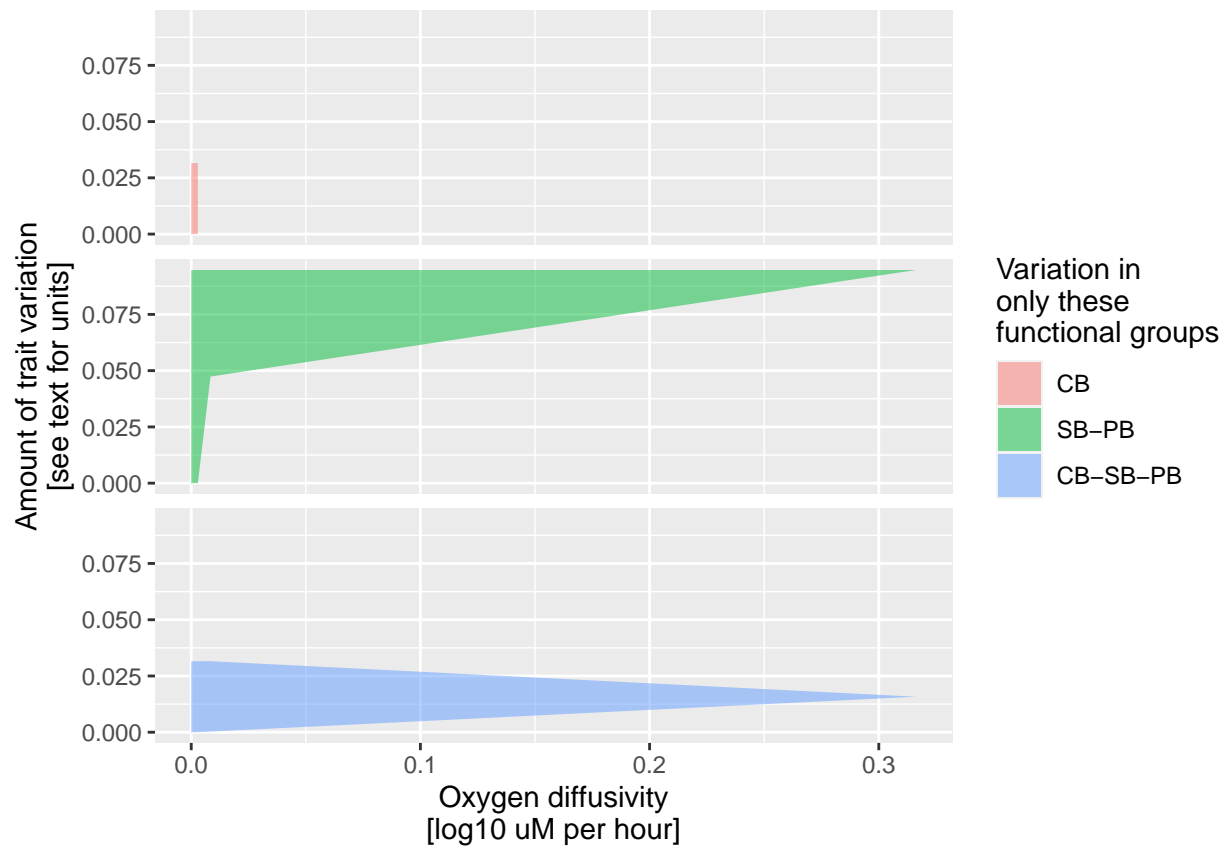
all_stab_results %>%
  filter(Species == "0") %>%
  ggplot(aes(x = var_gmax, y = hyst_range_raw, col=var_treat, shape = as_factor(sim_length))) +
  geom_point() +
  xlab("Amount of trait variation\n[see text for units]") +
  ylab("Extent of bistability region\n[log10 oxygen diffusivity]") +
  labs(col = "Variation in\nonly these\nfunctional groups")

```

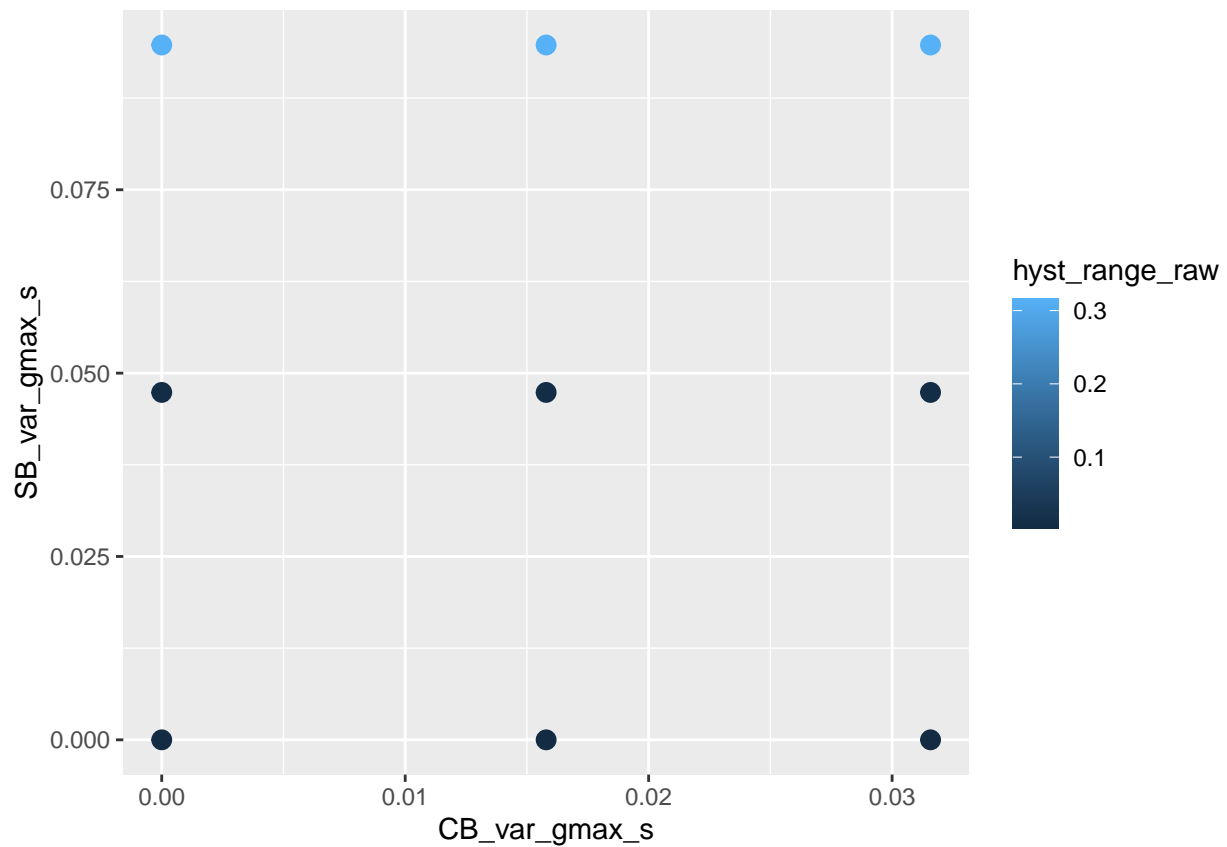



```
##ggsave("manuscript/figures/extent_of_bistab1.pdf", height = 4)

all_stab_results %>%
