Supplemental Material

Alexander Lalejini, Austin J. Ferguson, Nkrumah Grant, and Charles Ofria

2021-03-31

Contents

Chapter 1

Introduction

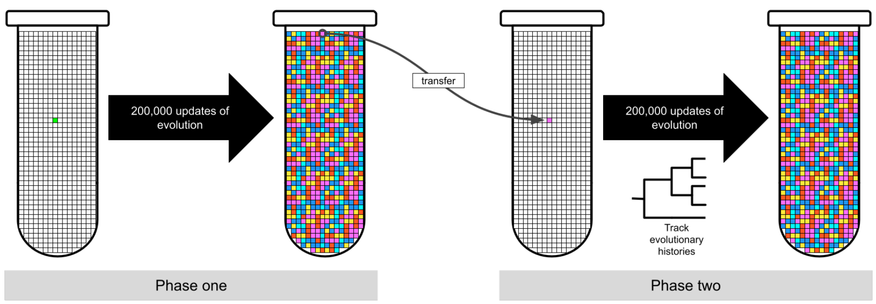


Figure 1.1: Experimental design overview

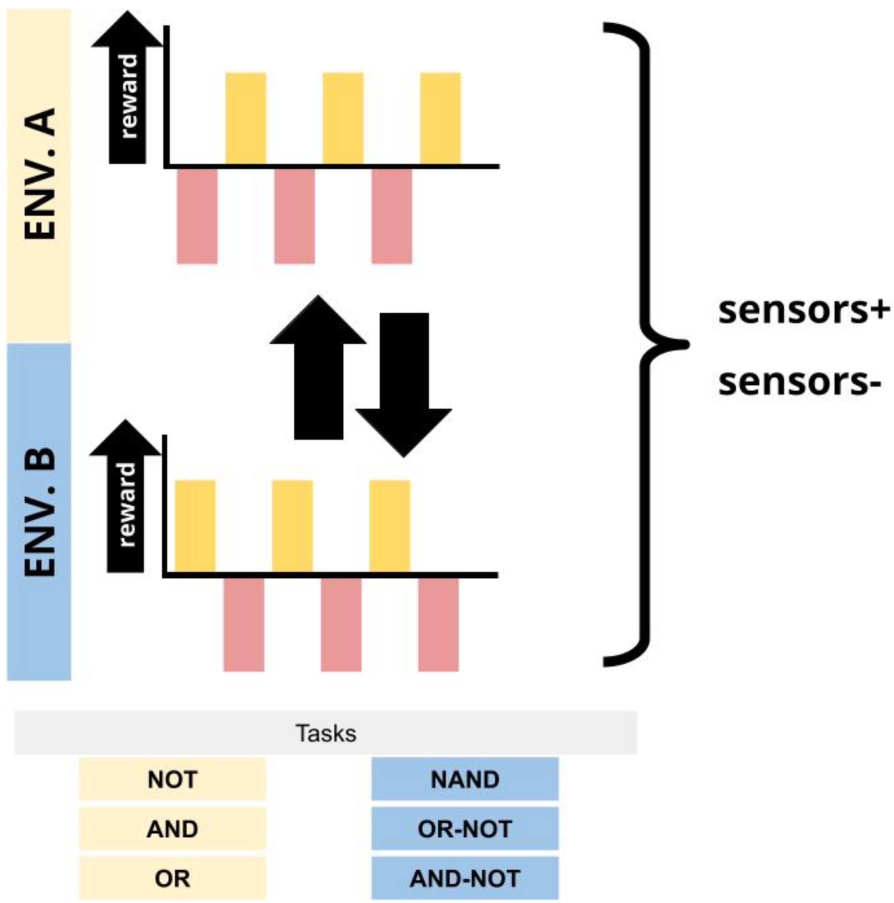


Figure 1.2: Fluctuating environment

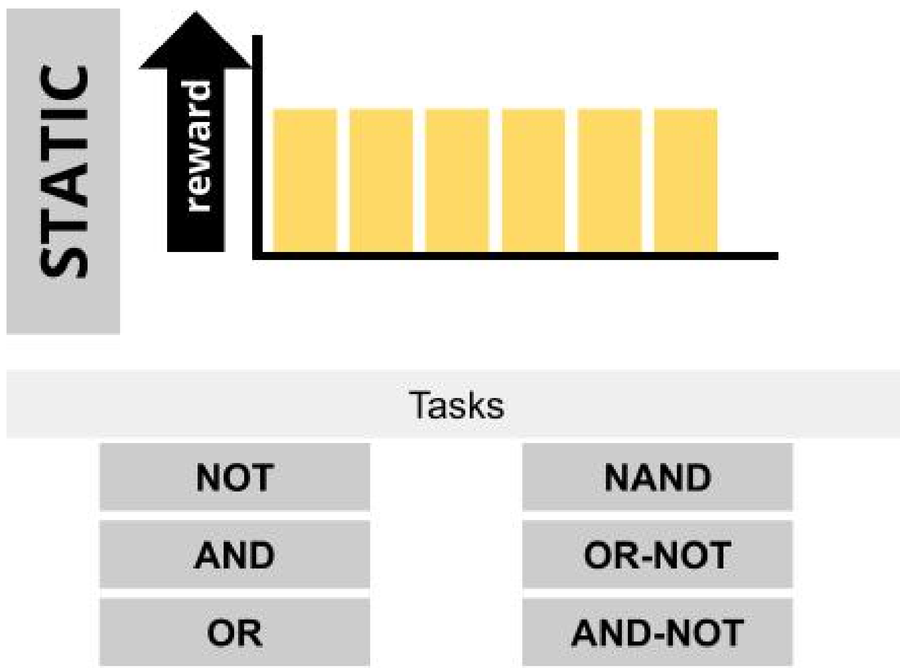


Figure 1.3: Static environment

Chapter 2

Validation experiment

In this experiment, we validate that (1) we observe the evolution of phenotypic plasticity in a changing environment when digital organisms have access to sensory instructions (capable of differentiating environmental states) and (2) that adaptive phenotypic plasticity does not evolve when populations lack access to sensory instructions.

2.1 Overview

```
total_updates <- 200000
replicates <- 100

all_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")

# Relative location of data.
working_directory <- "experiments/2021-01-07-validation/analysis/" # << For bookdown
# working_directory <- "./"                                                # << For local analysis
```

We evolved populations of digital organisms under four conditions:

1. A fluctuating environment with access to sensory instructions
2. A fluctuating environment without access to sensory instructions (i.e., sensory instructions are no-operations)
3. A constant environment with access to sensory instructions
4. A constant environment without access to sensory instructions

In fluctuating environments, we alternate between rewarding and punishing different sets of computational tasks. In one environment, we reward tasks not, and, or and punish tasks nand, ornot, andnot. In the alternative environment, we reward tasks nand, ornot, andnot and punish tasks not, and, or. In constant environments, we reward all tasks (not, nand, and, ornot, or, andnot).

For each replicate of each condition, we extract the dominant (i.e., most numerous) genotype at the end of the run to analyze further. We expect to observe the evolution of adaptive phenotypic plasticity in only the first experimental condition. In conditions without sensors, plasticity in any form should be unable to evolve.

2.2 Analysis dependencies

Load all required R libraries.

```
library(ggplot2)
library(tidyverse)
library(cowplot)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd9
```

These analyses were conducted/knitted with the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

2.3 Setup

```
data_loc <- paste0(working_directory, "data/aggregate.csv")
data <- read.csv(data_loc, na.strings="NONE")

data$DISABLE_REACTION_SENSORS <- as.factor(data$DISABLE_REACTION_SENSORS)
```

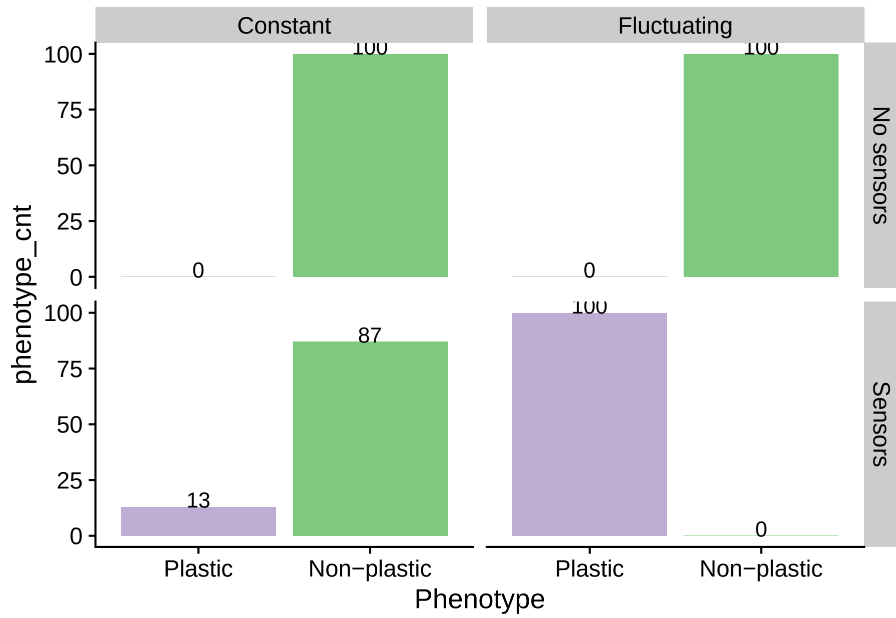
```
data$chg_env <- as.factor(data$chg_env)
data$dom_plastic_odd_even <- as.factor(data$dom_plastic_odd_even)
data$sensors <- data$DISABLE_REACTION_SENSORS == "0"
data$is_plastic <- data$dom_plastic_odd_even == "True"

env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
    return("Sensors")
  } else {
    return("No sensors")
  }
}

# Count observed plasticity for each condition (I'm sure there's a 'tidier' way to do this..)
observed_plasticity <- data.frame(
  environment=character(),
  sensors=character(),
  plastic=integer(),
  nonplastic=integer(),
  plastic_adaptive=integer(),
  plastic_optimal=integer(),
  plastic_nonadaptive=integer()
)
for (env_chg in levels(data$chg_env)) {
  for (disabled_sensors in levels(data$DISABLE_REACTION_SENSORS)) {
    cond_data <- filter(data, chg_env == env_chg & data$DISABLE_REACTION_SENSORS == disabled_sens
    environment_label <- env_label_fun(env_chg)
    sensors_label <- sensors_label_fun(disabled_sensors == "0")

    observed_plasticity <- observed_plasticity %>% add_row(
      environment=environment_label,
      sensors=sensors_label,
      plastic=nrow(filter(cond_data, is_plastic==TRUE)),
      nonplastic=nrow(filter(cond_data, is_plastic==FALSE)),
      plastic_adaptive=nrow(filter(cond_data, dom_adaptive_plasticity=="True")),
      plastic_optimal=nrow(filter(cond_data, dom_optimal_plastic=="True")),
      plastic_nonadaptive=nrow(filter(cond_data, is_plastic==TRUE & dom_adaptive_plasticity=="Fal
```

```
    )
  }
}

observed_plasticity <- pivot_longer(
  observed_plasticity,
  cols=c("plastic", "plastic_adaptive", "plastic_optimal", "plastic_nonadaptive", "nonp
  names_to="phenotype",
  values_to="phenotype_cnt"
)

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
```

2.4 Evolution of phenotypic plasticity

For each experimental condition, do we observe the evolution of phenotypic plasticity? To test for phenotypic plasticity, we culture digital organisms in both environments from the fluctuating condition (including organisms evolved in a constant environment). Any plasticity that we observe from digital organisms evolved under constant conditions is cryptic variation (as these organisms were never exposed to these culturing environments).

```
ggplot(filter(observed_plasticity, phenotype %in% c("plastic", "nonplastic")), aes(x=pl
  geom_bar(
    stat="identity",
    position=position_dodge(0.9)
  ) +
  geom_text(
    stat="identity",
    mapping=aes(label=phenotype_cnt),
    vjust=0.05
  ) +
  scale_fill_brewer(palette="Accent") +
  scale_x_discrete(
    name="Phenotype",
    limits=c("plastic", "nonplastic"),
    labels=c("Plastic", "Non-plastic")
  ) +
  facet_grid(sensors~environment) +
  theme(
    legend.position="none"
  )
```

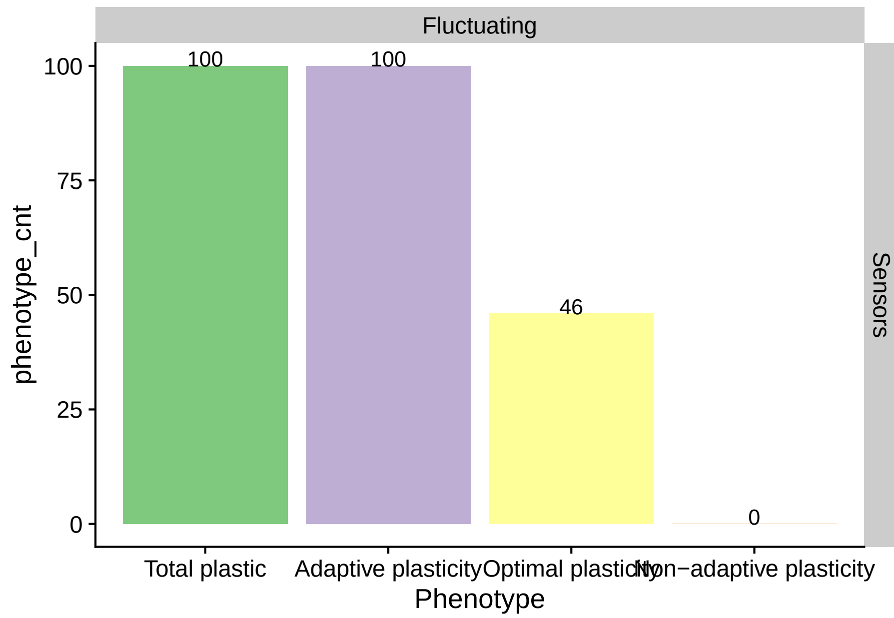


Indeed, we do not observe the evolution of phenotypic plasticity in any replicates in which digital organisms do not have access to sensory instructions. We do observe the evolution of plasticity (not necessarily adaptive plasticity) in both constant and fluctuating environments where sensors are enabled.

To what extent is the observed phenotypic plasticity adaptive?

```
ggplot(filter(observed_plasticity, environment=="Fluctuating" & sensors == "Sensors" & phenotype
  geom_bar(
    stat="identity",
    position=position_dodge(0.9)
  ) +
  geom_text(
    stat="identity",
    mapping=aes(label=phenotype_cnt),
    vjust=0.05
  ) +
  scale_fill_brewer(palette="Accent") +
  scale_x_discrete(
    name="Phenotype",
    limits=c("plastic",  "plastic_adaptive", "plastic_optimal", "plastic_nonadaptive"),
    labels=c("Total plastic", "Adaptive plasticity", "Optimal plasticity", "Non-adaptive plastici
  ) +
  facet_grid(sensors~environment) +
  theme(
    legend.position="none"
```

```
)
```


Fluctuating
Sensors
phenotype_cnt
100
75
50
25
0
100
100
46
0
Total plastic
Adaptive plasticity
Optimal plasticity
Non−adaptive plasticity
Phenotype

Chapter 3

Evolutionary change

The effect of adaptive phenotypic plasticity on evolutionary change.

