

# POMP group-level results - All boxes

Madeline Peters, Aaron King

2023-06-12

## Load required packages

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.1    v purrr  1.0.1
## v tibble  3.1.8    v dplyr  1.1.0
## v tidyr   1.3.0    v stringr 1.5.0
## v readr   2.1.4    v forcats 1.0.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringi)
library(doParallel)
```

```
## Loading required package: foreach
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##   accumulate, when
##
## Loading required package: iterators
## Loading required package: parallel
```

```
library(pomp)
```

```
##
## Welcome to pomp!
##
## As of version 4.6, no user-visible pomp function has a name that
## includes a dot ('.'). Function names have been changed to replace the
## dot with an underscore ('_'). For more information, see the pomp blog:
## https://kingaa.github.io/pomp/blog.html.
##
##
```

```
## Attaching package: 'pomp'
##
## The following object is masked from 'package:purrr':
##
##      map
```

```
library(panelPomp)
```

## Set working directory

```
setwd("~/Documents/GitHub/bdd/nw11_hier/final_files/")
```

## Colour-blind friendly colour palette

```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

## Read in PNAS data

```
read_csv("data.csv",
          col_types="iiinnnn"
) %>%
  mutate(
    mouseid=sprintf("%02d-%02d",box,mouse),
    box=sprintf("%02d",box),
    mouse=sprintf("%02d",mouse),
    paba=as.factor(box),
    rbc_density=rbc_density/1000
  ) %>%
  mutate(
    paba=recode(
      paba,
      "01"="0.05", "02"="0.005", "03"="0.0005", "04"="0", "05"="control"
    )
  ) %>%
  select(
    day,
    Pd=ama_density,
    RBC=rbc_density,
    Ter119=ter119_density,
    CD71=cd71_density,
    mouseid,
    paba,box,mouse
  ) %>%
  arrange(mouseid,day) %>%
  mutate(
    paba=as.character(paba),
    Ter119=ifelse(Ter119==0,NA,Ter119),
```

```

    CD71=ifelse(CD71==0,NA,CD71)
  ) %>%
  mutate(
    Eryth=(1-CD71/Ter119)*RBC,
    Retic=CD71/Ter119*RBC
  ) -> flow

```

## Obtain estimates for beta and dose

```

flow %>%
  filter(day<=4) %>%
  lm(log(Pd)~box:day+mouseid-1,data=.) -> fit2

coef(fit2) %>%
  bind_rows() %>%
  gather(var,val) %>%
  mutate(
    var=stri_replace_all_regex(var,"mouseid(\\d{2})-(\\d{2})","dose[$1-$2]"),
    var=stri_replace_all_regex(var,"box(\\d{2}):day","Beta[$1]"),
    val=exp(val)
  ) -> theta1

expand.grid(
  box=sprintf("%02d",1:5),
  mouse=sprintf("%02d",1:3)
) %>%
  mutate(
    mouseid=paste0(box,"-",mouse),
    betavar=sprintf("Beta[%s]",box),
    dosevar=sprintf("dose[%s]",mouseid)
  ) %>%
  left_join(theta1,by=c("betavar"="var")) %>%
  rename(Beta=val) %>%
  left_join(theta1,by=c("dosevar"="var")) %>%
  rename(dose=val) %>%
  select(-betavar,-dosevar) %>%
  arrange(box,mouse) %>%
  mutate(
    Beta=coalesce(Beta,0),
    dose=coalesce(dose,0)
  ) -> theta

registerDoParallel()

theta %>%
  mutate(
    sigmaPd = 2,
    sigmaRBC = 0.1,
    sigmaRetic = 0.3,
    sigmaW = 1,
    sigmaN = 0.5,

```

```

sigmaR = 0.5,
E_0 = 8.0e6,
R_0 = 3e5,
W_0 = 8.8e4,
N_0 = 8e5
) -> theta