  #filter(var_treat == "CB") %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = var_gmax,
             ymin = hyst_min_raw,
             ymax = hyst_max_raw,
             fill=var_treat)) +
  geom_ribbon(alpha = 0.5) +
  facet_wrap( ~ var_treat, nrow = 3) +
  xlab("Amount of trait variation\n[see text for units]") +
  ylab("Oxygen diffusivity\n[log10 uM per hour]") +
  labs(fill = "Variation in\nonly these\nfunctional groups") +
  coord_flip() +
  theme(
    strip.background = element_blank(),
    strip.text.x = element_blank()
  )
)
```

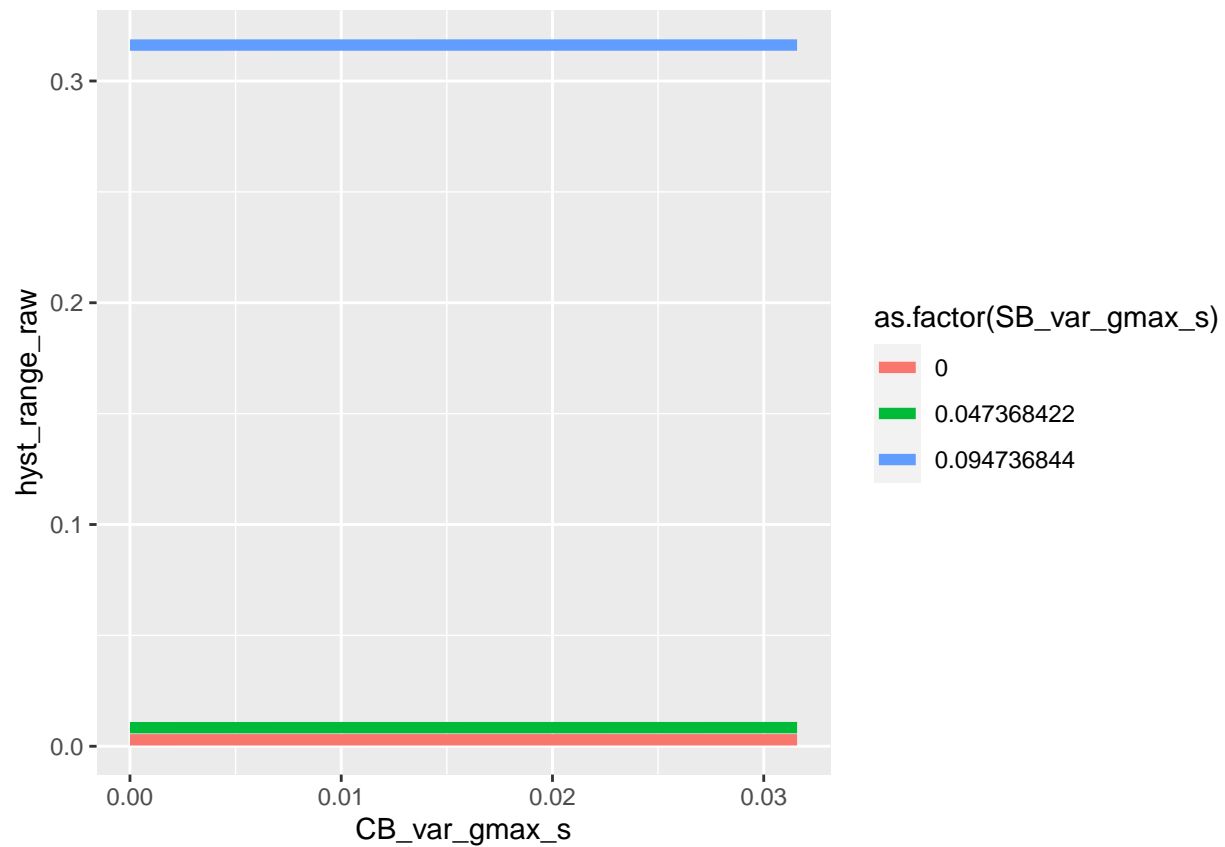


```