3.1 Overview

```
total_updates <- 200000
replicates <- 100
alpha <- 0.05

all_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")

# Relative location of data.
working_directory <- "experiments/2021-02-08-evo-dynamics/analysis/" # << For bookdown
# working_directory <- "./"                                              # << For local analysis
```

3.2 Analysis dependencies

Load all required R libraries.

```
library(ggplot2)
library(rstatix)
library(ggsignif)
library(scales)
library(tidyverse)
library(cowplot)
library(RColorBrewer)
library(Hmisc)
```

```
library(boot)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97
```

These analyses were conducted/knitted with the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

3.3 Setup

```
summary_data_loc <- paste0(working_directory, "data/aggregate.csv")
summary_data <- read.csv(summary_data_loc, na.strings="NONE")

summary_data$DISABLE_REACTION_SENSORS <- as.factor(summary_data$DISABLE_REACTION_SENSOR
summary_data$chg_env <- summary_data$chg_env == "True"
summary_data$dominant_plastic_odd_even <- as.factor(summary_data$dominant_plastic_odd_e
summary_data$sensors <- summary_data$DISABLE_REACTION_SENSORS == "0"
summary_data$is_plastic <- summary_data$dominant_plastic_odd_even == "True"

env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
```

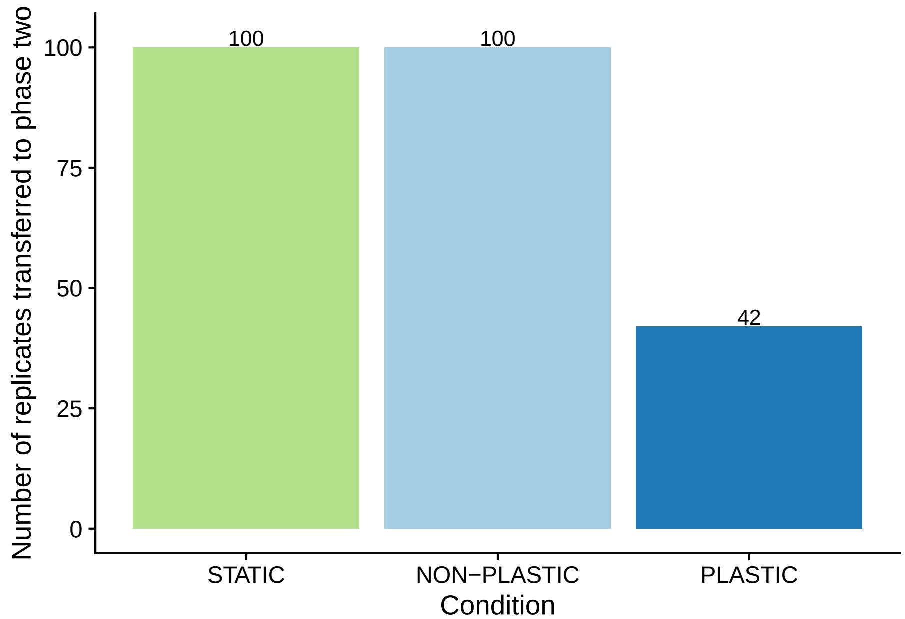
```r
    return("Sensors")
  } else {
    return("No sensors")
  }
}

# note that this labeler makes assumptions about how we set up our experiment
condition_label_fun <- function(has_sensors, env_chg) {
  if (has_sensors && env_chg) {
    return("PLASTIC")
  } else if (env_chg) {
    return("NON-PLASTIC")
  } else {
    return("STATIC")
  }
}

summary_data$env_label <- mapply(
  env_label_fun,
  summary_data$chg_env
)
summary_data$sensors_label <- mapply(
  sensors_label_fun,
  summary_data$sensors
)
summary_data$condition <- mapply(
  condition_label_fun,
  summary_data$sensors,
  summary_data$chg_env
)

condition_order = c(
  "STATIC",
  "NON-PLASTIC",
  "PLASTIC"
)
pairwise_comparisons <- list(
  c("STATIC", "NON-PLASTIC"),
  c("STATIC", "PLASTIC"),
  c("PLASTIC", "NON-PLASTIC")
)

p_label <- function(p_value) {
  threshold = 0.0001
  if (p_value < threshold) {
```

```
    return(paste0("p < ", threshold))
  } else {
    return(paste0("p = ", p_value))
  }
}

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
# Palette
cb_palette <- "Paired"
# Create a directory to store plots
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
# Define sample mean function
samplemean <- function(x, d) {
  return(mean(x[d]))
}
```

3.4 The evolution of phenotypic plasticity

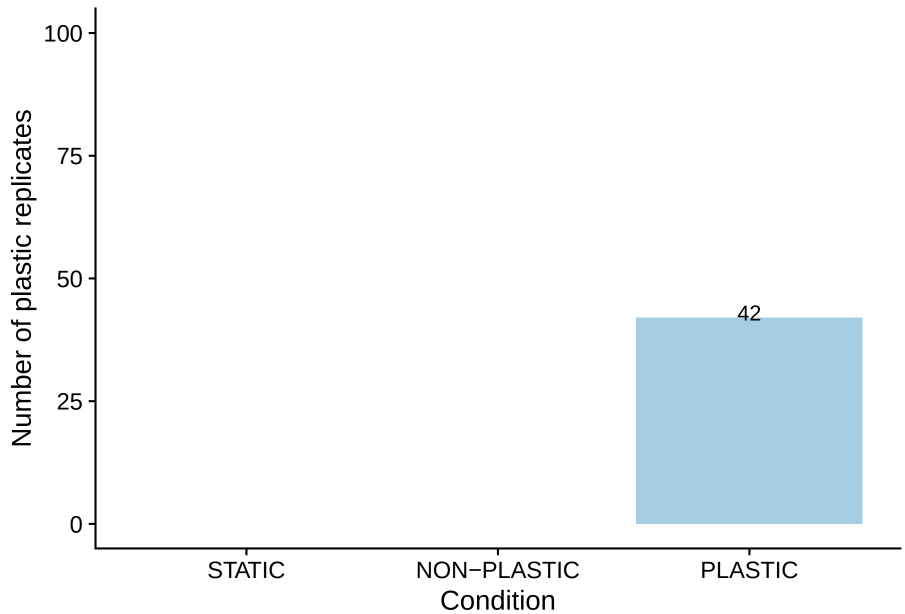
For sensor-enabled populations in fluctuating environments, we only transfered populations containing an optimally plastic genotype to phase-two.

```
summary_data_grouped = dplyr::group_by(summary_data, condition)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())

ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  ylab("Number of replicates transferred to phase two") +
  theme(
    legend.position="none"
  )
```



We can confirm our expectation that the dominant genotypes in non-plastic conditions are not phenotypically plastic.

```
summary_data_grouped = dplyr::group_by(summary_data, condition, is_plastic)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
ggplot(filter(summary_data_group_counts, is_plastic), aes(x=condition, y=n, fill=condition)) +
  geom_col(
    position=position_dodge(0.9)
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  geom_text(aes(label=n, y=n+1)) +
  ylab("Number of plastic replicates") +
  ylim(0, 100) +
  theme(
    legend.position="none"
  )
```

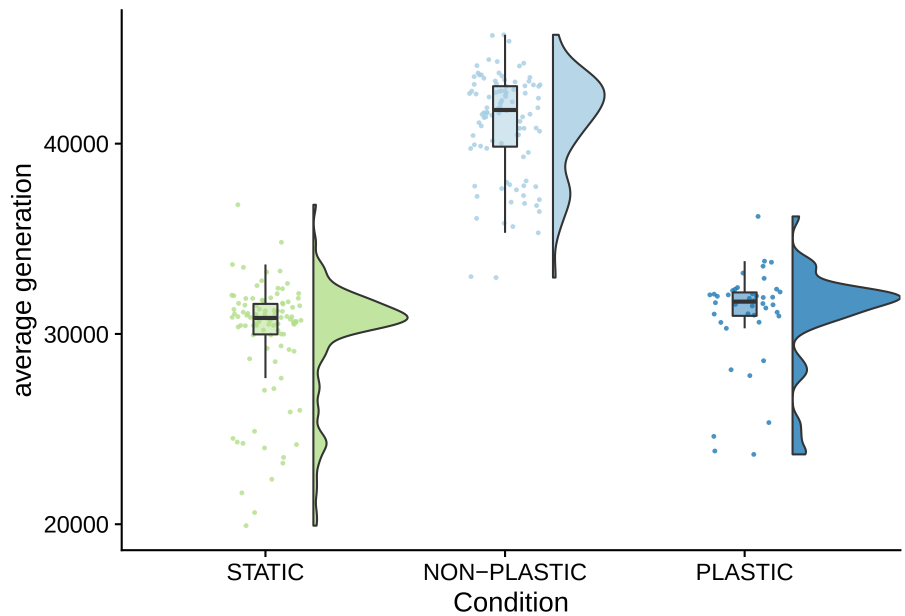


3.5 Average generation

```
ggplot(summary_data, aes(x=condition, y=time_average_generation, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
```

```
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  ylab("average generation") +
  theme(
    legend.position="none"
  ) +
  ggsave(paste0(working_directory, "plots/", "average-generation.png"))
```

```
## Saving 6.5 x 4.5 in image
```




```
kruskal.test(
  formula=time_average_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  time_average_generation by condition
## Kruskal-Wallis chi-squared = 177.33, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$time_average_generation,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$time_average_generation and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.004
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$time_average_generation)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$time_average_generation)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$time_average_generation)
  )
)
```

```
## [1] "PLASTIC median: 31697.65; STATIC median: 30839.75; NON-PLASTIC median: 41768.65
```

```r
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```r
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=time_average_generation~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
```

```
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=9982"
## [1] "STATIC<-->PLASTIC: W=2818"
## [1] "PLASTIC<-->NON-PLASTIC: W=4186"
```

3.6 Coalescence event count

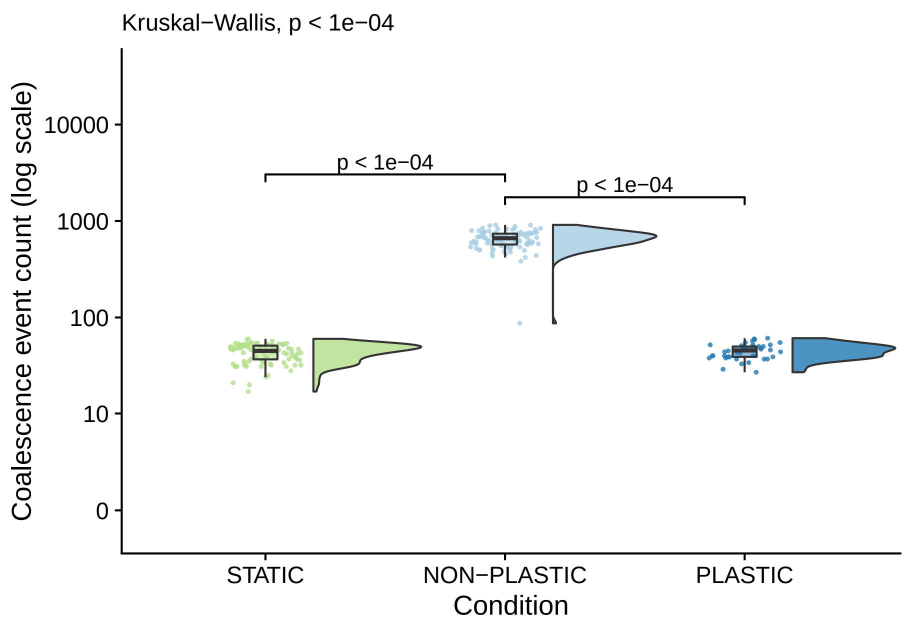
The number of times the most recent common ancestor changes gives us the number of selective sweeps that occur during the experiment.

```
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(phylo_mrca_changes ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition",step.increase=1)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <-   log10(stat.test$y.position) * c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

coalescence_events_fig <- ggplot(
    summary_data,
    aes(x=condition, y=phylo_mrca_changes,fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
```

```
    labels=condition_order,
    breaks=condition_order
  ) +
  scale_y_continuous(
    name="Coalescence event count (log scale)",
    trans=pseudo_log_trans(sigma = 1, base = 10),
    breaks=c(0, 10, 100, 1000, 10000),
    limits=c(-1, 35000)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=phylo_mrca_changes~condition, data=summary_da
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "selective-sweeps.pdf"),
    width=5,
    height=5
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
coalescence_events_fig
```




```
kruskal.test(
  formula=phylo_mrca_changes~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  phylo_mrca_changes by condition
## Kruskal-Wallis chi-squared = 175.46, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$phylo_mrca_changes,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$phylo_mrca_changes and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      1
##
```

```
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$phylo_mrca_changes)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$phylo_mrca_changes)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$phylo_mrca_changes)
  )
)
```

```
## [1] "PLASTIC median: 45.5; STATIC median: 45; NON-PLASTIC median: 663.5"
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=phylo_mrca_changes~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=10000"
## [1] "STATIC<-->PLASTIC: W=2215"
## [1] "PLASTIC<-->NON-PLASTIC: W=4200"
```

3.6.1 Average number of generations between coalescence events

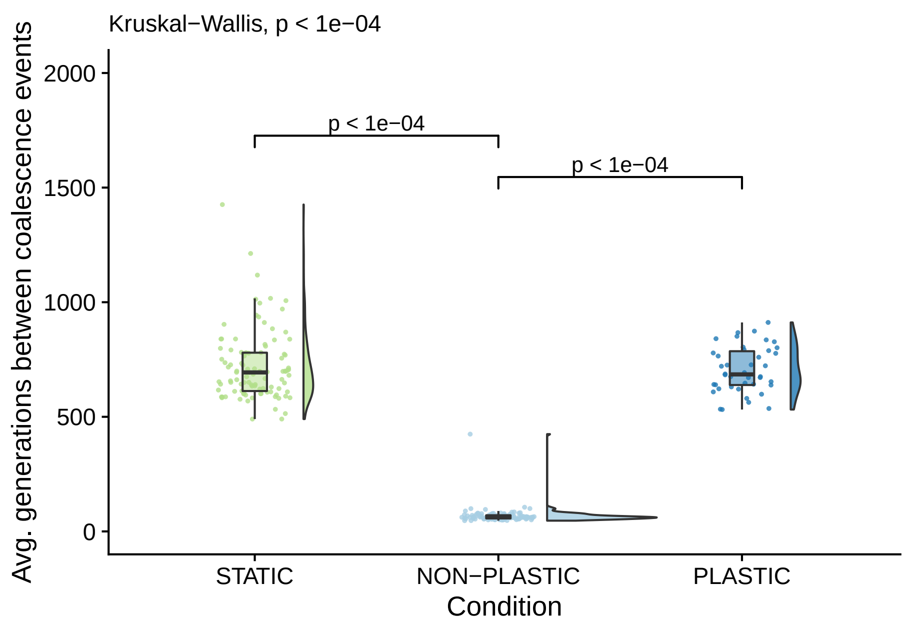
```
# Compute frequency of coalescence events
summary_data$generations_per_mrca_change <- summary_data$time_average_generation / summ

# Compute manual labels for geom_signif
```

```
stat.test <- summary_data %>%
  wilcox_test(generations_per_mrca_change ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition")
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <-  stat.test$y.position
stat.test$label <- mapply(p_label,stat.test$p.adj)

coalescence_events_freq_fig <- ggplot(
    summary_data,
    aes(x=condition, y=generations_per_mrca_change, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Avg. generations between coalescence events",
    limits=c(0, 2000),
    breaks=seq(0, 2000, 500)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
```

```
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=generations_per_mrca_change~condition, data=s
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "generations-between-selective-sweeps.png"),
    width=5,
    height=5
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
coalescence_events_freq_fig
```



```r
kruskal.test(
  formula=generations_per_mrca_change~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  generations_per_mrca_change by condition
## Kruskal-Wallis chi-squared = 175.33, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$generations_per_mrca_change,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$generations_per_mrca_change and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      1
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$generations_per_mrca_change)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$generations_per_mrca_change)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$generations_per_mrca_change)
  )
)
```

```
## [1] "PLASTIC median: 685.001780758557; STATIC median: 693.676265008576; NON-PLASTIC median: 62
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=generations_per_mrca_change~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=0"
## [1] "STATIC<-->PLASTIC: W=2151"
## [1] "PLASTIC<-->NON-PLASTIC: W=0"
```

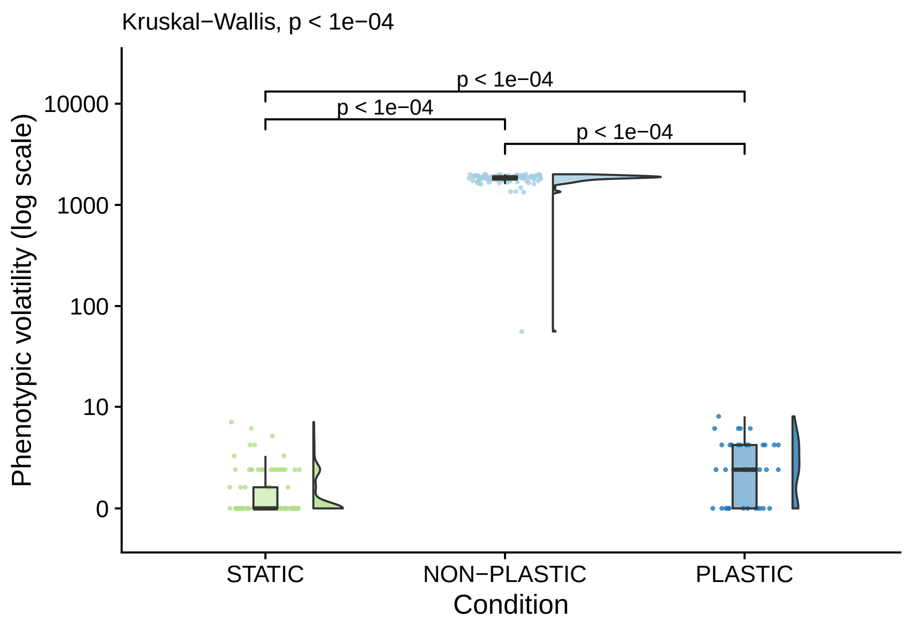
3.7 Phenotypic volatility along the dominant lineage

```
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_trait_volatility ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition", step.increase=1)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad be
stat.test$manual_position <-  log10(stat.test$y.position) * c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

phenotypic_volatility_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_trait_volatility, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
```

```
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Phenotypic volatility (log scale)",
    trans=pseudo_log_trans(sigma = 1, base = 10),
    breaks=c(0, 10, 100, 1000, 10000),
    limits=c(-1, 21000)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_trait_volatility~condition, data=summa
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj<=alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "phenotypic-volatility.pdf"),
    width=5,
    height=5
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
phenotypic_volatility_fig
```




```
kruskal.test(
  formula=dominant_lineage_trait_volatility~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_trait_volatility by condition
## Kruskal-Wallis chi-squared = 190.78, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_volatility,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_trait_volatility and summary_data$condition
##
##         NON-PLASTIC PLASTIC
```

```
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     8.7e-07
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_trait_volatility)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_trait_volatility)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_trait_volatility)
  )
)
```

```
## [1] "PLASTIC median: 2; STATIC median: 0; NON-PLASTIC median: 1868"
```

```
print("Wilcox rank sum test statistics:")
```

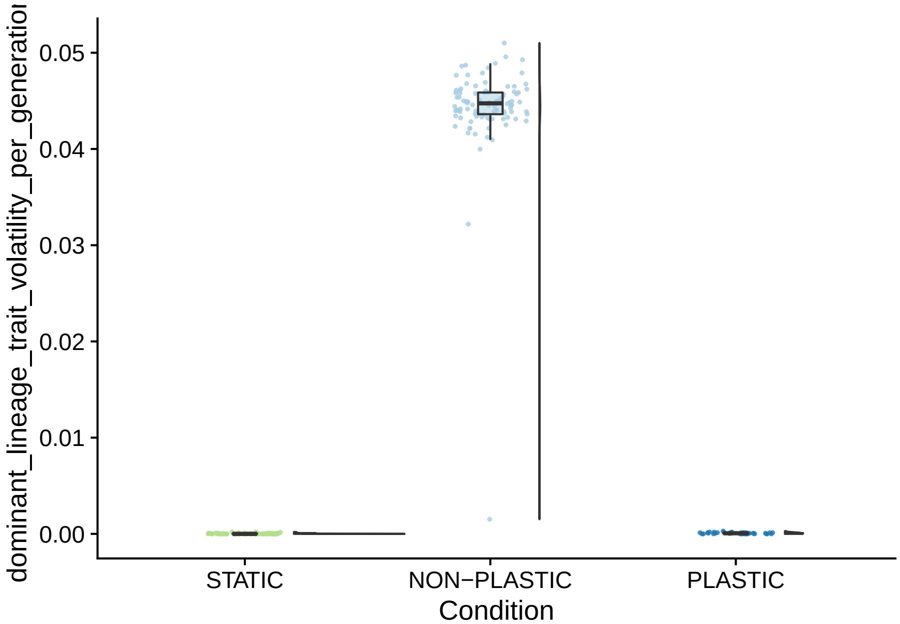
```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_trait_volatility~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=10000"
## [1] "STATIC<-->PLASTIC: W=3116.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=4200"
```

3.7.1 Phenotypic volatility normalized by generations elapsed

```
summary_data$dominant_lineage_trait_volatility_per_generation <- summary_data$dominant_

ggplot(summary_data, aes(x=condition, y=dominant_lineage_trait_volatility_per_generatio
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  )
```




```r
kruskal.test(
  formula=dominant_lineage_trait_volatility_per_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_trait_volatility_per_generation by condition
## Kruskal-Wallis chi-squared = 189.62, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_volatility_per_generation,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_trait_volatility_per_generation and summary_data$conditio
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     4.2e-06
##
```

```
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_trait_volatility
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_trait_volatility_
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_trait_volati
  )
)
```

```
## [1] "PLASTIC median: 6.33339279717772e-05; STATIC median: 0; NON-PLASTIC median: 0.0
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_trait_volatility_per_generation~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=10000"
## [1] "STATIC<-->PLASTIC: W=3061.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=4200"
```

3.8 Phenotypic fidelity

Frequency that an offspring's genotype is identical to a parent genotype (along the dominant lineage).