```

Create object pos with pomp object for each mouse

```

foreach (m = iter(theta,"row"),.inorder=TRUE,.combine=c) %dopar% {

  flow %>%
    filter(mouseid==m$mouseid) %>%
    select(day,Pd,RBC,Retic) %>%
    mutate(Retic=if_else(day %in% c(0,14),NA_real_,Retic)) %>%
    pomp(
      params=select(m,-mouseid,-box,-mouse) %>% unlist(),
      times="day",
      t0=0,
      rmeasure=Csnippet("
Retic = rlnorm(log(1+R),sigmaRetic)-1;
RBC = rlnorm(log(1+E+R),sigmaRBC)-1;
Pd = rlnorm(log(1+K),sigmaPd)-1;"),
      dmeasure=Csnippet("
double l1, l2, l3;
l1 = (R_FINITE(Retic)) ? dlnorm(1+Retic,log(1+R),sigmaRetic,1) : 0;
l2 = (R_FINITE(RBC)) ? dlnorm(1+RBC,log(1+E+R),sigmaRBC,1) : 0;
l3 = (R_FINITE(Pd) && Pd>0) ? dlnorm(1+Pd,log(1+K),sigmaPd,1) : 0;
lik = (give_log) ? l1+l2+l3 : exp(l1+l2+l3);"),
      rprocess=discrete_time(
        step.fun=Csnippet("
double Mold = M;
M = Beta*K*exp(-(W+N)/(R+E));
E = (R+E)*exp(-(Mold+N)/(R+E));
N = rlnorm(log(N),sigmaN);
W = rlnorm(log(W),sigmaW);
R = rlnorm(log(R),sigmaR);
K = (R+E>0) ? (R+E)*(1-exp(-M/(R+E))): 0;
"),
        delta.t=1
      ),
      partrans=parameter_trans(
        log=c("sigmaW","sigmaR","sigmaN",
              "sigmaPd","sigmaRBC","sigmaRetic",
              "N_0","W_0","E_0","R_0")
      ),
      rinit=Csnippet("
E = E_0;
R = R_0;
N = N_0;
W = W_0;
M = 0;

```

```

K = dose;"),
  statenames=c("E","R","W","N","M","K"),
  paramnames=c(
    "Beta","dose",
    "sigmaPd","sigmaRBC","sigmaRetic",
    "sigmaW","sigmaN","sigmaR",
    "E_0","R_0","W_0","N_0"
  )
)
} %>%
  set_names(theta$mouseid) -> pos

```

## Obtain MLE for beta and dose

```

pf4name <- "m5pf4.rds"
pf4 <- readRDS(pf4name)

pf4 %>%
  select(loglik,starts_with("sigma"),ends_with("_0")) %>%
  filter(loglik==max(loglik)) -> mle

mle %>% unlist() -> p
pos %>% panelPomp(shared=p) %>% coef() %>%
  rbind() %>% as_tibble() -> theta
theta %>%
  select(starts_with("Beta"),starts_with("dose")) %>%
  gather(variable, value) %>%
  tidyr::extract(variable, into=c("variable","mouseid"),
    regex="([[:alnum:]]+)((.+(.+)\\))" %>%
  spread(variable,value) -> bdf

```

## Load in PNAS trajectories

```

sm1name <- "m5sm1.rds"
sm1 <- readRDS(sm1name)

```

Create dataframe with weighted trajectories (grouped by mouseid first, then by box)

```

sm1 |>
  as_tibble() |>
  filter(mouseid!="01-02",mouseid!="02-03") |> #remove underdosed mice
  pivot_wider(names_from=variable,values_from=value) |>
  left_join(bdf,by=c("mouseid")) |>
  separate_wider_delim(cols="mouseid",delim="-",names=c("box","mouse"),cols_remove=FALSE) |>
  group_by(mouseid) |>
  mutate(

```

```

SM=exp(-M/(R+E)),
SN=exp(-N/(R+E)),
SW=exp(-W/(R+E)),
Qps=(1-SM)*SW*SN,
Qpn=N/(N+W)*(1-SM)*(1-SW*SN),
Qpw=W/(N+W)*(1-SM)*(1-SW*SN),
Qun=SM*(1-SN),
Qus=SM*SN,
lambda_r=Beta*R/M*Qps,
lambda_e=Beta*E/M*Qps,
lambda_n=Beta*(R+E)/M*Qpn,
lambda_w=Beta*(R+E)/M*Qpw,
lambda_u=Beta-lambda_r-lambda_e-lambda_n-lambda_w,
loss=(R+E)*(1-Qus),
rbc=E+R,
varN=N/rbc,
lik=exp(loglik-max(loglik))
) |>
ungroup() |>
select(-loglik,-Beta,-dose) |>
gather(variable,value,-rep,-time,-mouse,-box,-lik,-mouseid) |>
filter(is.finite(value)) %>%
group_by(box,time,variable) |>
dplyr::reframe(
  value=pomp::wquant(x=value,probs=c(0.05,0.5,0.95),weights=lik),
  name=c("lo","med","hi")
) |>
ungroup() |>
pivot_wider() -> group_traj

group_traj$pABA <- factor(group_traj$box,levels=c("05","04","03","02","01"),
  labels=c("Control","0%","0.05%","0.5%","5%"))

```

Remove estimates for W for control mice

```
group_traj <- group_traj |> dplyr::slice(-which(group_traj$box=="05"&group_traj$variable=="W"))
```

Plot group-level trajectories for all variables

```

group_traj |>
filter(time<=20,variable%in%c("rbc","E","R","N","W","K")) |>
mutate(variable=case_match(variable,
  "K"~"Parasites",
  "rbc"~"RBC",
  "R"~"Reticulocytes",
  "E"~"Erythrocytes",
  "N"~"Indiscriminate killing",
  "W"~"Targeted killing")) |>
ggplot()+