##ggsave("manuscript/figures/extent_of_bistab2.pdf", height = 4)
stab_data %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = CB_var_gmax_s, y = SB_var_gmax_s, fill = hyst_range_raw, col = hyst_range_raw)) +
  geom_point(size = 3)
```



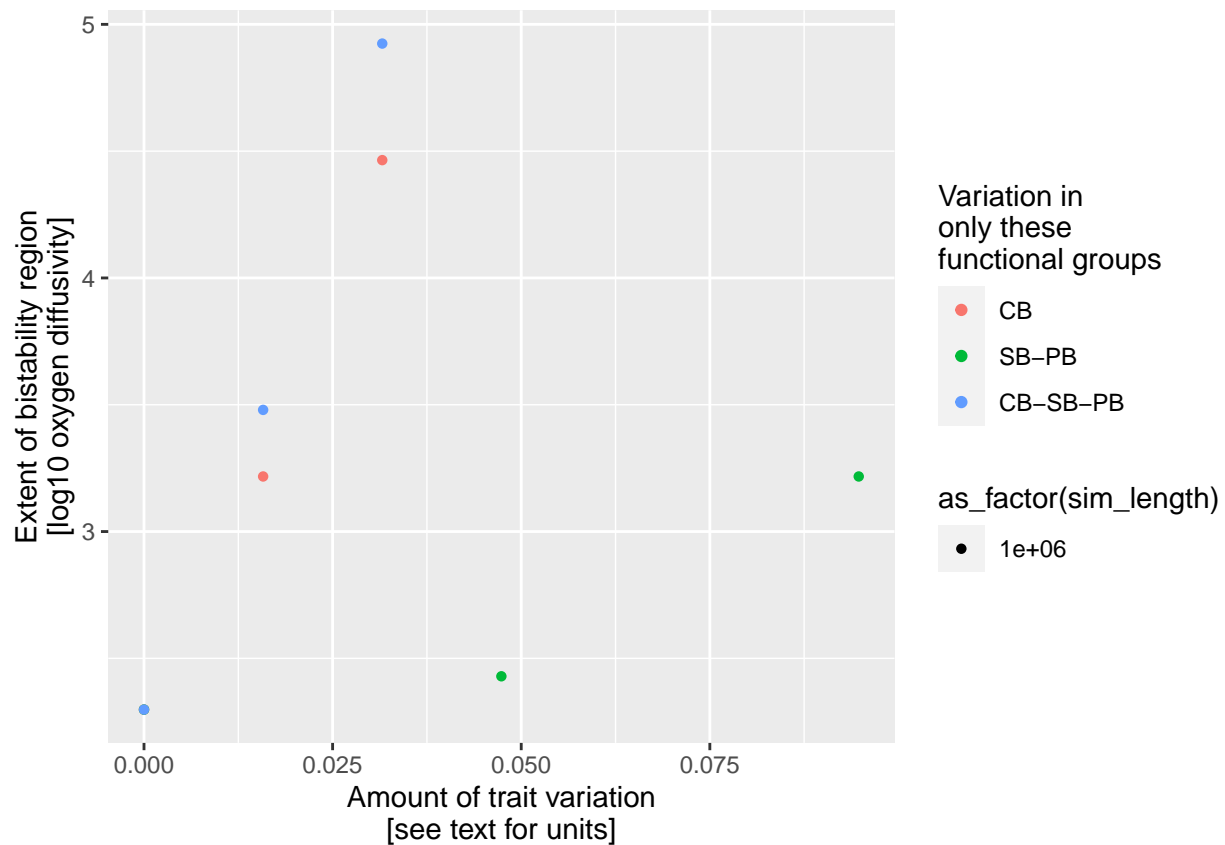
```
##ggsave("manuscript/figures/extent_of_bistab2.pdf", height = 4)

stab_data %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = CB_var_gmax_s, y = hyst_range_raw, col = as.factor(SB_var_gmax_s))) +
  geom_line(size = 2)
```



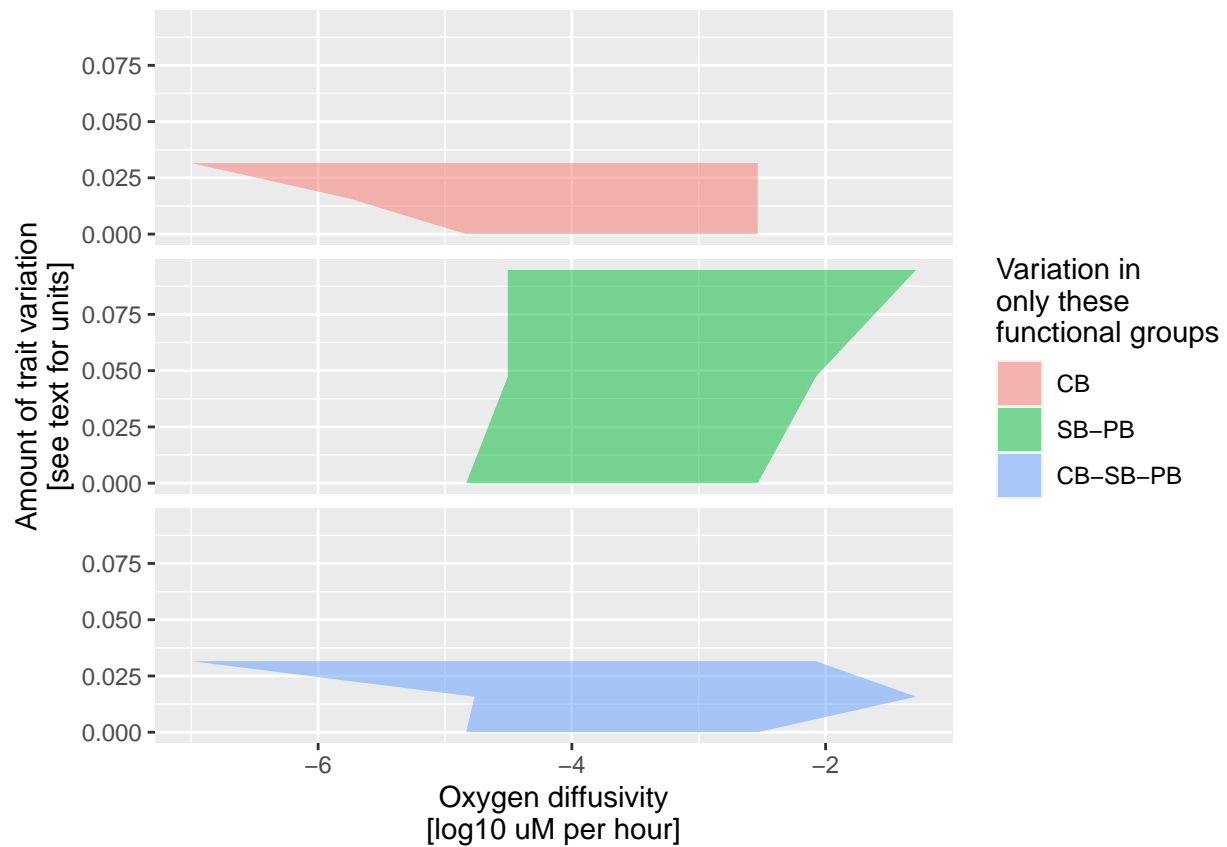
log

```
all_stab_results %>%
  filter(Species == "0") %>%