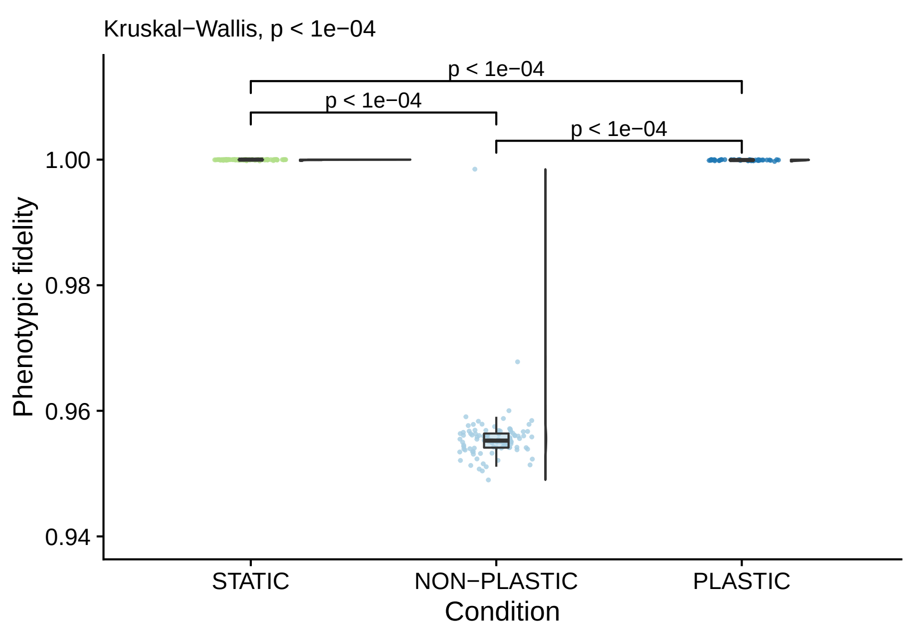
```
summary_data$dominant_lineage_trait_fidelity <- (summary_data$dominant_generation_born

# Compute manual labels for geom_signif
```

```
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_trait_fidelity ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition",step.increase=1.5)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <-  stat.test$y.position * c(1.0,1.0,1.0005)
stat.test$label <- mapply(p_label,stat.test$p.adj)

phenotypic_fidelity_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_trait_fidelity, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Phenotypic fidelity",
    limits=c(0.94, 1.013),
    breaks=c(0.94, 0.96, 0.98, 1.0) #seq(0.94, 1.0, 0.01)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
```

```
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_trait_fidelity~condition, da
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "phenotypic-fidelity.pdf"),
    width=5,
    height=5
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
phenotypic_fidelity_fig
```



```r
kruskal.test(
  formula=dominant_lineage_trait_fidelity~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_trait_fidelity by condition
## Kruskal-Wallis chi-squared = 189.62, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_fidelity,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_trait_fidelity and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     4.2e-06
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_trait_fidelity)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_trait_fidelity)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_trait_fidelity)
  )
)
```

```
## [1] "PLASTIC median: 0.999936666072028; STATIC median: 1; NON-PLASTIC median: 0.95525598543618
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_trait_fidelity~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=0"
## [1] "STATIC<-->PLASTIC: W=1138.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=0"
```

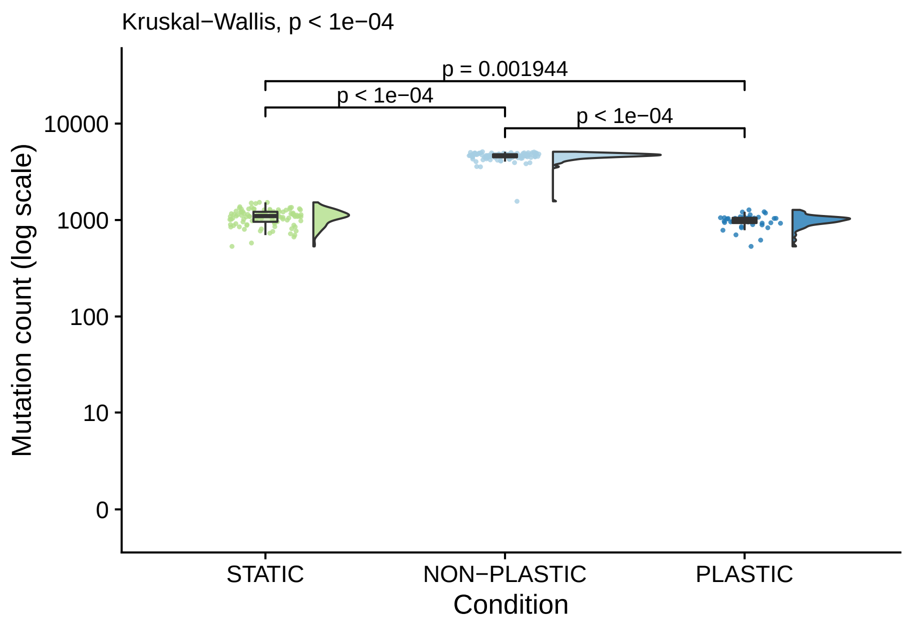
3.9 Mutation count

```
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_total_mut_cnt ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition",step.increase=1)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad be
stat.test$manual_position <- log10(stat.test$y.position) *  c(1.0,1.0,1.03) # c(1.0,1.
stat.test$label <- mapply(p_label,stat.test$p.adj)

mutation_count_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_total_mut_cnt, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
```

```
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Mutation count (log scale)",
    trans=pseudo_log_trans(sigma = 1, base = 10),
    breaks=c(0, 10, 100, 1000, 10000),
    limits=c(-1, 35000)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_total_mut_cnt~condition, data=summary_
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "mutation-accumulation.pdf"),
    width=5,
    height=4
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
mutation_count_fig
```




```
kruskal.test(
  formula=dominant_lineage_total_mut_cnt~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_total_mut_cnt by condition
## Kruskal-Wallis chi-squared = 179.33, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_total_mut_cnt,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_total_mut_cnt and summary_data$condition
##
##             NON-PLASTIC PLASTIC
```

```
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.0019
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_total_mut_cnt)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_total_mut_cnt)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_total_mut_cnt)
  )
)
```

```
## [1] "PLASTIC median: 998.5; STATIC median: 1100; NON-PLASTIC median: 4657.5"
```

```r
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

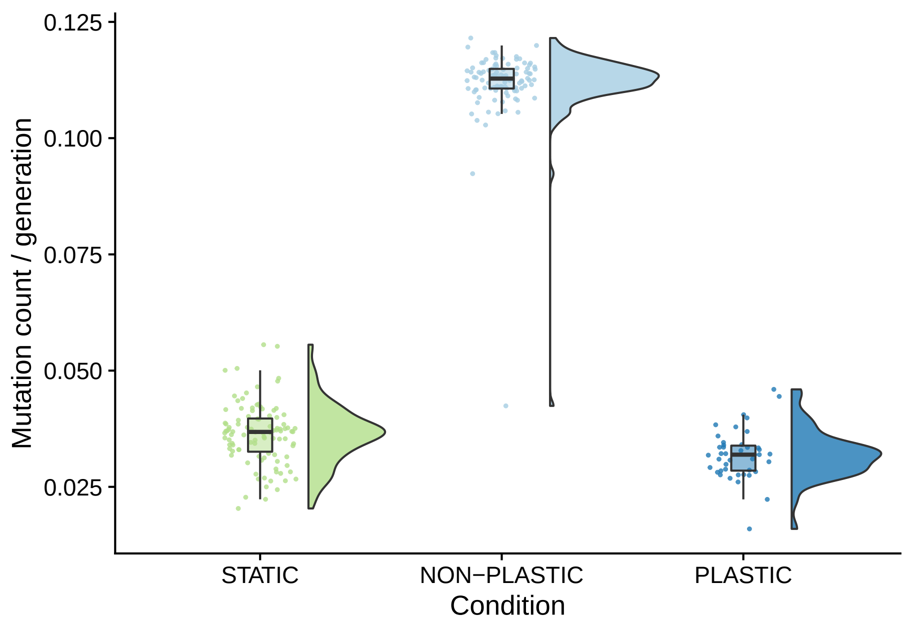
```r
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_total_mut_cnt~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=10000"
## [1] "STATIC<-->PLASTIC: W=1336.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=4200"
```

3.9.1 Mutation count normalized by generations elapsed

```r
summary_data$mutations_per_generation <- summary_data$dominant_lineage_total_mut_cnt / summary_da

ggplot(summary_data, aes(x=condition, y=mutations_per_generation, fill=condition)) +
```

```
geom_flat_violin(
  position = position_nudge(x = .2, y = 0),
  alpha = .8
) +
geom_point(
  mapping=aes(color=condition),
  position = position_jitter(width = .15),
  size = .5,
  alpha = 0.8
) +
geom_boxplot(
  width = .1,
  outlier.shape = NA,
  alpha = 0.5
) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
ylab("Mutation count / generation") +
scale_fill_brewer(
  palette=cb_palette
) +
scale_color_brewer(
  palette=cb_palette
) +
# coord_flip() +
theme(
  legend.position="none"
)
```




```r
kruskal.test(
  formula=mutations_per_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  mutations_per_generation by condition
## Kruskal-Wallis chi-squared = 180.11, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$mutations_per_generation,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$mutations_per_generation and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      2e-04
##
```

```
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$mutations_per_generation)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$mutations_per_generation)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$mutations_per_generation)
  )
)
```

```
## [1] "PLASTIC median: 0.0319267181456982; STATIC median: 0.0368157192941933; NON-PLAS
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=mutations_per_generation~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=9987"
## [1] "STATIC<-->PLASTIC: W=1206"
## [1] "PLASTIC<-->NON-PLASTIC: W=4198"
```

3.10 Genotypic fidelity

The frequency that an offspring's genotype is the same as a parent's genotype.

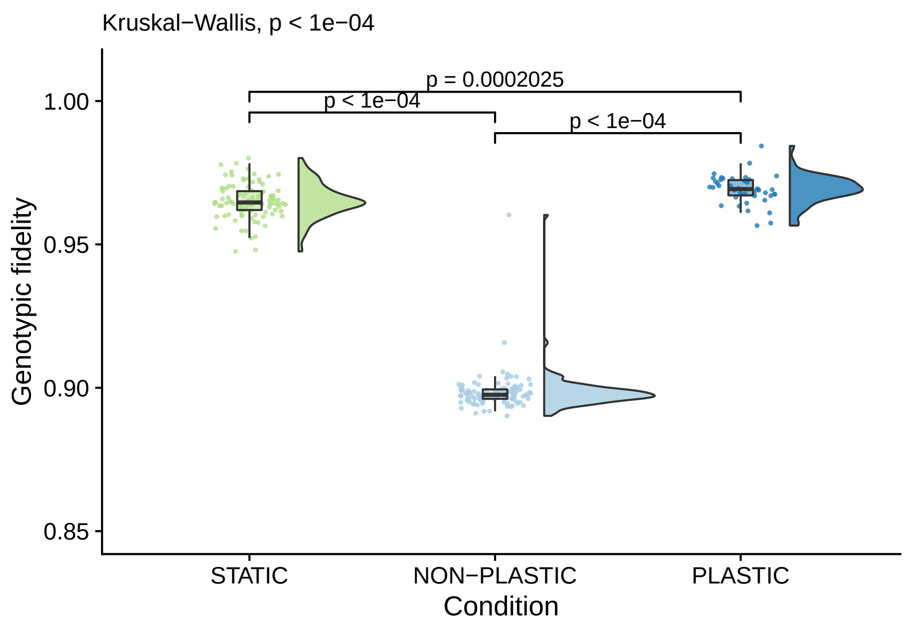
```
summary_data$dominant_lineage_genotypic_fidelity <- (summary_data$dominant_generation_l

# Compute manual labels for geom_signif
stat.test <- summary_data %>%
```

```
  wilcox_test(dominant_lineage_genotypic_fidelity ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition",step.increase=0.2)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <- stat.test$y.position * c(1.0,1.0,1.0)
stat.test$label <- mapply(p_label,stat.test$p.adj)

genotypic_fidelity_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_genotypic_fidelity, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Genotypic fidelity",
    limits=c(0.85, 1.01),
    breaks=c(0.85, 0.90, 0.95, 1.0) #seq(0.85, 1.0, 0.02)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  labs(
```

```
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_genotypic_fidelity~conditio
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "genotypic-fidelity.pdf"),
    width=5,
    height=4
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
genotypic_fidelity_fig
```




```
kruskal.test(
  formula=dominant_lineage_genotypic_fidelity~condition,
```

```
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_genotypic_fidelity by condition
## Kruskal-Wallis chi-squared = 179.86, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_genotypic_fidelity,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_genotypic_fidelity and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      2e-04
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_genotypic_fidelity)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_genotypic_fidelity)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_genotypic_fidelity)
  )
)
```

```
## [1] "PLASTIC median: 0.969286906891951; STATIC median: 0.964620594632577; NON-PLASTIC median:
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_genotypic_fidelity~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=18"
## [1] "STATIC<-->PLASTIC: W=2992"
## [1] "PLASTIC<-->NON-PLASTIC: W=2"
```

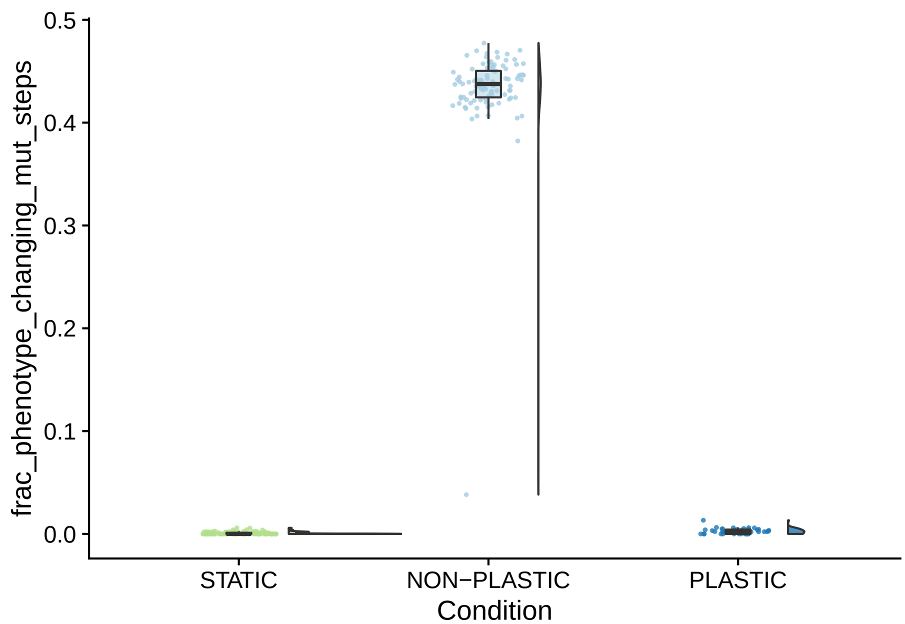
3.11 Characterizing variation along dominant lineages

3.11.1 What fraction of mutations along the lineage result in phenotypic changes?