```

```

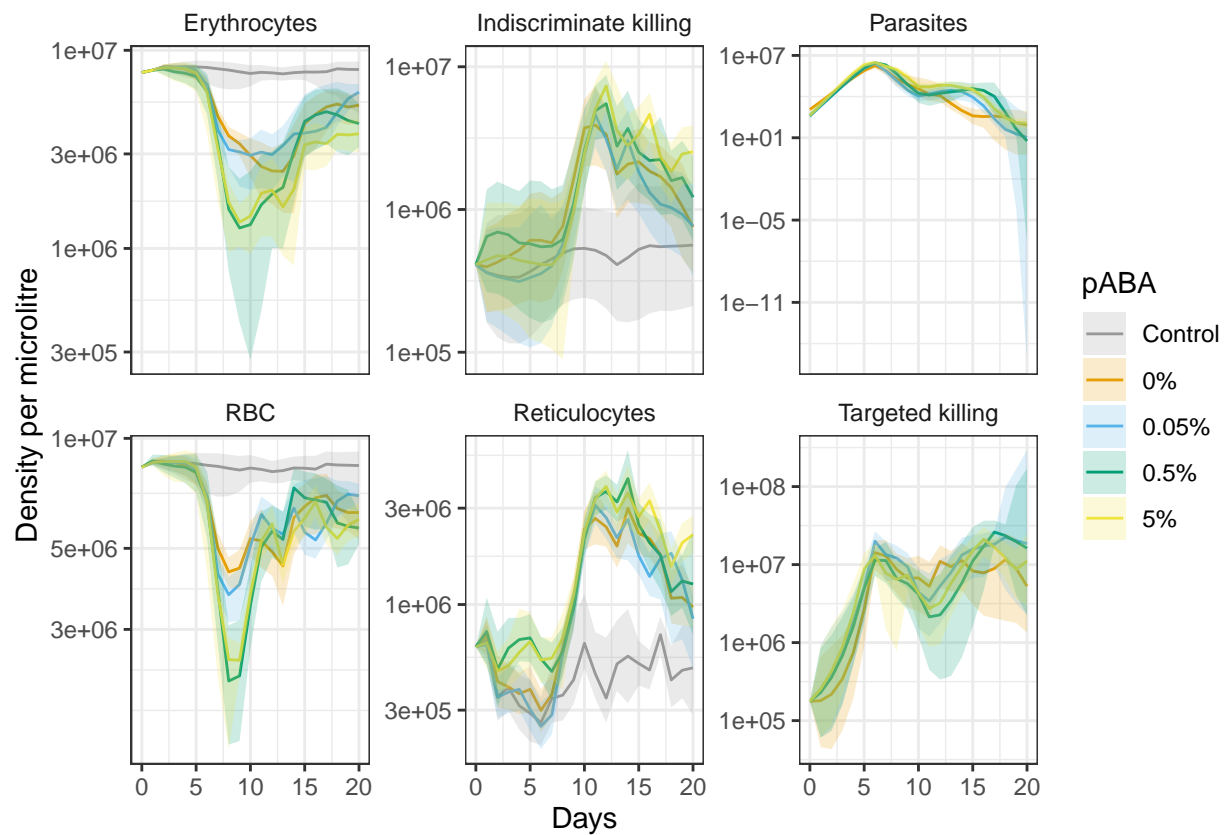
geom_line(aes(x=time,y=med,col=pABA))+
geom_ribbon(aes(x=time,ymin=lo,ymax=hi,fill=pABA),alpha=0.2)+
scale_y_log10()+
scale_colour_manual(values=cbPalette)+
scale_fill_manual(values=cbPalette)+
xlab("Days")+ylab("Density per microlitre")+
facet_wrap(variable~.,scales="free_y")+
theme_bw()+
theme(
  strip.background = element_blank()
)