  ggplot(aes(x = var_gmax, y = hyst_range_log, col=var_treat, shape = as_factor(sim_length))) +
  geom_point() +
  xlab("Amount of trait variation\n[see text for units]") +
  ylab("Extent of bistability region\n[log10 oxygen diffusivity]") +
  labs(col = "Variation in\nonly these\nfunctional groups")
```

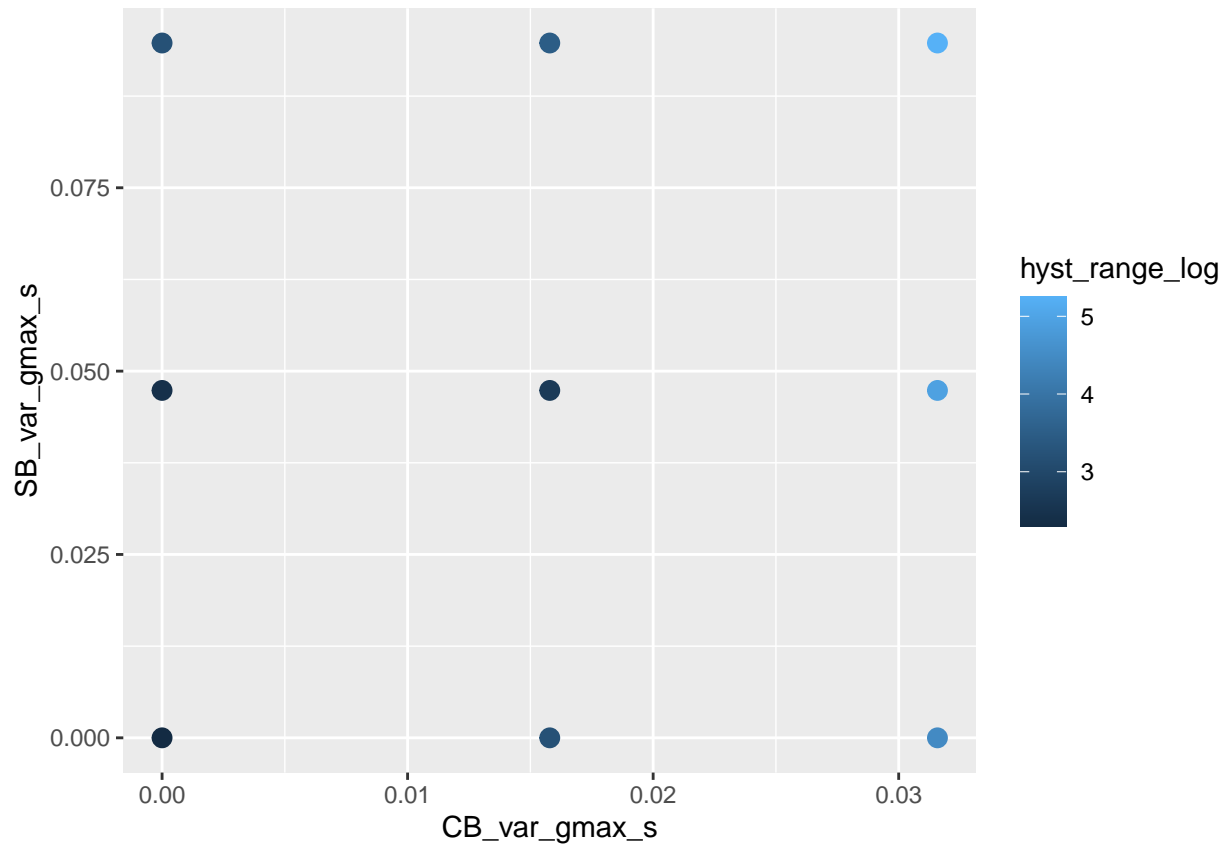


```
##ggsave("manuscript/figures/extent_of_bistab1.pdf", height = 4)

all_stab_results %>%
  #filter(var_treat == "CB") %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = var_gmax,
             ymin = hyst_min_log,