```
summary_data$frac_phenotype_changing_mut_steps <- summary_data$dominant_lineage_num_mut
ggplot(summary_data, aes(x=condition, y=frac_phenotype_changing_mut_steps, fill=condit
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
```

```
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(paste0(working_directory, "plots/", "frac_phenotype_changing_mutational_steps.png"))
```

```
## Saving 6.5 x 4.5 in image
```




```
kruskal.test(
  formula=frac_phenotype_changing_mut_steps~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_phenotype_changing_mut_steps by condition
## Kruskal-Wallis chi-squared = 191.23, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$frac_phenotype_changing_mut_steps,
```

```
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$frac_phenotype_changing_mut_steps and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     2.3e-07
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$frac_phenotype_changing_mut_steps
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$frac_phenotype_changing_mut_steps)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$frac_phenotype_changing_mut_s
  )
)
```

```
## [1] "PLASTIC median: 0.00224941742616098; STATIC median: 0; NON-PLASTIC median: 0.43
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=frac_phenotype_changing_mut_steps~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
```

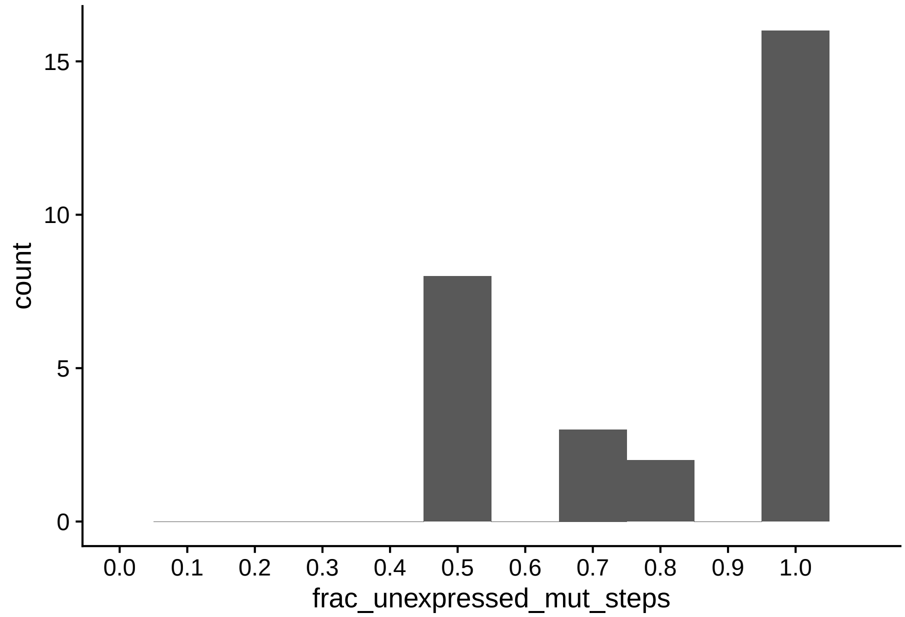
```
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=10000"
## [1] "STATIC<-->PLASTIC: W=3172"
## [1] "PLASTIC<-->NON-PLASTIC: W=4200"
```

3.11.2 For PLASTIC populations, what fraction of phenotype-altering mutations occurred in the unexpressed phenotype?

```
summary_data$frac_unexpressed_mut_steps <- summary_data$dominant_lineage_num_mut_steps_that_chang
summary_data$frac_expressed_mut_steps <- summary_data$dominant_lineage_num_mut_steps_that_change_

ggplot(filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change_agg
  geom_histogram(binwidth=0.1) +
  scale_x_continuous(
    limits=c(0, 1.1),
    breaks=seq(0, 1.0, 0.1)
  ) +
  theme(
    legend.position="none"
  )
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
print(paste0("PLASTIC - Mean with bootstrapped 95% CI"))
```

```
## [1] "PLASTIC - Mean with bootstrapped 95% CI"
```

```
bo <- boot(filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_t
print(bo)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_s
##     0)$frac_unexpressed_mut_steps, statistic = samplemean, R = 10000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.8247126 0.0003025862  0.03989834
```

```
print(boot.ci(bo, conf=0.95, type="perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bo, conf = 0.95, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   ( 0.7443,  0.8994 )
## Calculations and Intervals on Original Scale
```

```
plastic_summary_data <- filter(summary_data, condition=="PLASTIC")
aggregate_frac_mut_steps_that_change_unexpressed_phenotype <- sum(plastic_summary_data$
sum(plastic_summary_data$dominant_lineage_num_mut_steps_that_change_unexpressed_phenoty
```

```
## [1] 83
```

```
sum(plastic_summary_data$dominant_lineage_num_mut_steps_that_change_aggregate_phenotype
```

```
## [1] 102
```

```
aggregate_frac_mut_steps_that_change_unexpressed_phenotype
```

```
## [1] 0.8137255
```

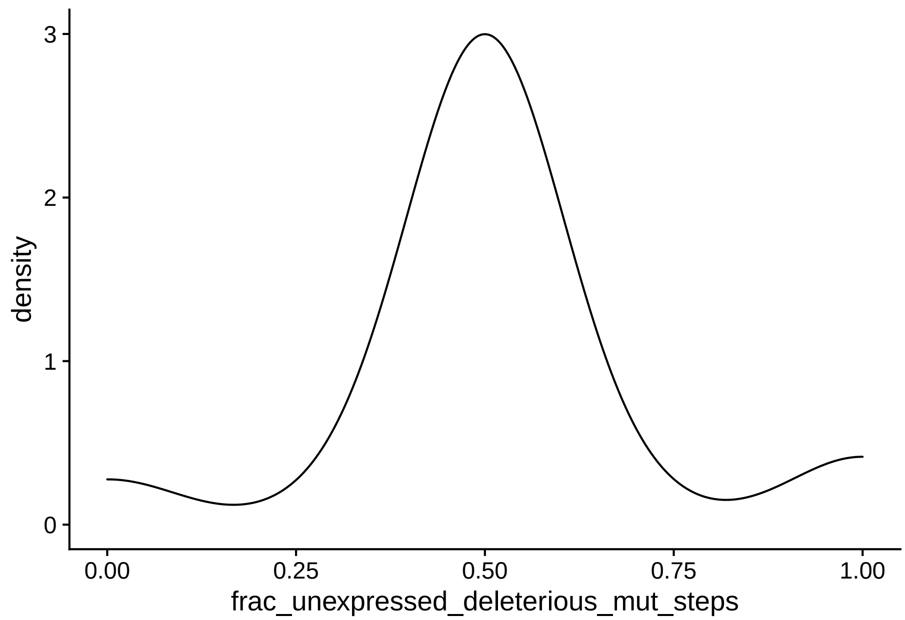
83 / 102 (0.8137255)

3.11.3 For PLASTIC populations, what fraction of mutations that affect the unexpressed phenotype are deleterious versus beneficial?

```
aggregate_frac_unexpressed_deleterious_mut_steps <- sum(plastic_summary_data$dominant_lineage_num
aggregate_frac_unexpressed_beneficial_mut_steps <- sum(plastic_summary_data$dominant_lineage_num_
```

3.11.3.1 Deleterious mutations

```
summary_data$frac_unexpressed_deleterious_mut_steps <- summary_data$dominant_lineage_num_mut_step
ggplot(
  filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change_unexpres
  aes(x=frac_unexpressed_deleterious_mut_steps)
  ) +
  geom_density() +
  theme(
    legend.position="none"
  )
```




```
bo <- boot(filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change
print(bo)
```

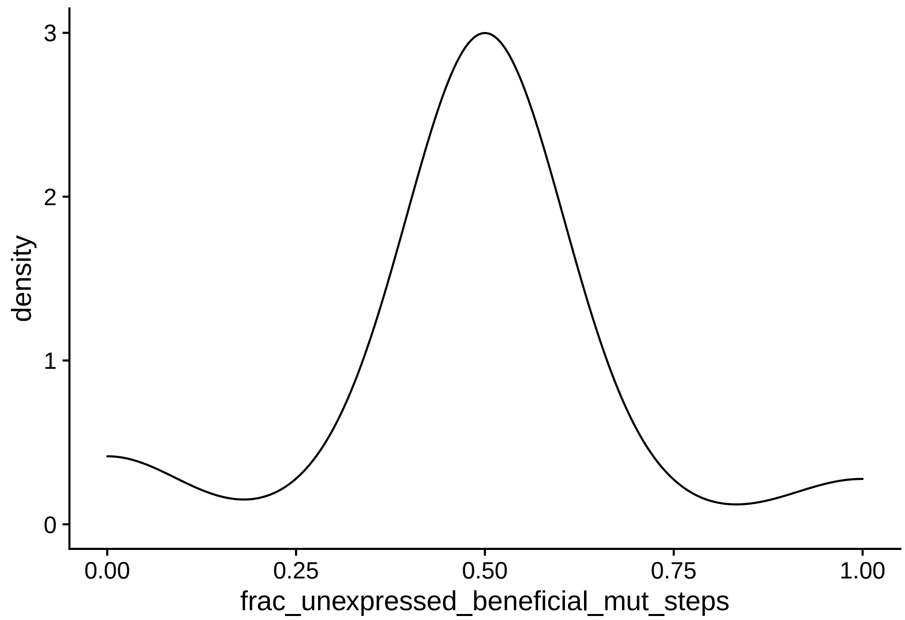
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
```

```
##
## Call:
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_s
##     0)$frac_unexpressed_deleterious_mut_steps, statistic = samplemean,
##     R = 10000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.5172414 -0.0003644828  0.03961947
```

```
print(boot.ci(bo, conf=0.95, type="perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bo, conf = 0.95, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   ( 0.4402,  0.5954 )
## Calculations and Intervals on Original Scale
```

```
summary_data$frac_unexpressed_beneficial_mut_steps <- summary_data$dominant_lineage_nu

ggplot(
  filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_chan
  aes(x=frac_unexpressed_beneficial_mut_steps)
  ) +
  geom_density() +
  theme(
    legend.position="none"
  )
```




```r
bo <- boot(filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change
print(bo)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_steps_that_
##     0)$frac_unexpressed_beneficial_mut_steps, statistic = samplemean,
##     R = 10000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.4827586 -8.16092e-05  0.03986791
```

```r
print(boot.ci(bo, conf=0.95, type="perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bo, conf = 0.95, type = "perc")
##
## Intervals :
```

```
## Level     Percentile
## 95%   ( 0.4046,  0.5598 )
## Calculations and Intervals on Original Scale
```

3.12 Manuscript figures

Figures styled for the paper.

```
grid <- plot_grid(
  coalescence_events_fig +
    theme(
      legend.position="none",
      # axis.ticks.x=element_blank(),
      # axis.text.x=element_blank(),
      axis.title.x=element_blank()
    ) +
    ggtitle("Coalescence events count"),
  mutation_count_fig +
    theme(
      legend.position="none",
      # axis.ticks.x=element_blank(),
      # axis.text.x=element_blank(),
      axis.title.x=element_blank()
    ) +
    ggtitle("Mutation count"),
  phenotypic_volatility_fig +
    theme(
      legend.position="none",
      # axis.ticks.x=element_blank(),
      # axis.text.x=element_blank(),
      axis.title.x=element_blank()
    ) +
    ggtitle("Phenotypic volatility"),
  coalescence_events_freq_fig +
    theme(
      legend.position="none",
      axis.title.x=element_blank()
    ) +
    ggtitle("Generations between coalescence events"),
  genotypic_fidelity_fig +
    theme(
      legend.position="none",
      axis.title.x=element_blank()
    ) +
    ggtitle("Genotypic fidelity"),
```

```
  phenotypic_fidelity_fig +
    theme(
      legend.position="none",
      axis.title.x=element_blank()
    ) +
    ggtitle("Phenotypic fidelity"),
  nrow=2,
  ncol=3,
  align="v",
  labels="auto"
)
save_plot(
  paste0(working_directory, "plots/", "evolutionary-change-full-panel.pdf"),
  grid,
  #  base_height=12,
  #  base_asp=3/2
  base_height=12,
  base_asp=3/2
)
```

Chapter 4

Evolution and maintenance of novel traits

The effect of adaptive phenotypic plasticity on the evolution and maintenance of novel tasks.

4.1 Overview

```
total_updates <- 200000
replicates <- 100
alpha <- 0.05

focal_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")
extra_traits <- c(
  "nor","xor","equals",
  "logic_3aa","logic_3ab","logic_3ac",
  "logic_3ad","logic_3ae","logic_3af",
  "logic_3ag","logic_3ah","logic_3ai",
  "logic_3aj","logic_3ak","logic_3al",
  "logic_3am","logic_3an","logic_3ao",
  "logic_3ap","logic_3aq","logic_3ar",
  "logic_3as","logic_3at","logic_3au",
  "logic_3av","logic_3aw","logic_3ax",
  "logic_3ay","logic_3az","logic_3ba",
  "logic_3bb","logic_3bc","logic_3bd",
  "logic_3be","logic_3bf","logic_3bg",
  "logic_3bh","logic_3bi","logic_3bj",
```

```
  "logic_3bk","logic_3bl","logic_3bm",
  "logic_3bn","logic_3bo","logic_3bp",
  "logic_3bq","logic_3br","logic_3bs",
  "logic_3bt","logic_3bu","logic_3bv",
  "logic_3bw","logic_3bx","logic_3by",
  "logic_3bz","logic_3ca","logic_3cb",
  "logic_3cc","logic_3cd","logic_3ce",
  "logic_3cf","logic_3cg","logic_3ch",
  "logic_3ci","logic_3cj","logic_3ck",
  "logic_3cl","logic_3cm","logic_3cn",
  "logic_3co","logic_3cp"
)

# Relative location of data.
working_directory <- "experiments/2021-01-31-complex-features/analysis/" # << For book
# working_directory <- "./"
```

4.2 Analysis dependencies

Load all required R libraries.

```
library(ggplot2)
library(rstatix)
library(ggsignif)
library(scales)
library(tidyverse)
library(cowplot)
library(RColorBrewer)
library(Hmisc)
library(boot)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd9
```

These analyses were conducted/knitted with the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
```

```
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

4.3 Setup

```r
####### summary data #######
summary_data_loc <- paste0(working_directory, "data/aggregate.csv")
summary_data <- read.csv(summary_data_loc, na.strings="NONE")

summary_data$DISABLE_REACTION_SENSORS <- as.factor(summary_data$DISABLE_REACTION_SENSORS)
summary_data$chg_env <- summary_data$chg_env == "True"
summary_data$dominant_plastic_odd_even <- as.factor(summary_data$dominant_plastic_odd_even)
summary_data$sensors <- summary_data$DISABLE_REACTION_SENSORS == "0"
summary_data$is_plastic <- summary_data$dominant_plastic_odd_even == "True"
summary_data$extra_task_value <- as.factor(summary_data$extra_task_value)
summary_data <- filter(summary_data, extra_task_value == 0.1)

env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
    return("Sensors")
  } else {
    return("No sensors")
  }
}

condition_label_fun <- function(has_sensors, env_chg) {
  if (has_sensors && env_chg) {
    return("PLASTIC")
  } else if (env_chg) {
    return("NON-PLASTIC")
  } else {
    return("STATIC")
```

```
  }
}

summary_data$env_label <- mapply(
  env_label_fun,
  summary_data$chg_env
)
summary_data$sensors_label <- mapply(
  sensors_label_fun,
  summary_data$sensors
)
summary_data$condition <- mapply(
  condition_label_fun,
  summary_data$sensors,
  summary_data$chg_env
)

condition_order = c(
  "STATIC",
  "NON-PLASTIC",
  "PLASTIC"
)
pairwise_comparisons <- list(
  c("STATIC", "NON-PLASTIC"),
  c("STATIC", "PLASTIC"),
  c("PLASTIC", "NON-PLASTIC")
)

p_label <- function(p_value) {
  threshold = 0.0001
  if (p_value < threshold) {
    return(paste0("p < ", threshold))
  } else {
    return(paste0("p = ", p_value))
  }
}

###### time series #####
lineage_time_series_data_loc <- paste0(working_directory, "data/lineage_series.csv")
lineage_time_series_data <- read.csv(lineage_time_series_data_loc)

lineage_time_series_data$DISABLE_REACTION_SENSORS <- as.factor(lineage_time_series_data
lineage_time_series_data$chg_env <- lineage_time_series_data$chg_env == "True"
lineage_time_series_data$sensors <- lineage_time_series_data$DISABLE_REACTION_SENSORS =
lineage_time_series_data$extra_task_value <- as.factor(lineage_time_series_data$extra_t
```