```

```

## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis

```



Plot group-level trajectories for reticulocytes with day of peak parasitaemia

```

group_traj |> filter(pABA%in%c("0%"),variable=="K") -> tmp
peak.day <- tmp$time[which(tmp$med==max(tmp$med))]

group_traj |>

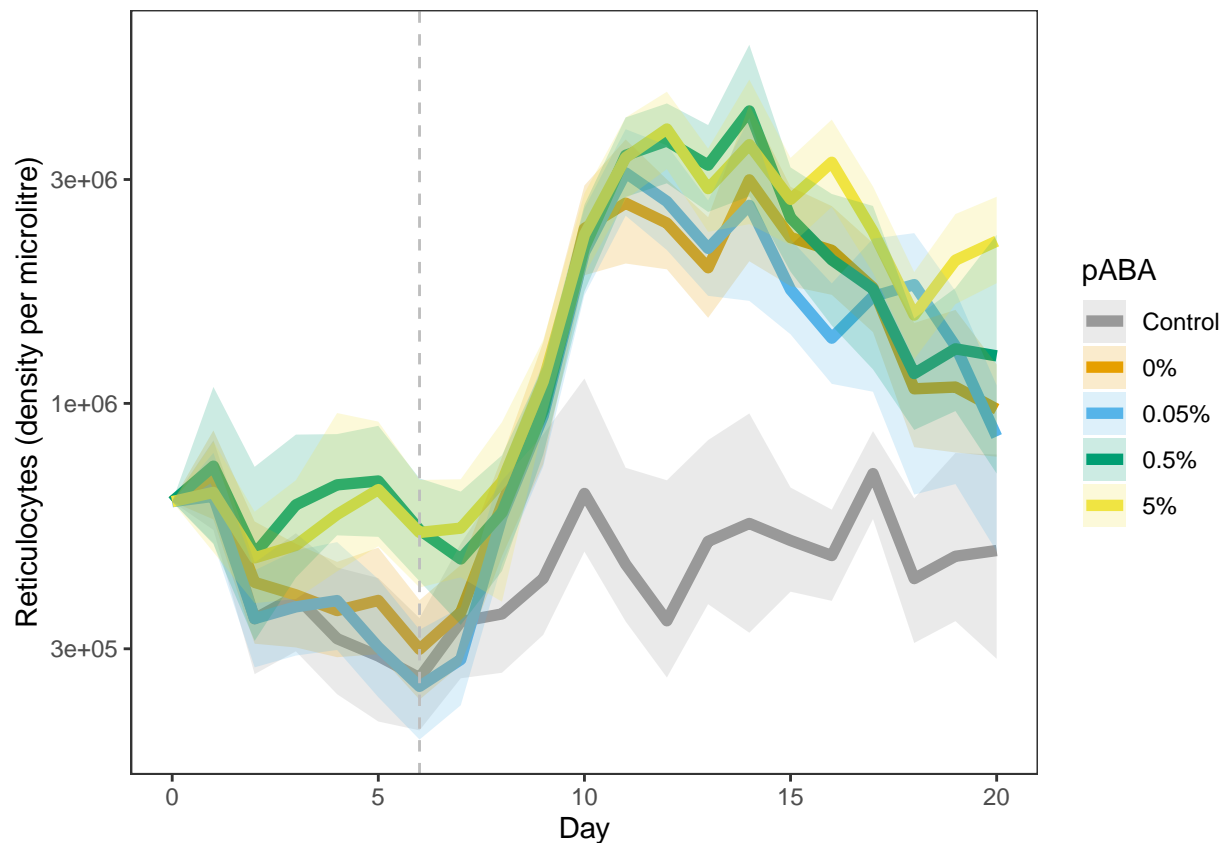
```

```

filter(time<=20,variable%in%c("R")) |>
mutate(variable=case_match(variable,
                           "R"~"Reticulocytes")) |>

ggplot()+
geom_line(aes(x=time,y=med,color=pABA),linewidth=2)+
geom_ribbon(aes(x=time,ymin=lo,ymax=hi,fill=pABA),alpha=0.2)+
geom_vline(xintercept=peak.day,linetype="dashed",col="grey")+
scale_y_log10()+
scale_colour_manual(values=cbPalette)+
scale_fill_manual(values=cbPalette)+
xlab("Day")+ylab("Reticulocytes (density per microlitre)")+
theme_bw()+
theme(strip.background=element_blank(),
       panel.grid = element_blank())
)

```



## Plot group-level swirlies

```

group_traj |>
  filter(time<=20) |>
  select(-hi,-lo) |>
  pivot_wider(names_from=variable,values_from=med) -> swirls