             ymax = hyst_max_log,
             fill=var_treat)) +
  geom_ribbon(alpha = 0.5) +
  facet_wrap( ~ var_treat, nrow = 3) +
  xlab("Amount of trait variation\n[see text for units]") +
  ylab("Oxygen diffusivity\n[log10 uM per hour]") +
  labs(fill = "Variation in\nonly these\nfunctional groups") +
  coord_flip() +
  theme(
    strip.background = element_blank(),
    strip.text.x = element_blank()
  )
)
```

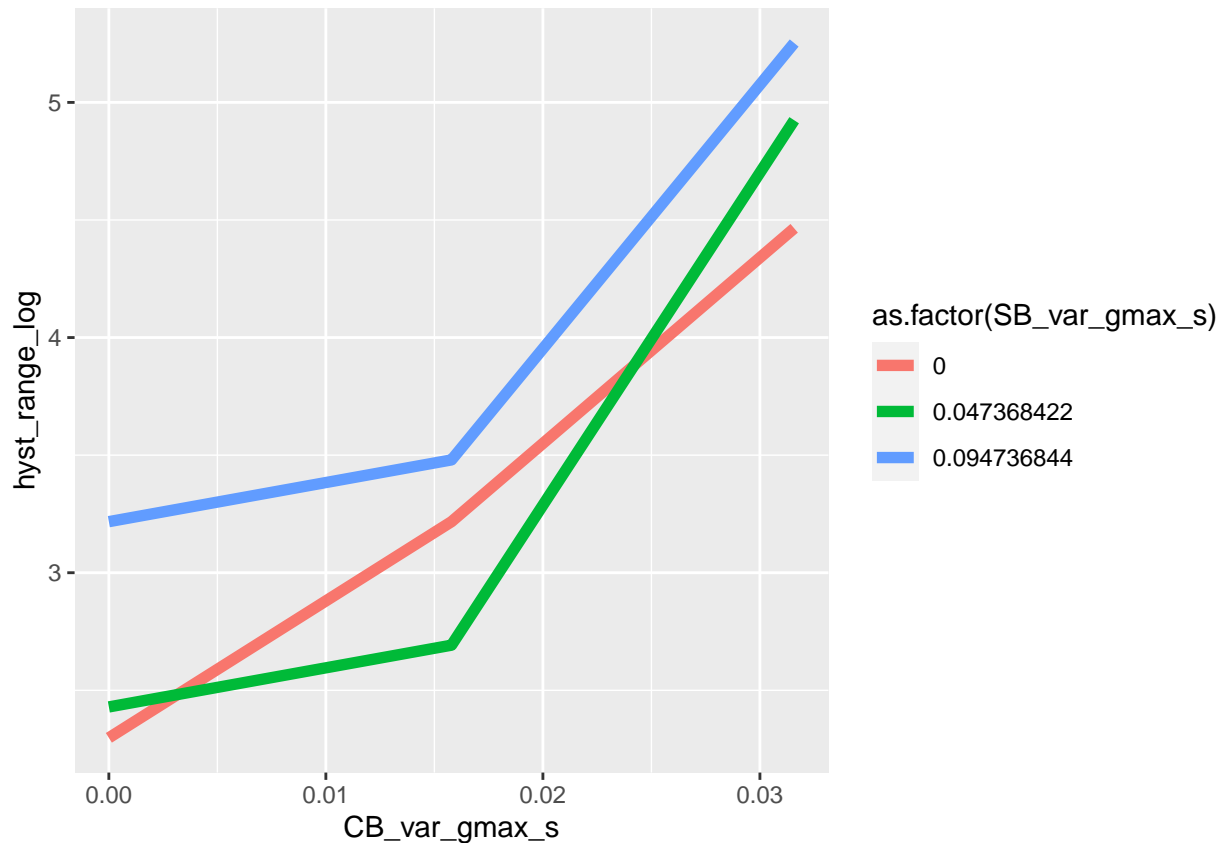


```
##ggsave("manuscript/figures/extent_of_bistab2.pdf", height = 4)
stab_data %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = CB_var_gmax_s, y = SB_var_gmax_s, fill = hyst_range_log, col = hyst_range_log)) +
  geom_point(size = 3)
```



```
##ggsave("manuscript/figures/extent_of_bistab2.pdf", height = 4)

stab_data %>%
  filter(Species == "0",
         sim_length == 1e6) %>%
  ggplot(aes(x = CB_var_gmax_s, y = hyst_range_log, col = as.factor(SB_var_gmax_s))) +
  geom_line(size = 2)
```



What effect of changing the length of the simulations

No Experimental data.

FROM HERE NOT CHANGED TO NEW FORMAT!!!!!!

Some puzzles

Puzzle 1

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 ($\log_{10}(a_{\text{O}}) = -4.8$) 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/experiment 1/data/ss_data_1e6_x2x6_factorial.RDS"))
var_expt_levels <- var_expt[,1:6]
no_diversity <- which(rowSums(abs(var_expt_levels))==0)
max_diversity_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 <- readRDS(here("experiments/experiment 1/data/sim_res_novar3.RDS"))
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/experiment 1/data/puzzle1_1.RDS"))

```

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 occurring. 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