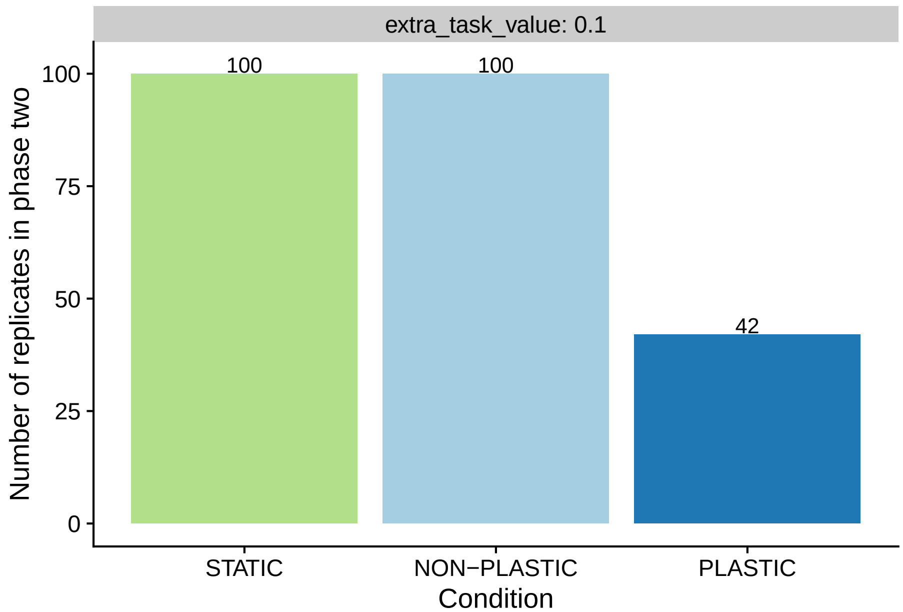
```
lineage_time_series_data$env_label <- mapply(
  env_label_fun,
  lineage_time_series_data$chg_env
)
lineage_time_series_data$sensors_label <- mapply(
  sensors_label_fun,
  lineage_time_series_data$sensors
)
lineage_time_series_data$condition <- mapply(
  condition_label_fun,
  lineage_time_series_data$sensors,
  lineage_time_series_data$chg_env
)

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
# Palette
cb_palette <- "Paired"
# Create directory to dump plots
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
# Sample mean function
samplemean <- function(x, d) {
  return(mean(x[d]))
}
```

4.4 The evolution of phenotypic plasticity

For sensor-enabled populations in fluctuating environments, we only transfered populations containing an optimally plastic genotype to phase two.

```
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition, extra_task_va
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())

ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
```

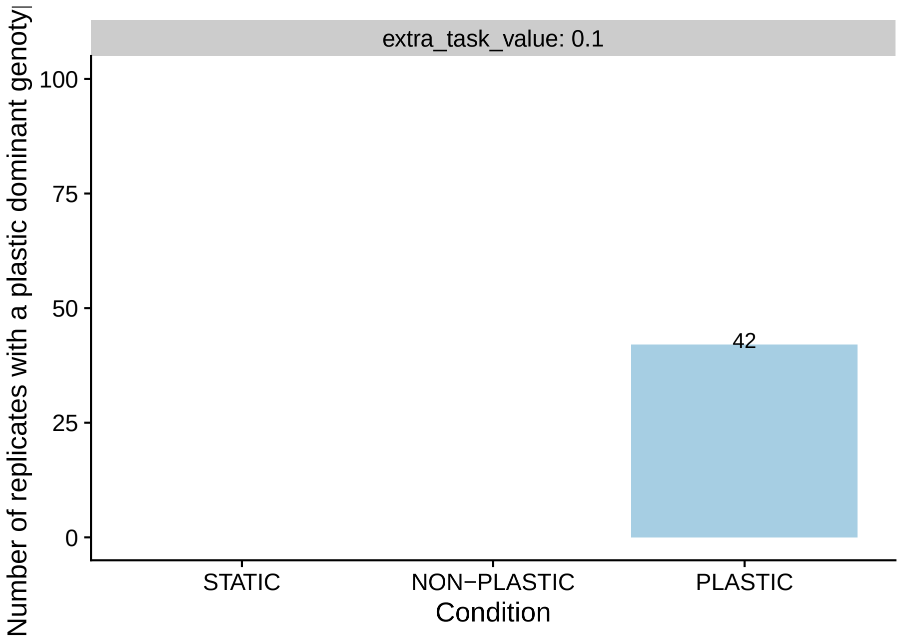
```
  scale_color_brewer(
    palette=cb_palette
  ) +
  ylab("Number of replicates in phase two") +
  facet_wrap(~extra_task_value, labeller=label_both) +
  theme(
    legend.position="none"
  )
```



We can confirm our expectation that the dominant genotypes in non-plastic conditions are not phenotypically plastic.

```
summary_data_grouped = dplyr::group_by(summary_data, condition, is_plastic, extra_task_
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
ggplot(filter(summary_data_group_counts, is_plastic), aes(x=condition, y=n, fill=condi
  geom_col(position=position_dodge(0.9)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
```

```
  ) +
  ylim(0, 100) +
  geom_text(aes(label=n, y=n+1)) +
  ylab("Number of replicates with a plastic dominant genotype") +
  facet_wrap(~extra_task_value, labeller=label_both) +
  theme(
    legend.position="none"
  )
```



4.5 Final novel task count (dominant genotype)

How many novel tasks do final dominant genotypes perform?

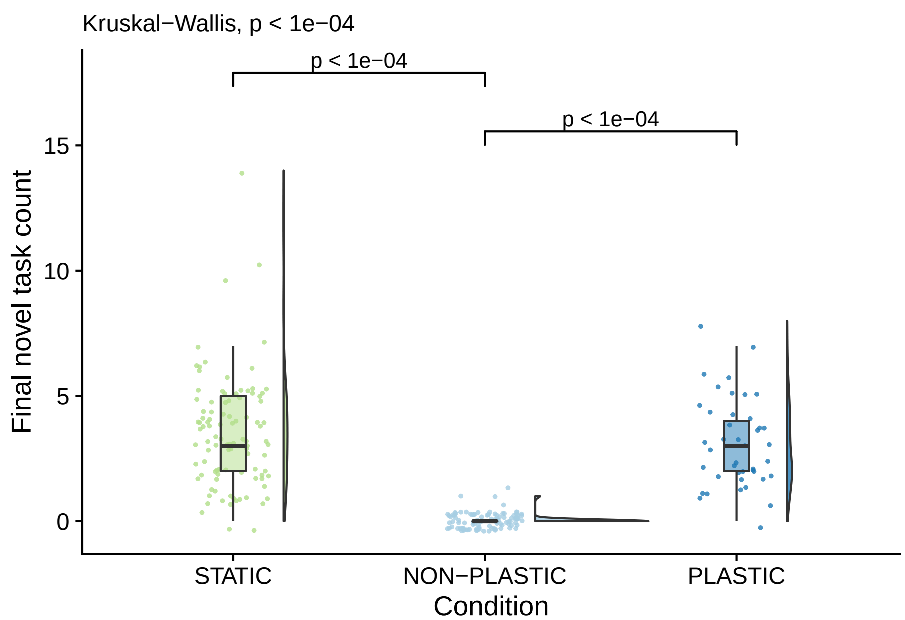
```
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(dominant_extra_tasks ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition") # ,step.increase=1
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <- stat.test$y.position #* c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

final_novel_task_count_fig <- ggplot(
```

```
    summary_data,
    aes(x=condition, y=dominant_extra_tasks, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Final novel task count"
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_extra_tasks~condition, data=summary_
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip()
```

```
  theme(
    legend.position="none"
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
final_novel_task_count_fig
```




```
kruskal.test(
  formula=dominant_extra_tasks~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_extra_tasks by condition
## Kruskal-Wallis chi-squared = 177.17, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_extra_tasks,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_extra_tasks and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.9
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_extra_tasks)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_extra_tasks)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_extra_tasks)
  )
)
```

```
## [1] "PLASTIC median: 3; STATIC median: 3; NON-PLASTIC median: 0"
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

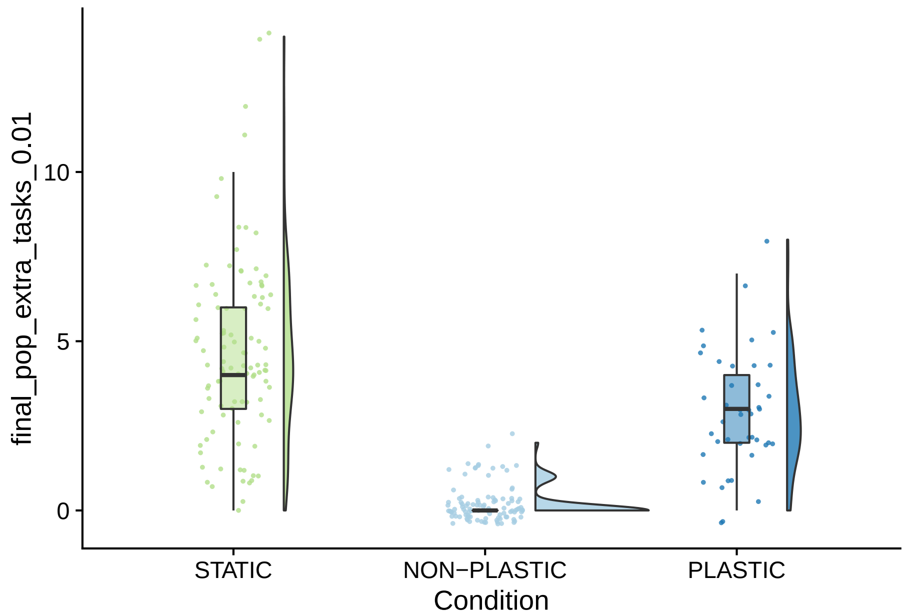
```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_extra_tasks~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=184"
## [1] "STATIC<-->PLASTIC: W=1871"
## [1] "PLASTIC<-->NON-PLASTIC: W=64"
```

4.6 Novel task count (final population)

How many novel tasks are performed across the final population (1% of organisms must perform to count)?

```
ggplot(summary_data, aes(x=condition, y=final_pop_extra_tasks_0.01, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  )
```




```
kruskal.test(
  formula=final_pop_extra_tasks_0.01~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  final_pop_extra_tasks_0.01 by condition
## Kruskal-Wallis chi-squared = 169.47, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$final_pop_extra_tasks_0.01,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$final_pop_extra_tasks_0.01 and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
```

```
## STATIC  < 2e-16     0.00016
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$final_pop_extra_tasks_0.01)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$final_pop_extra_tasks_0.01)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$final_pop_extra_tasks_0.01)
  )
)
```

```
## [1] "PLASTIC median: 3; STATIC median: 4; NON-PLASTIC median: 0"
```

```r
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```r
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=final_pop_extra_tasks_0.01~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=227.5"
## [1] "STATIC<-->PLASTIC: W=1203"
## [1] "PLASTIC<-->NON-PLASTIC: W=225.5"
```

4.7 Novel task discovery (lineage)

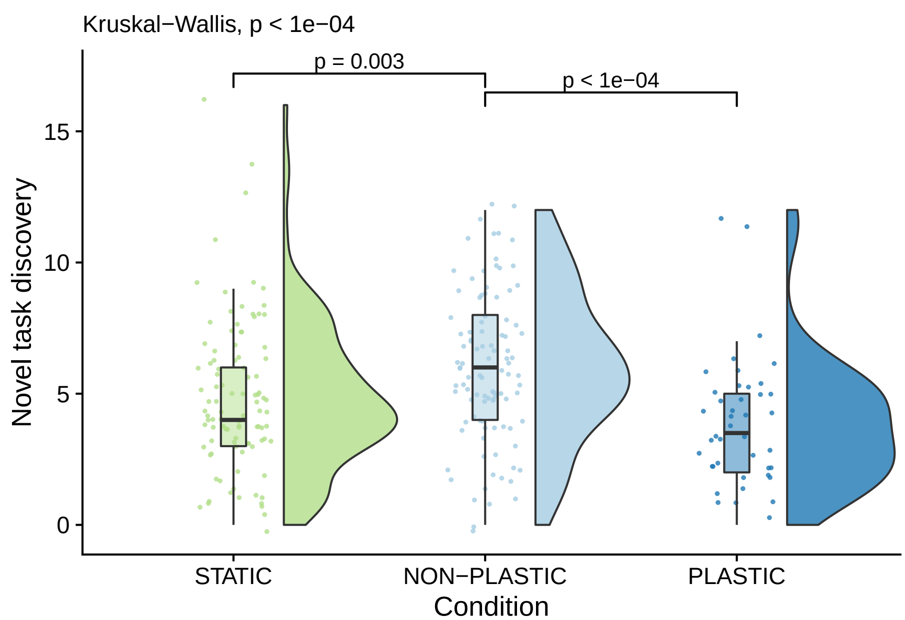
```r
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_extra_traits_discovered ~ condition) %>%
```

```
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition") # ,step.increase=1
# Tweak y.position manually to account for scaled axis (edge case that triggers bad be
stat.test$manual_position <- stat.test$y.position #* c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

lineage_novel_task_discovery_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_extra_traits_discovered, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Novel task discovery"
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_extra_traits_discovered~conc
    )
```

```
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip()
  theme(
    legend.position="none"
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
lineage_novel_task_discovery_fig
```




```
kruskal.test(
  formula=dominant_lineage_extra_traits_discovered~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_extra_traits_discovered by condition
## Kruskal-Wallis chi-squared = 24.099, df = 2, p-value = 5.846e-06
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_extra_traits_discovered,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_extra_traits_discovered and summary_data$condi
##
##         NON-PLASTIC PLASTIC
## PLASTIC 1.7e-05     -
## STATIC  0.0035      0.0561
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_di
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_extra_traits_dis
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_trait
  )
)
```

```
## [1] "PLASTIC median: 3.5; STATIC median: 4; NON-PLASTIC median: 6"
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_extra_traits_discovered~condition,
    data=pair_data,
```

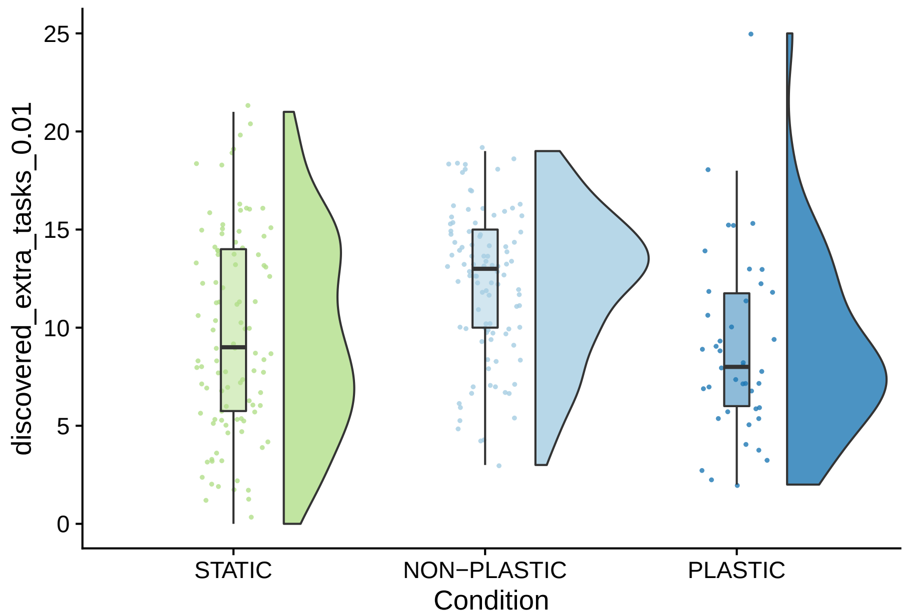
```
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=6319.5"
## [1] "STATIC<-->PLASTIC: W=1578"
## [1] "PLASTIC<-->NON-PLASTIC: W=3110.5"
```

4.8 Novel task discovery (population)

```
ggplot(
    summary_data,
    aes(x=condition, y=discovered_extra_tasks_0.01, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
```

```
)
```




```
kruskal.test(
  formula=discovered_extra_tasks_0.01~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  discovered_extra_tasks_0.01 by condition
## Kruskal-Wallis chi-squared = 24.271, df = 2, p-value = 5.365e-06
```

```
pairwise.wilcox.test(
  x=summary_data$discovered_extra_tasks_0.01,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$discovered_extra_tasks_0.01 and summary_data$condition
##
```

```
##         NON-PLASTIC PLASTIC
## PLASTIC 2.4e-05     -
## STATIC  0.00035     1.00000
##
## P value adjustment method: bonferroni
```

```r
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$discovered_extra_tasks_0.01)
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$discovered_extra_tasks_0.01)
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$discovered_extra_tasks_0.01)
  )
)
```

```
## [1] "PLASTIC median: 8; STATIC median: 9; NON-PLASTIC median: 13"
```

```r
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```r
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=discovered_extra_tasks_0.01~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=6573.5"
## [1] "STATIC<-->PLASTIC: W=1918.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=3096"
```

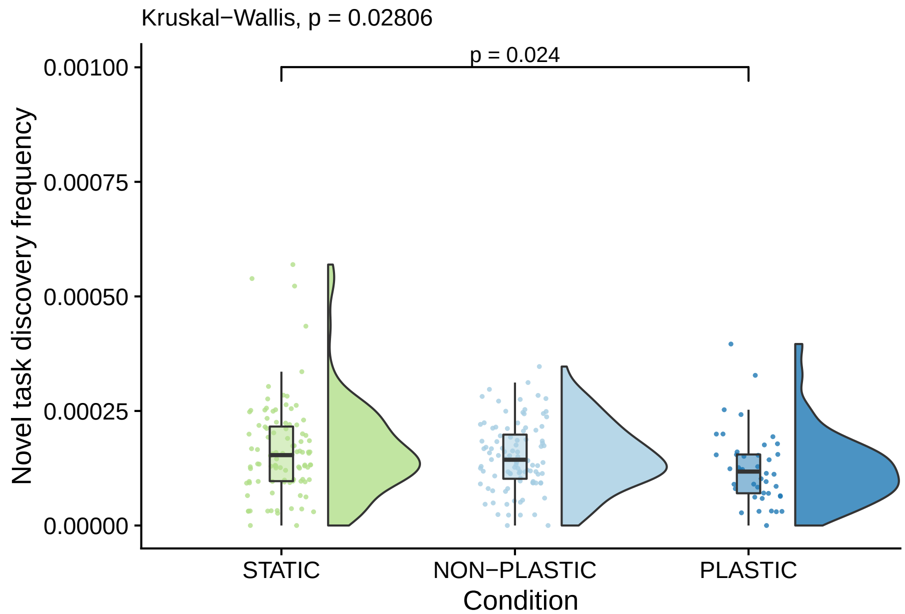
4.9 Novel task discovery frequency (lineage)