```



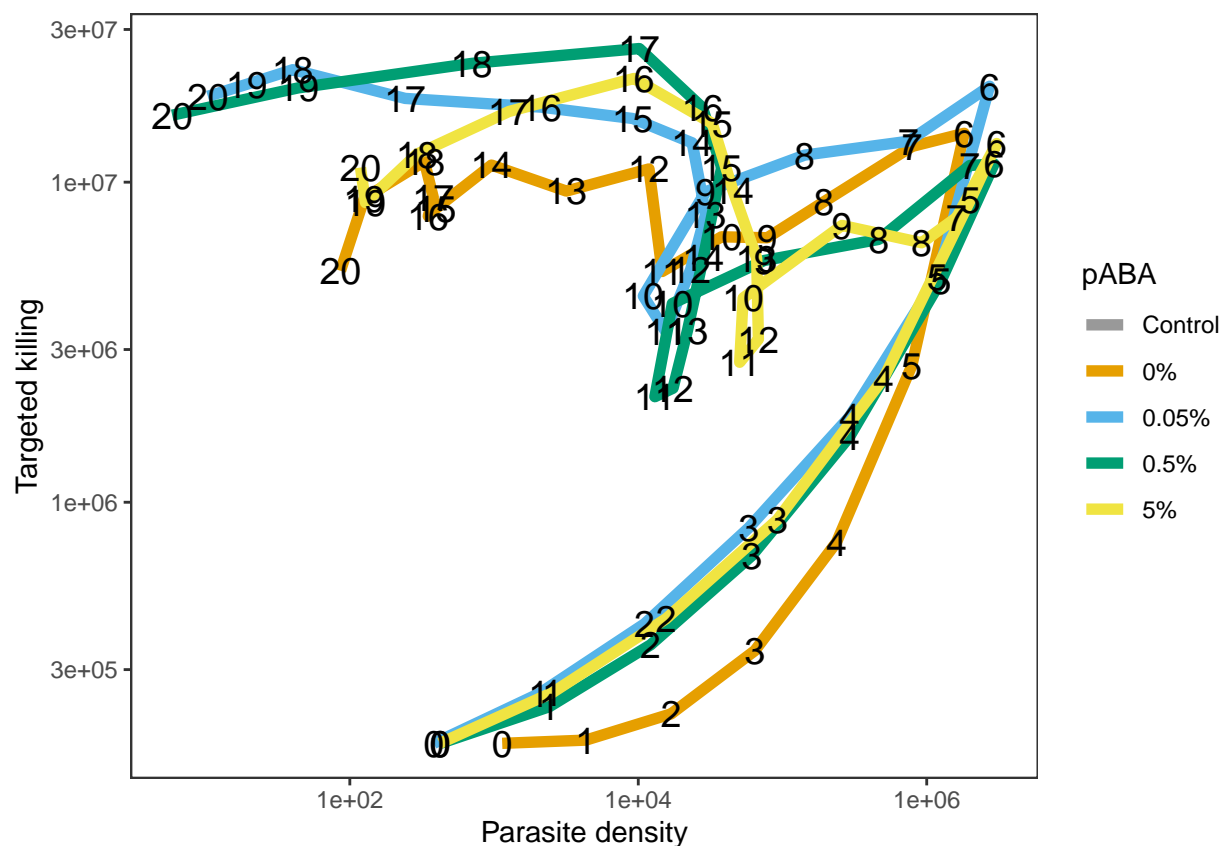
```
swirls |>
  ggplot(aes(x=K,y=W,color=pABA))+
  geom_path(linewidth=2)+
  geom_text(aes(label=time),col="black",size=5)+
  scale_x_log10()+scale_y_log10()+
  scale_colour_manual(values=cbPalette)+
  xlab("Parasite density")+ylab("Targeted killing")+
  theme_bw()+
  theme(
    panel.grid = element_blank()
  )
```

## Warning: Transformation introduced infinite values in continuous x-axis

## Transformation introduced infinite values in continuous x-axis

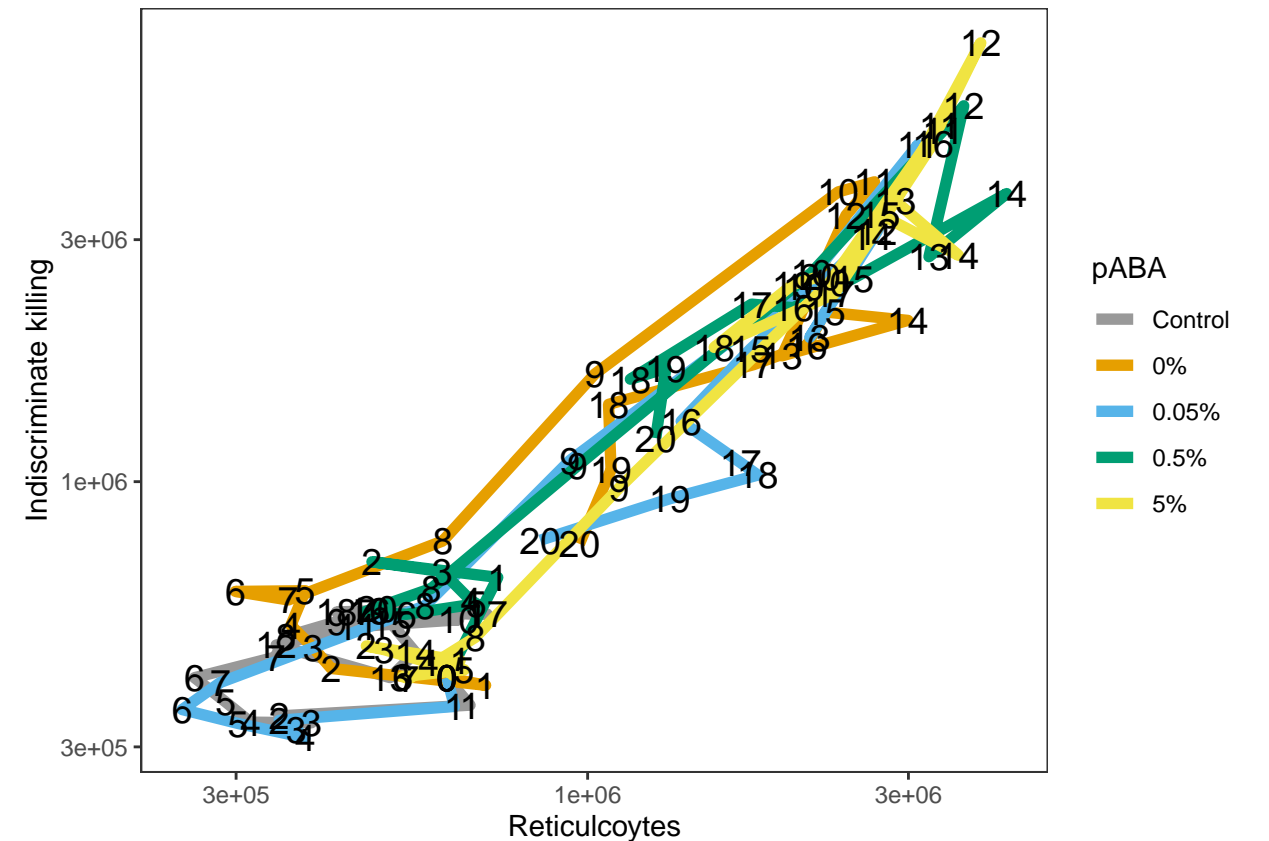
## Warning: Removed 21 rows containing missing values ('geom\_path()').