```
summary_data$dominant_lineage_extra_traits_discovered_per_generation <- summary_data$do
summary_data$dominant_lineage_extra_traits_generations_per_discovery <- summary_data$do

# Compute manual labels for geom_signif
# stat.test <- filter(summary_data, dominant_lineage_extra_traits_discovered > 0) %>%
#   wilcox_test(dominant_lineage_extra_traits_generations_per_discovery ~ condition) %>
#   adjust_pvalue(method = "bonferroni") %>%
#   add_significance() %>%
#   add_xy_position(x="condition") # ,step.increase=1
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_extra_traits_discovered_per_generation ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition", step.increase=0.0001) # ,step.increase=1
# Tweak y.position manually to account for scaled axis (edge case that triggers bad beh
stat.test$manual_position <- stat.test$y.position #* c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

lineage_novel_task_discovery_freq_fig <- ggplot(
    # filter(summary_data, dominant_lineage_extra_traits_discovered > 0),
    summary_data,
    aes(x=condition, y=dominant_lineage_extra_traits_discovered_per_generation, fill=co
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("Novel task discovery frequency") +
```

```
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_extra_traits_discovered_per_generation
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
lineage_novel_task_discovery_freq_fig
```




```r
kruskal.test(
  formula=dominant_lineage_extra_traits_discovered_per_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_extra_traits_discovered_per_generation by condition
## Kruskal-Wallis chi-squared = 7.1465, df = 2, p-value = 0.02806
```

```r
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_extra_traits_discovered_per_generation,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_extra_traits_discovered_per_generation and summ
##
##         NON-PLASTIC PLASTIC
## PLASTIC 0.092       -
```

```
## STATIC  1.000      0.025
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_discovered_pe
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_extra_traits_discovered_per
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_traits_discovere
  )
)
```

```
## [1] "PLASTIC median: 0.000117695011124939; STATIC median: 0.00015363220504867; NON-PLASTIC med
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

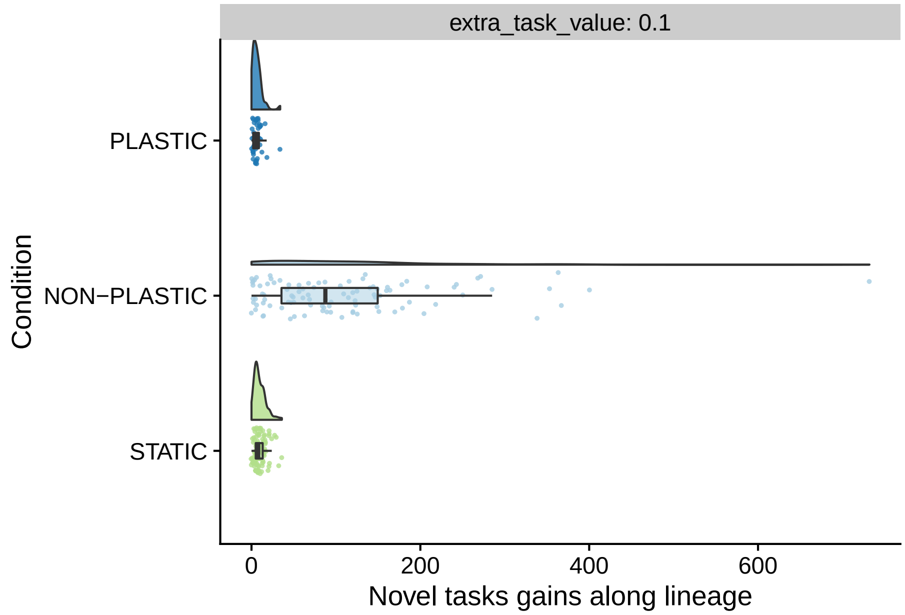
```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_extra_traits_discovered_per_generation~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=4751"
## [1] "STATIC<-->PLASTIC: W=1510.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=2584"
```

4.10 Novel tasks gained (lineage)

```
ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_extra_traits_gained, fill=condition)
```

```
) +
geom_flat_violin(
  position = position_nudge(x = .2, y = 0),
  alpha = .8
) +
geom_point(
  mapping=aes(color=condition),
  position = position_jitter(width = .15),
  size = .5,
  alpha = 0.8
) +
geom_boxplot(
  width = .1,
  outlier.shape = NA,
  alpha = 0.5
) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
ylab("Novel tasks gains along lineage") +
scale_fill_brewer(
  palette=cb_palette
) +
scale_color_brewer(
  palette=cb_palette
) +
coord_flip() +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
) +
ggsave(
  paste0(working_directory, "plots/dominant-lineage-extra-tasks-gained.pdf"),
  width=15,
  height=10
)
```



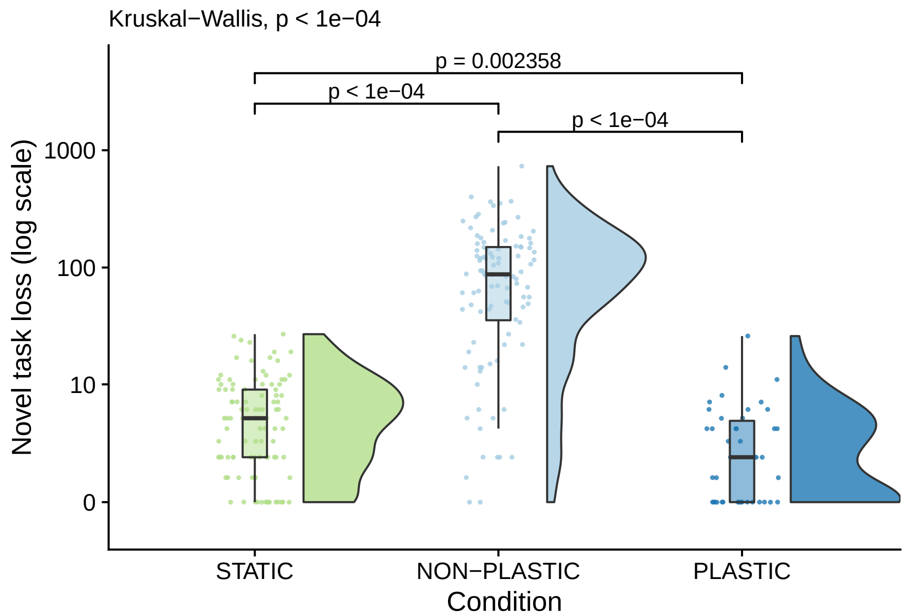
4.11 Novel task loss (lineage)

```
# Compute manual labels for geom_signif
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_extra_traits_lost ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition", step.increase=1)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <-  log10(stat.test$y.position) * c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

lineage_novel_task_loss_fig <- ggplot(
    summary_data,
    aes(x=condition, y=dominant_lineage_extra_traits_lost, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
```

```
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Novel task loss (log scale)",
    trans=pseudo_log_trans(sigma=1, base=10),
    breaks=c(0,10,100,1000),
    limits=c(-1,5000)
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_extra_traits_lost~condition
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj<=alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  # coord_flip()
  theme(
    legend.position="none"
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
lineage_novel_task_loss_fig
```




```
kruskal.test(
  formula=dominant_lineage_extra_traits_lost~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_extra_traits_lost by condition
## Kruskal-Wallis chi-squared = 129.06, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_extra_traits_lost,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_extra_traits_lost and summary_data$condition
##
```

```
##         NON-PLASTIC PLASTIC
## PLASTIC 2.7e-16     -
## STATIC  < 2e-16     0.0024
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_los
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_extra_traits_lost
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_traits
  )
)
```

```
## [1] "PLASTIC median: 2; STATIC median: 5; NON-PLASTIC median: 87.5"
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_extra_traits_lost~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=9105"
## [1] "STATIC<-->PLASTIC: W=1353.5"
## [1] "PLASTIC<-->NON-PLASTIC: W=3959"
```

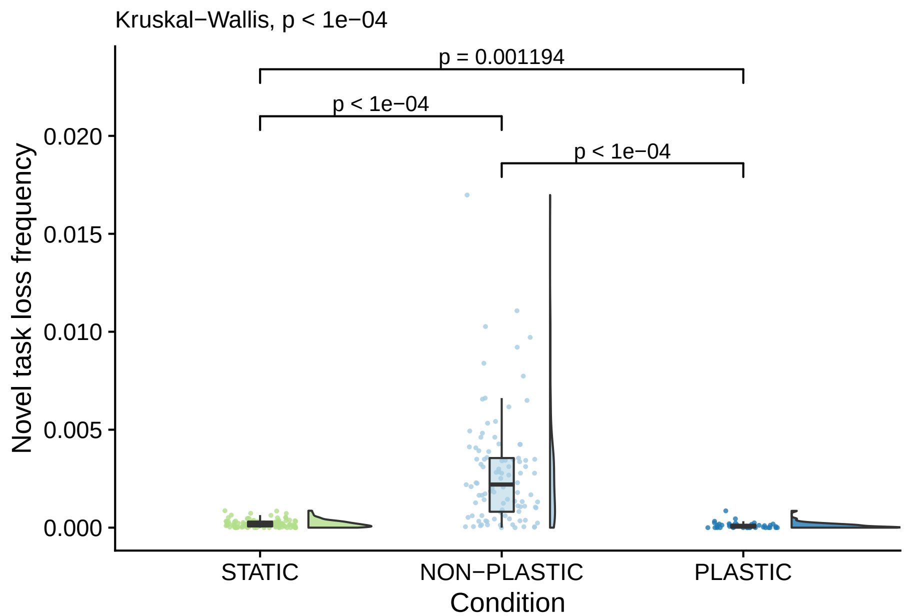
4.12 Frequency of novel task loss (lineage)

```
summary_data$dominant_lineage_extra_traits_lost_per_generation <- summary_data$dominant_lineage_e
summary_data$dominant_lineage_extra_traits_generations_per_loss <- summary_data$dominant_generati

# Compute manual labels for geom_signif
# stat.test <- filter(summary_data, dominant_lineage_extra_traits_lost > 0) %>%
#   wilcox_test(dominant_lineage_extra_traits_generations_per_loss ~ condition) %>%
#   adjust_pvalue(method = "bonferroni") %>%
#   add_significance() %>%
#   add_xy_position(x="condition", step.increase=1)
stat.test <- summary_data %>%
  wilcox_test(dominant_lineage_extra_traits_lost_per_generation ~ condition) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="condition", step.increase=.1)
# Tweak y.position manually to account for scaled axis (edge case that triggers bad behavior in g
stat.test$manual_position <-  stat.test$y.position #* c(1.0,1.0,1.03)
stat.test$label <- mapply(p_label,stat.test$p.adj)

lineage_novel_task_loss_freq_fig <- ggplot(
    # filter(summary_data, dominant_lineage_extra_traits_lost > 0),
    summary_data,
    aes(x=condition, y=dominant_lineage_extra_traits_lost_per_generation, fill=condition)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("Novel task loss frequency") +
```

```
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(signif(kruskal.test(formula=dominant_lineage_extra_traits_lost_per_genera
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj<=alpha),
    aes(xmin=group1,xmax=group2,annotations=label,y_position=manual_position),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  )
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax, annotations, y_position
```

```
lineage_novel_task_loss_freq_fig
```



```
kruskal.test(
  formula=dominant_lineage_extra_traits_lost_per_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_extra_traits_lost_per_generation by condition
## Kruskal-Wallis chi-squared = 121.41, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_extra_traits_lost_per_generation,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_extra_traits_lost_per_generation and summary_data$conditi
##
##         NON-PLASTIC PLASTIC
## PLASTIC 1.1e-15     -
## STATIC  < 2e-16     0.0012
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_lost_per_gene
  ),
  paste0(
    "STATIC median: ",
    median(filter(summary_data, condition=="STATIC")$dominant_lineage_extra_traits_lost_per_gener
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_traits_lost_per_
  )
)
```

```
## [1] "PLASTIC median: 6.25141973661864e-05; STATIC median: 0.000161396283669756; NON-PLASTIC me
```

```r
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```r
for (pair in pairwise_comparisons) {
  pair_data <- filter(summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=dominant_lineage_extra_traits_lost_per_generation~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=8940"
## [1] "STATIC<-->PLASTIC: W=1311"
## [1] "PLASTIC<-->NON-PLASTIC: W=3922"
```

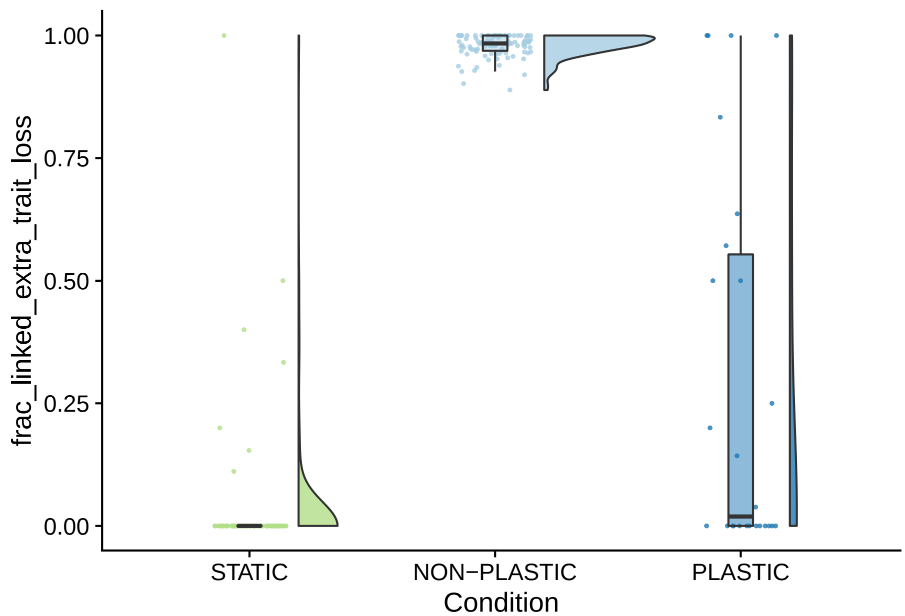
4.13 How many instances of novel trait loss co-occurred with changes in base phenotype?

Task loss linked with primary trait changes.

```r
lost_traits_summary_data <- filter(summary_data, extra_task_value==0.1 & dominant_linea
lost_traits_summary_data$frac_linked_extra_trait_loss <- lost_traits_summary_data$domin
```

```r
ggplot(lost_traits_summary_data, aes(x=condition, y=frac_linked_extra_trait_loss, fill=
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
```

```
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette=cb_palette
  ) +
  scale_color_brewer(
    palette=cb_palette
  ) +
  # coord_flip() +
  theme(
    legend.position="none"
  )
```




```
kruskal.test(
  formula=frac_linked_extra_trait_loss~condition,
  data=lost_traits_summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_linked_extra_trait_loss by condition
## Kruskal-Wallis chi-squared = 153.68, df = 2, p-value < 2.2e-16
```

```
pairwise.wilcox.test(
  x=lost_traits_summary_data$frac_linked_extra_trait_loss,
```

```
  g=lost_traits_summary_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  lost_traits_summary_data$frac_linked_extra_trait_loss and lost_traits_summary
##
##         NON-PLASTIC PLASTIC
## PLASTIC 1.9e-08     -
## STATIC  < 2e-16     1.8e-06
##
## P value adjustment method: bonferroni
```

```
paste(
  sep="; ",
  paste0(
    "PLASTIC median: ",
    median(filter(lost_traits_summary_data, condition=="PLASTIC")$frac_linked_extra_tra
  ),
  paste0(
    "STATIC median: ",
    median(filter(lost_traits_summary_data, condition=="STATIC")$frac_linked_extra_trai
  ),
  paste0(
    "NON-PLASTIC median: ",
    median(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$frac_linked_extra
  )
)
```

```
## [1] "PLASTIC median: 0.0192307692307692; STATIC median: 0; NON-PLASTIC median: 0.98
```

```
print("Wilcox rank sum test statistics:")
```

```
## [1] "Wilcox rank sum test statistics:"
```

```
for (pair in pairwise_comparisons) {
  pair_data <- filter(lost_traits_summary_data, condition %in% pair)
  pair_data$condition <- as.factor(pair_data$condition)
  wt <- wilcox.test(
    formula=frac_linked_extra_trait_loss~condition,
    data=pair_data,
    exact=FALSE,
    paired=FALSE
```

```
  )
  print(paste0(pair[1], "<-->", pair[2], ": W=",wt$statistic))
}
```

```
## [1] "STATIC<-->NON-PLASTIC: W=8344"
## [1] "STATIC<-->PLASTIC: W=1602"
## [1] "PLASTIC<-->NON-PLASTIC: W=2212"
```

```
sum(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_traits_lost
```

```
## [1] 10998
```

```
sum(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_traits_lost
```

```
## [1] 11229
```

```
aggregate_frac_linked_extra_trait_loss_nonplastic <- sum(filter(lost_traits_summary_data, conditi
aggregate_frac_linked_extra_trait_loss_nonplastic
```

```
## [1] 0.9794283
```

```
sum(filter(lost_traits_summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_lost_lin
```

```
## [1] 29
```

```
sum(filter(lost_traits_summary_data, condition=="PLASTIC")$dominant_lineage_extra_traits_lost)
```

```
## [1] 142
```

```
aggregate_frac_linked_extra_trait_loss_plastic <- sum(filter(lost_traits_summary_data, condition=
aggregate_frac_linked_extra_trait_loss_plastic
```

```
## [1] 0.2042254
```

```
sum(filter(lost_traits_summary_data, condition=="STATIC")$dominant_lineage_extra_traits_lost_link
```

```
## [1] 13
```

```
sum(filter(lost_traits_summary_data, condition=="STATIC")$dominant_lineage_extra_traits_lost)
```

```
## [1] 631
```

```
aggregate_frac_linked_extra_trait_loss_nonplastic <- sum(filter(lost_traits_summary_data, conditi
aggregate_frac_linked_extra_trait_loss_nonplastic
```

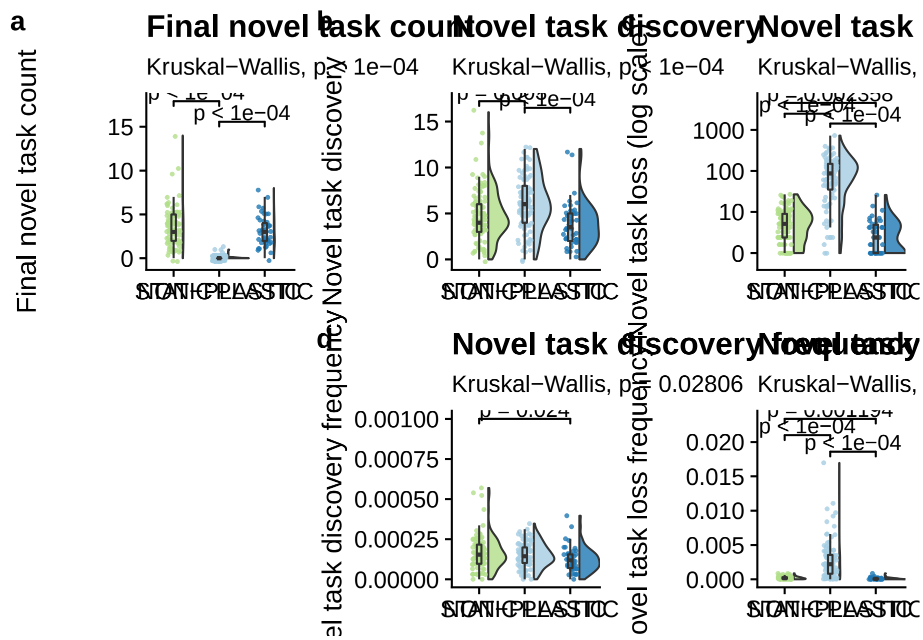
```
## [1] 0.02060222
```

4.14 Manuscript figures

4.15 Combined panel

```r
grid <- plot_grid(
  final_novel_task_count_fig +
    theme(
      axis.title.x=element_blank()
    ) +
    ggtitle("Final novel task count"),
  lineage_novel_task_discovery_fig +
    theme(
      axis.title.x=element_blank(),
      # axis.text.x=element_blank(),
      # axis.ticks.x=element_blank()
    ) +
    ggtitle("Novel task discovery"),
  lineage_novel_task_loss_fig +
    theme(
      axis.title.x=element_blank(),
      # axis.text.x=element_blank(),
      # axis.ticks.x=element_blank()
    ) +
    ggtitle("Novel task loss"),
  NULL,
  lineage_novel_task_discovery_freq_fig +
    theme(
      axis.title.x=element_blank()
    ) +
    ggtitle("Novel task discovery frequency"),
  lineage_novel_task_loss_freq_fig +
    theme(
      axis.title.x=element_blank()
    ) +
    ggtitle("Novel task loss frequency"),
  nrow=2,
  align="v",
  # labels="auto"
  labels=c("a", "b","c","", "d","e")
)
save_plot(
   paste0(working_directory, "plots/", "complex-traits-panel.pdf"),
   grid,
   base_height=12,
   base_asp=3/2
```

```
)
grid
```


a
Final novel task count
Kruskal−Wallis, p < 1e−04
Final novel task count
b
Novel task discovery
Kruskal−Wallis, p < 1e−04
Novel task discovery
c
Novel task
Kruskal−Wallis,
Novel task loss (log scale)
d
Novel task discovery frequency
Kruskal−Wallis, p = 0.02806
Novel task discovery frequency
p = 0.024
Novel task
Kruskal−Wallis,
Novel task loss frequency
p < 1e−04
STATIC
PLASTIC

Chapter 5

Accumulation of deleterious instructions

The effect of adaptive phenotypic plasticity on the accumulation of deleterious genes.

5.1 Overview

```
total_updates <- 200000
replicates <- 100

focal_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")

# Relative location of data.
working_directory <- "experiments/2021-02-05-hitchhiking/analysis/" # << For bookdown
# working_directory <- "./"
```

5.2 Analysis dependencies

Load all required R libraries.

```
library(RColorBrewer)
library(ggplot2)
library(tidyverse)
library(cowplot)
library(Hmisc)
```

```r
library(boot)
library(fmsb)
library(knitr)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd9
```

These analyses were conducted/knitted with the following computing environment:

```r
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

5.3 Setup

```r
####### summary data #######
summary_data_loc <- paste0(working_directory, "data/aggregate.csv")
summary_data <- read.csv(summary_data_loc, na.strings="NONE")

summary_data$DISABLE_REACTION_SENSORS <- as.factor(summary_data$DISABLE_REACTION_SENSO
summary_data$chg_env <- summary_data$chg_env == "True"
summary_data$dominant_plastic_odd_even <- as.factor(summary_data$dominant_plastic_odd_
summary_data$sensors <- summary_data$DISABLE_REACTION_SENSORS == "0"
summary_data$is_plastic <- summary_data$dominant_plastic_odd_even == "True"
summary_data$POISON_PENALTY <- as.factor(summary_data$POISON_PENALTY)

env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
```

```
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
    return("Sensors")
  } else {
    return("No sensors")
  }
}

condition_label_fun <- function(has_sensors, env_chg) {
  if (has_sensors && env_chg) {
    return("PLASTIC")
  } else if (env_chg) {
    return("NON-PLASTIC")
  } else {
    return("STATIC")
  }
}

summary_data$env_label <- mapply(
  env_label_fun,
  summary_data$chg_env
)
summary_data$sensors_label <- mapply(
  sensors_label_fun,
  summary_data$sensors
)
summary_data$condition <- mapply(
  condition_label_fun,
  summary_data$sensors,
  summary_data$chg_env
)

condition_order = c(
  "STATIC",
  "NON-PLASTIC",
  "PLASTIC"
)

###### time series #####
lineage_time_series_data_loc <- paste0(working_directory, "data/lineage_series.csv")
lineage_time_series_data <- read.csv(lineage_time_series_data_loc)

lineage_time_series_data$DISABLE_REACTION_SENSORS <- as.factor(lineage_time_series_data$DISABLE_R
```

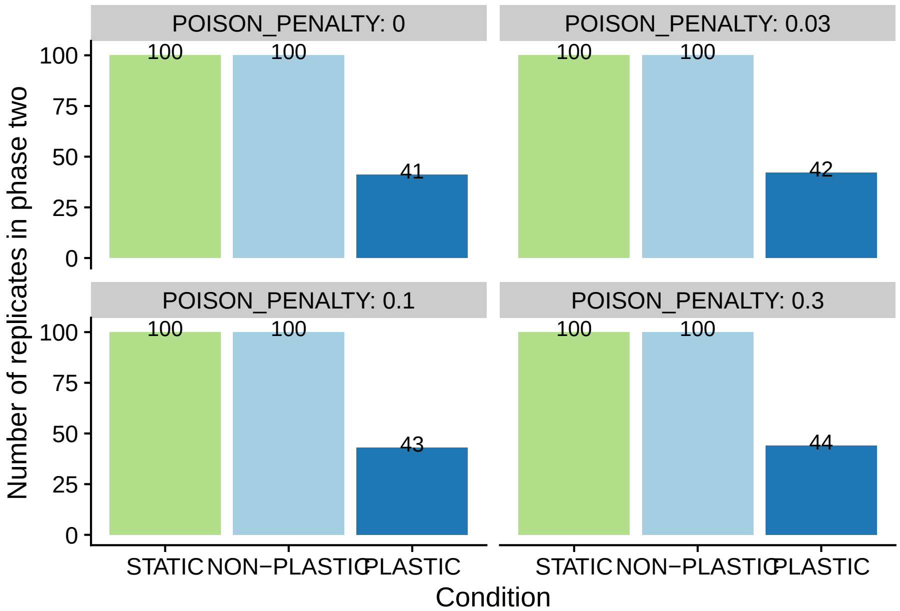
```
lineage_time_series_data$chg_env <- lineage_time_series_data$chg_env == "True"
lineage_time_series_data$sensors <- lineage_time_series_data$DISABLE_REACTION_SENSORS =
lineage_time_series_data$POISON_PENALTY <- as.factor(lineage_time_series_data$POISON_VA

lineage_time_series_data$env_label <- mapply(
  env_label_fun,
  lineage_time_series_data$chg_env
)
lineage_time_series_data$sensors_label <- mapply(
  sensors_label_fun,
  lineage_time_series_data$sensors
)
lineage_time_series_data$condition <- mapply(
  condition_label_fun,
  lineage_time_series_data$sensors,
  lineage_time_series_data$chg_env
)

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
samplemean <- function(x, d) {
  return(mean(x[d]))
}
```

5.4 Evolution of phenotypic plasticity

For sensor-enabled populations in fluctuating environments, we only transfered populations containing an optimally plastic genotype to phase-two.

```
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition, POI
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())

ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
```

```
    palette="Paired"
  ) +
  ylab("Number of replicates in phase two") +
  facet_wrap(~POISON_PENALTY, labeller=label_both) +
  theme(
    legend.position="none"
  )
```



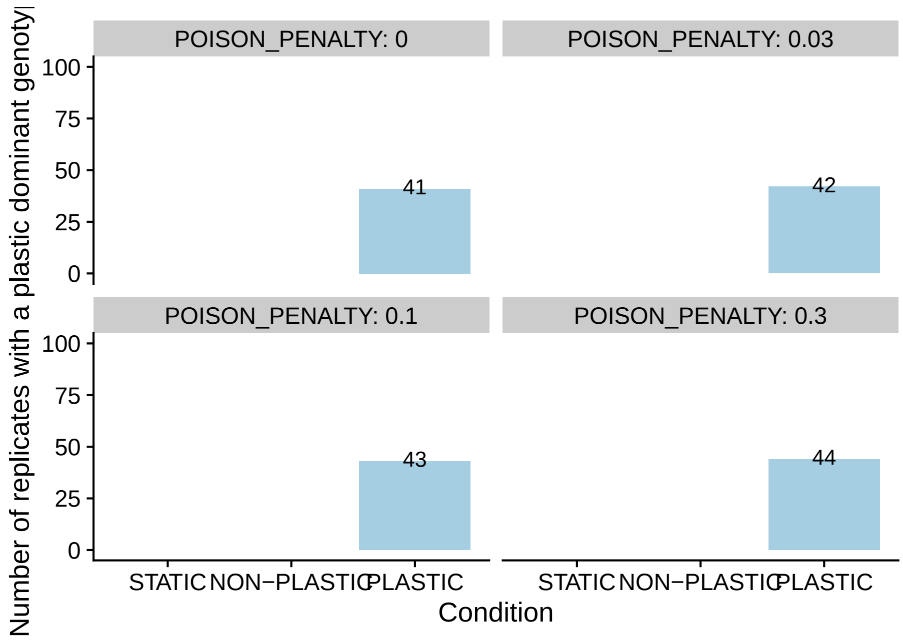
We can confirm our expectation that the dominant genotypes in non-plastic conditions are not phenotypically plastic.

```
summary_data_grouped = dplyr::group_by(summary_data, condition, is_plastic, POISON_PENALTY)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
```

```
## `summarise()` has grouped output by 'condition', 'is_plastic'. You can override using the `.gr
```

```
ggplot(filter(summary_data_group_counts, is_plastic), aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  geom_text(aes(label=n, y=n+1)) +
  scale_fill_brewer(
    palette="Paired"
  ) +
```

```
  scale_color_brewer(
    palette="Paired"
  ) +
  ylab("Number of replicates with a plastic dominant genotype") +
  ylim(0, 100) +
  facet_wrap(~POISON_PENALTY, labeller=label_both) +
  theme(
    legend.position="none"
  )
```



5.5 Poison instruction execution

5.5.1 Number of replicates where final dominant genotype executes the poison instruction

```
poison_penalty <- 0.1
```

```
occurrences <- c(
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="NON-PLASTIC"
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="PLASTIC" & d
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="STATIC" & do
)
trials <- c(
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="NON-PLASTIC"
```

```r
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="PLASTIC")$RANDOM_SEED)
  length(filter(summary_data, POISON_PENALTY==poison_penalty & condition=="STATIC" )$RANDOM_SEED)
)
names(trials) <- c(
  "NON-PLASTIC",
  "PLASTIC",
  "STATIC"
)
names(occurrences) <- c(
  "NON-PLASTIC",
  "PLASTIC",
  "STATIC"
)

poison_exec_table <- data.frame(
  executes.poison=occurrences,
  replicates=trials
)
kable(poison_exec_table)
```

	executes.poison	replicates
NON-PLASTIC	14	100
PLASTIC	0	43
STATIC	0	100

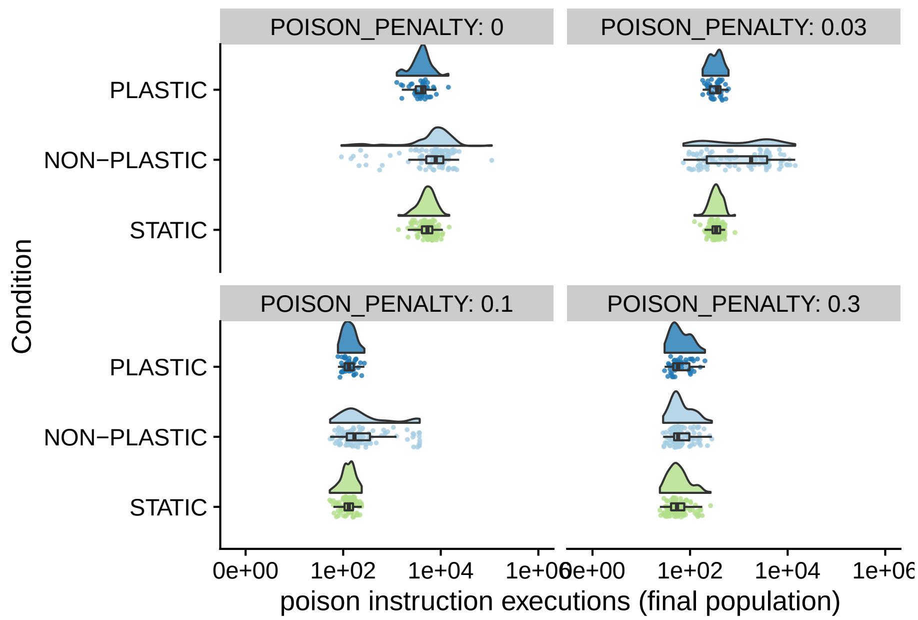
```r
pairwise.fisher.test(x=occurrences, n=trials, p.adjust.method="bonferroni")
```

```
##
##  Pairwise comparisons using Pairwise comparison of proportions (Fisher)
##
## data:  occurrences out of trials
##
##         NON-PLASTIC PLASTIC
## PLASTIC 0.03212     -
## STATIC  0.00022     1.00000
##
## P value adjustment method: bonferroni
```

5.5.2 Poison instruction execution in final population

```r
ggplot(summary_data, aes(x=condition, y=final_population_poison, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
```

```
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_y_continuous(
    name="poison instruction executions (final population)",
    trans="pseudo_log",
    breaks=c(0,100,10000,1000000),
    limits=c(-1,1000000)
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/final-population-poison-log.pdf"),
    width=15,
    height=10
  )
```