## Warning: Removed 21 rows containing missing values ('geom\_text()').



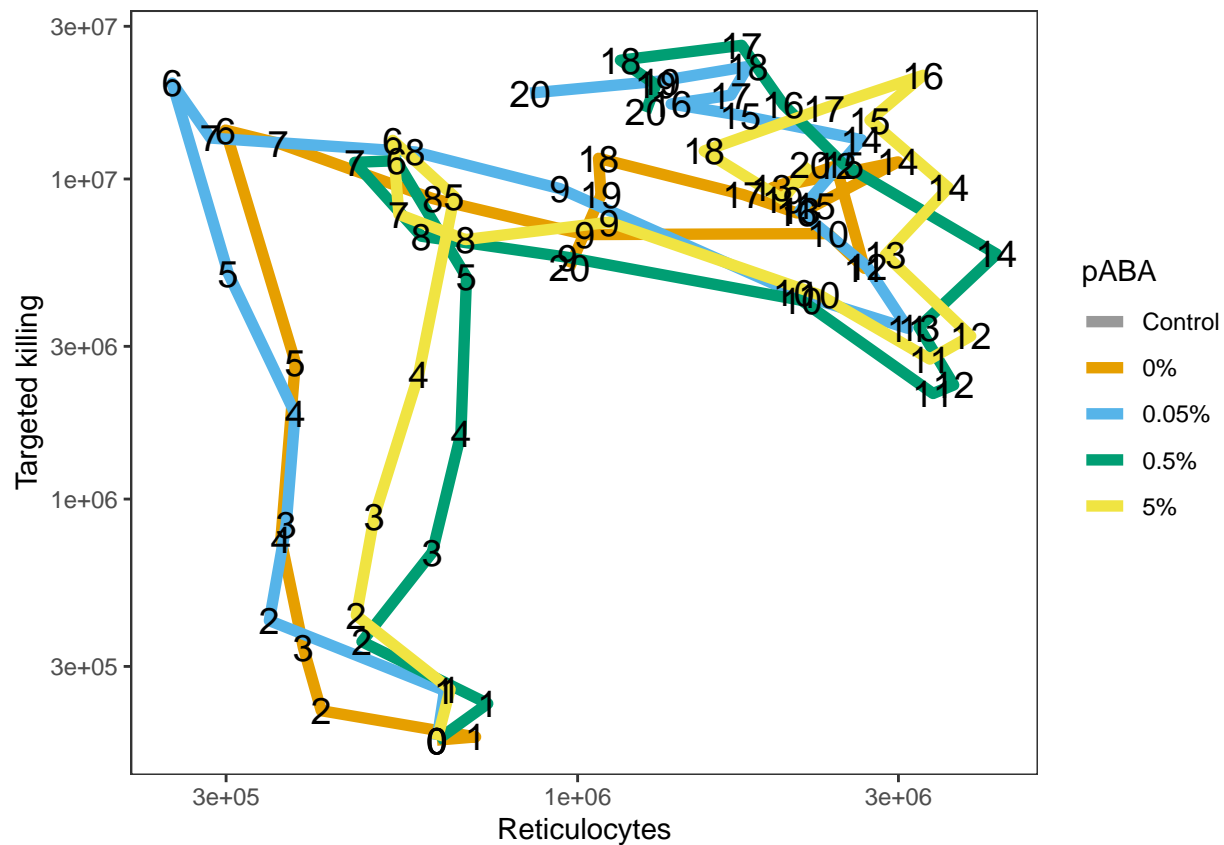
```
swirls |>
  ggplot(aes(x=R,y=N,color=pABA))+
  geom_path(linewidth=2)+
  geom_text(aes(label=time),col="black",size=5)+
  scale_x_log10()+scale_y_log10()+
```

```
scale_colour_manual(values=cbPalette)+
xlab("Reticulocytes")+ylab("Indiscriminate killing")+
theme_bw()+
theme(
  panel.grid = element_blank()
)
```



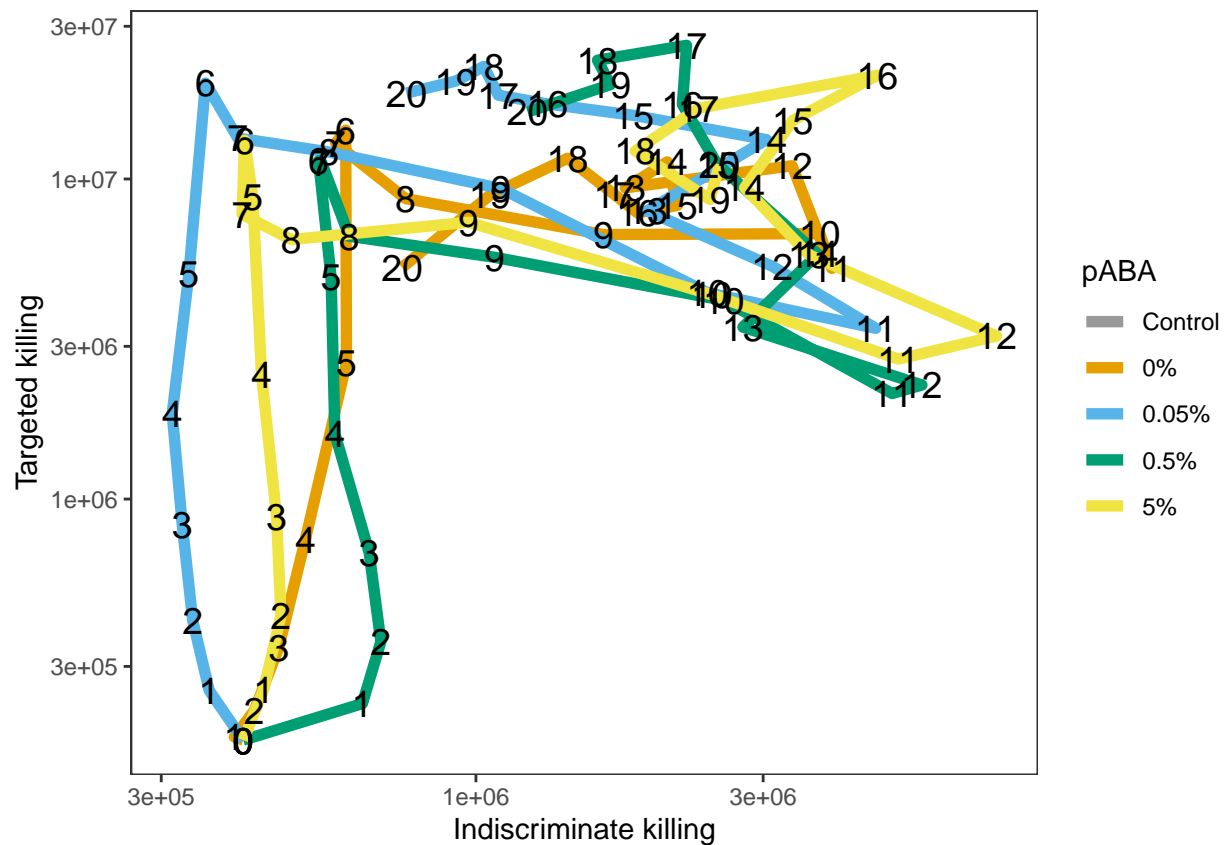
```
swirls |>
  ggplot(aes(x=R,y=W,color=pABA))+
  geom_path(linewidth=2)+
  geom_text(aes(label=time),col="black",size=5)+
  scale_x_log10()+scale_y_log10()+
  scale_colour_manual(values=cbPalette)+
  xlab("Reticulocytes")+ylab("Targeted killing")+
  theme_bw()+
  theme(
    panel.grid = element_blank()
  )
```

```
## Warning: Removed 21 rows containing missing values ('geom_path()').
## Removed 21 rows containing missing values ('geom_text()').
```



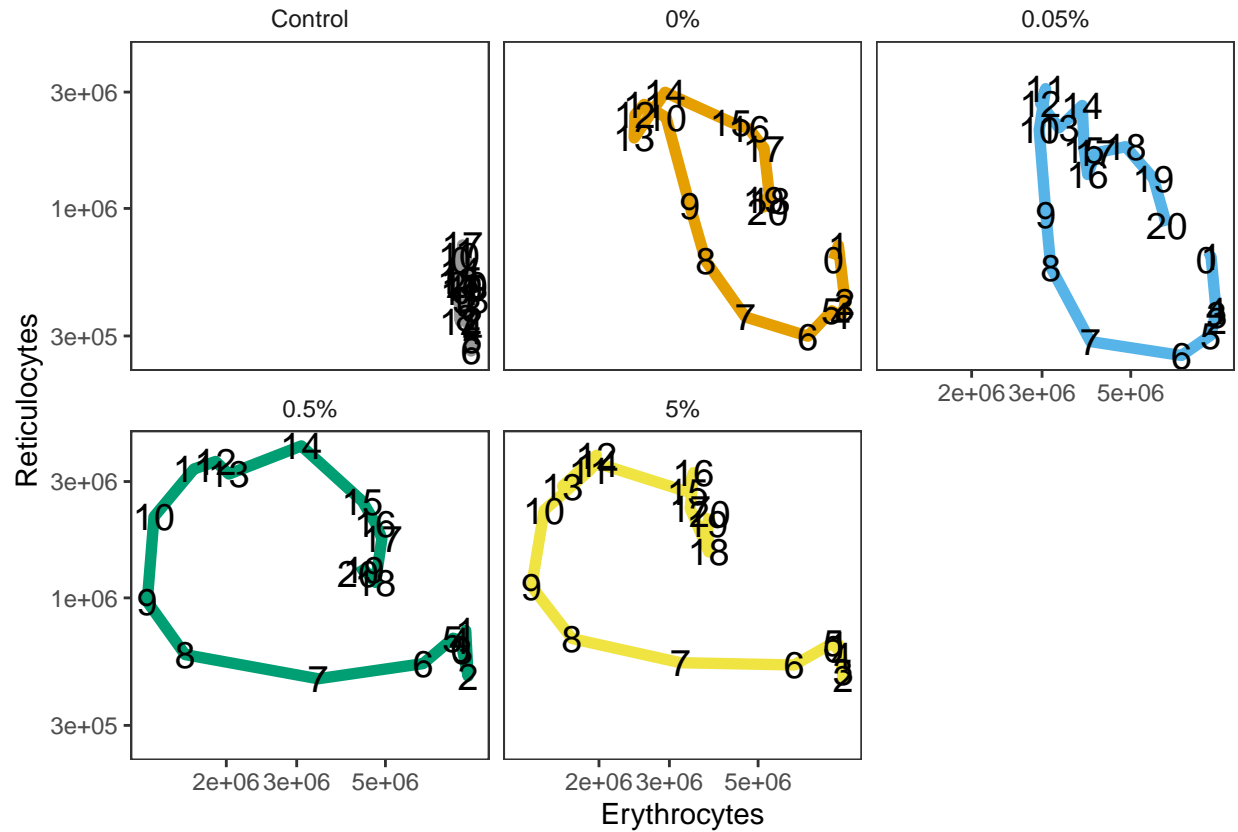
```
swirls |>
  ggplot(aes(x=N,y=W,color=pABA))+
  geom_path(linewidth=2)+
  geom_text(aes(label=time),col="black",size=5)+
  scale_x_log10()+scale_y_log10()+
  scale_colour_manual(values=cbPalette)+
  xlab("Indiscriminate killing")+ylab("Targeted killing")+
  theme_bw()+
  theme(
    panel.grid = element_blank()
  )
```

```
## Warning: Removed 21 rows containing missing values ('geom_path()').
## Removed 21 rows containing missing values ('geom_text()').
```



## Plot reticulocytes versus erythrocytes

```
ggplot()+
  geom_path(data=swirls,aes(x=E,y=R,color=pABA),linewidth=2)+
  geom_text(data=swirls,aes(x=E,y=R,label=time),col="black",size=5)+
  xlab("Erythrocytes")+ylab("Reticulocytes")+
  theme_bw()+
  scale_x_log10()+scale_y_log10()+
  scale_colour_manual(values=cbPalette)+
  facet_wrap(pABA~.)+
  theme(
    panel.grid = element_blank(),
    legend.position = "none",
    strip.background = element_blank()
  )
```



```
ggplot()+
  geom_path(data=swirls,aes(x=E,y=R,color=pABA),linewidth=2)+
  geom_text(data=swirls,aes(x=E,y=R,label=time),col="black",size=5)+
  xlab("Erythrocytes")+ylab("Reticulocytes")+
  theme_bw()+
  scale_colour_manual(values=cbPalette)+
  facet_wrap(pABA~.)+
  theme(
    panel.grid = element_blank(),
    legend.position = "none",
    strip.background = element_blank()
  )
```