```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
      formula=final_population_poison~condition,
      data=stat_data
    )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
  if (kt$p.value > 0.05) { next }
  print(
    pairwise.wilcox.test(
      x=stat_data$final_population_poison,
      g=stat_data$condition,
      p.adjust.method="bonferroni"
    )
  )
}
```

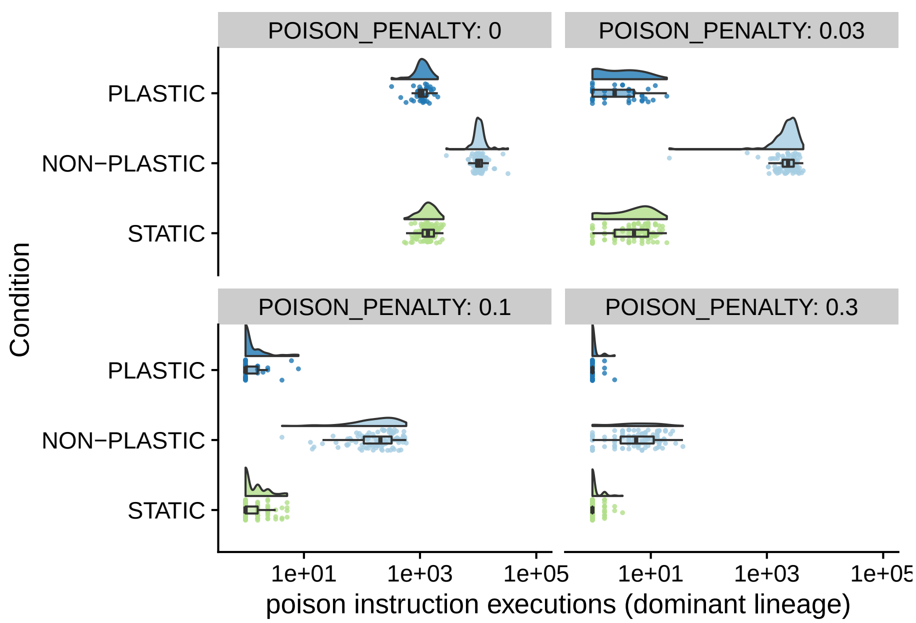
```
## [1] "PENALTY: 0"
##
##  Kruskal-Wallis rank sum test
##
## data:  final_population_poison by condition
## Kruskal-Wallis chi-squared = 43.589, df = 2, p-value = 3.426e-10
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$final_population_poison and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 8.7e-07     -
## STATIC  9.8e-07     0.00074
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
##  Kruskal-Wallis rank sum test
##
## data:  final_population_poison by condition
## Kruskal-Wallis chi-squared = 20.74, df = 2, p-value = 3.136e-05
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$final_population_poison and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 0.003       -
## STATIC  1e-04       1.000
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
##  Kruskal-Wallis rank sum test
##
## data:  final_population_poison by condition
## Kruskal-Wallis chi-squared = 20.608, df = 2, p-value = 3.35e-05
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$final_population_poison and stat_data$condition
##
```

```
##         NON-PLASTIC PLASTIC
## PLASTIC 0.0093      -
## STATIC  4.9e-05     1.0000
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
##  Kruskal-Wallis rank sum test
##
## data:  final_population_poison by condition
## Kruskal-Wallis chi-squared = 3.3994, df = 2, p-value = 0.1827
```

5.5.3 Poison instruction execution along final dominant lineage (cummulative)

```r
ggplot(summary_data, aes(x=condition, y=dominant_lineage_times_poison_executed, fill=condition))
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_y_continuous(
    name="poison instruction executions (dominant lineage)",
    trans="pseudo_log",
    breaks=c(10,1000,100000),
    limits=c(-1,100000)
  ) +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both
  ) +
```

```
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/final-dominant-lineage-poison-log.pdf"),
    width=15,
    height=10
  )
```




```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
```

```
      formula=dominant_lineage_times_poison_executed~condition,
      data=stat_data
    )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
  if (kt$p.value > 0.05) { next }
  print(
    pairwise.wilcox.test(
      x=stat_data$dominant_lineage_times_poison_executed,
      g=stat_data$condition,
      p.adjust.method="bonferroni"
    )
  )
}
```

```
## [1] "PENALTY: 0"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 178.84, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_times_poison_executed and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.0018
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 178.62, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_times_poison_executed and stat_data$condition
```

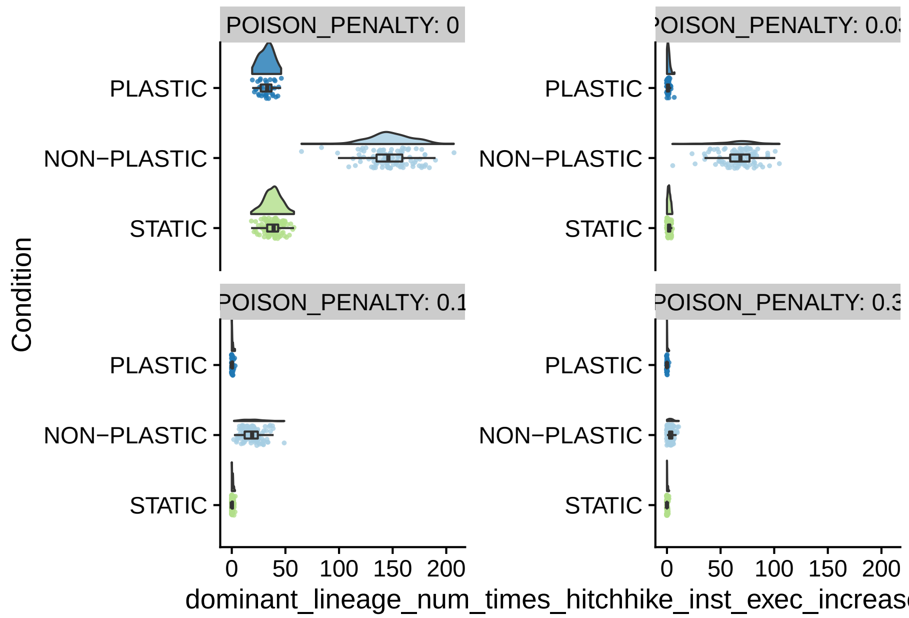
```
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.011
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 184.83, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_times_poison_executed and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.21
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 149.48, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_times_poison_executed and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 4.4e-16     -
## STATIC  < 2e-16     0.84
##
## P value adjustment method: bonferroni
```

5.6 Characterizing mutations that increase poison instruction execution

5.6.1 Number of offspring along dominant lineage with increase in poison instruction execution

```
ggplot(summary_data, aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases,
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/final-dominant-lineage-poison-increase-num-mutants-log.png")
    width=15,
    height=10
```

```
  )
```




```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
      formula=dominant_lineage_num_times_hitchhike_inst_exec_increases~condition,
      data=stat_data
    )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
  if (kt$p.value > 0.05) { next }
  print(
    pairwise.wilcox.test(
      x=stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases,
      g=stat_data$condition,
      p.adjust.method="bonferroni"
    )
```

```
  )
}
```

```
## [1] "PENALTY: 0"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases by condition
## Kruskal-Wallis chi-squared = 179.79, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_data$condit
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     0.00046
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases by condition
## Kruskal-Wallis chi-squared = 179.35, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_data$condit
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.03
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases by condition
## Kruskal-Wallis chi-squared = 185.34, df = 2, p-value < 2.2e-16
##
##
```

```
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_d
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.27
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases by condition
## Kruskal-Wallis chi-squared = 146.35, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_d
##
##         NON-PLASTIC PLASTIC
## PLASTIC 7.8e-16     -
## STATIC  < 2e-16     0.86
##
## P value adjustment method: bonferroni
```

```
sum(filter(summary_data, condition=="NON-PLASTIC" & POISON_PENALTY==0.1)$dominant_linea
```

```
## [1] 1916
```

```
sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_lineage_n
```

```
## [1] 18
```

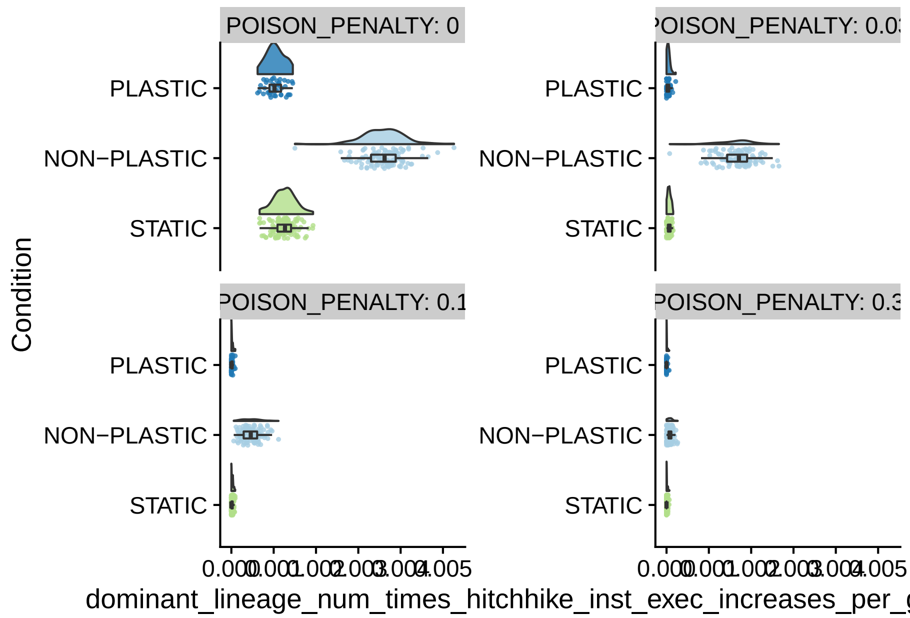
```
sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_lineage_nu
```

```
## [1] 58
```

```
# sum(filter(summary_data, condition=="NON-PLASTIC" & POISON_PENALTY==0.1)$dominant_li
# sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_lineag
# sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_lineage_
```

5.6.2 Per-generation increases in poison instruction execution

```
summary_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation <- summary_d
ggplot(summary_data, aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases_
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/final-dominant-lineage-poison-increase-per-generation.png"),
    width=15,
    height=10
  )
```




```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
      formula=dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation~c
      data=stat_data
    )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
  if (kt$p.value > 0.05) { next }
  print(
    pairwise.wilcox.test(
      x=stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generati
      g=stat_data$condition,
      p.adjust.method="bonferroni"
    )
  )
}
```

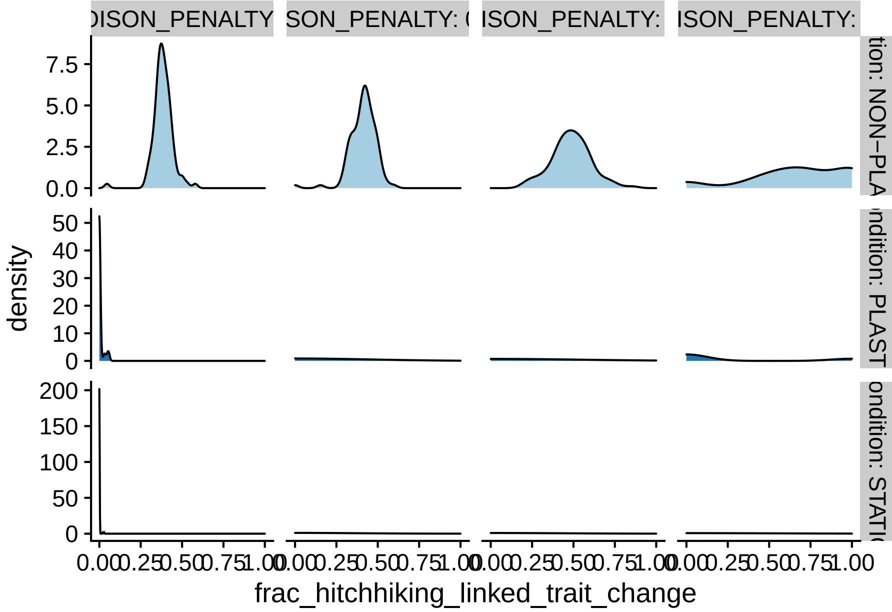
```
## [1] "PENALTY: 0"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation by condition
## Kruskal-Wallis chi-squared = 180.05, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation and s
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     7.8e-05
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation by condition
## Kruskal-Wallis chi-squared = 176.25, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation and s
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.019
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation by condition
## Kruskal-Wallis chi-squared = 184.17, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation and s
##
```

```
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.2
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation by co
## Kruskal-Wallis chi-squared = 140.99, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases_per_genera
##
##         NON-PLASTIC PLASTIC
## PLASTIC 2.2e-15     -
## STATIC  < 2e-16     0.79
##
## P value adjustment method: bonferroni
```

5.6.3 What fraction of mutations that increase poison instruction execution co-occur with base trait changes?

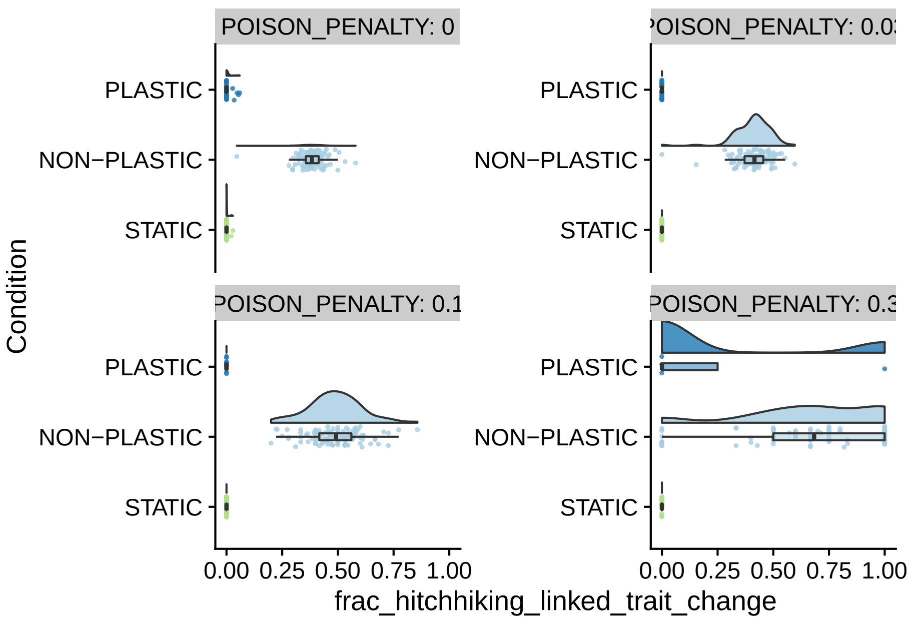
```
summary_data$frac_hitchhiking_linked_trait_change <- summary_data$dominant_lineage_num_

ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0)
  geom_density() +
  facet_grid(
    condition~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
```

```
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-frac_hitchhiking_linked_trait_change.png"),
    width=15,
    height=10
  )
```




```
ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0 ), aes(x=c
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
```

```
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  )
```




```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty & dominant_lineage_num_time
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
```

```
  kt <- kruskal.test(
      formula=frac_hitchhiking_linked_trait_change~condition,
      data=stat_data
    )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
  if (kt$p.value > 0.05) { next }
  print(
    pairwise.wilcox.test(
      x=stat_data$frac_hitchhiking_linked_trait_change,
      g=stat_data$condition,
      p.adjust.method="bonferroni",
      exact=FALSE
    )
  )
}
```

```
## [1] "PENALTY: 0"
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_hitchhiking_linked_trait_change by condition
## Kruskal-Wallis chi-squared = 211.29, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$frac_hitchhiking_linked_trait_change and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.031
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_hitchhiking_linked_trait_change by condition
## Kruskal-Wallis chi-squared = 186.88, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
```

```
##
## data:  stat_data$frac_hitchhiking_linked_trait_change and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 2.9e-16     -
## STATIC  < 2e-16     -
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_hitchhiking_linked_trait_change by condition
## Kruskal-Wallis chi-squared = 113.72, df = 2, p-value < 2.2e-16
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$frac_hitchhiking_linked_trait_change and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 3.3e-08     -
## STATIC  < 2e-16     -
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
##  Kruskal-Wallis rank sum test
##
## data:  frac_hitchhiking_linked_trait_change by condition
## Kruskal-Wallis chi-squared = 34.791, df = 2, p-value = 2.788e-08
##
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  stat_data$frac_hitchhiking_linked_trait_change and stat_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 0.26        -
## STATIC  2.4e-08     0.18
##
## P value adjustment method: bonferroni
```

```r
denom <- sum(filter(summary_data, condition=="NON-PLASTIC" & POISON_PENALTY==0.1)$domi
num <- sum(filter(summary_data, condition=="NON-PLASTIC" & POISON_PENALTY==0.1)$dominar
paste0("NON-PLASTIC: ", num/denom, "(", num, "/", denom, ")")
```

```
## [1] "NON-PLASTIC: 0.498956158663883(956/1916)"
```

```
denom <- sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_lineage_nu
num <- sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_lineage_num_
paste0("PLASTIC: ", num/denom, " (", num, "/", denom, ")")
```

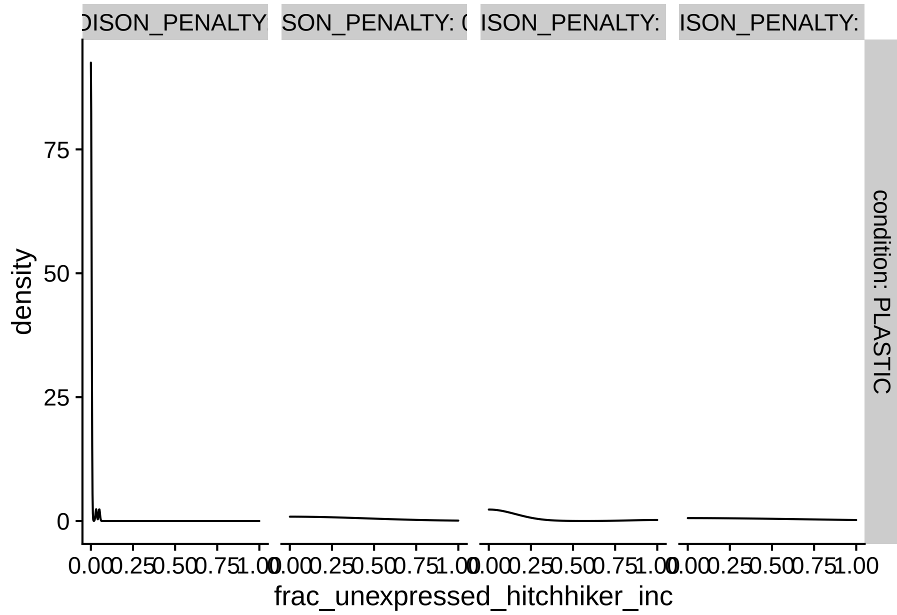
```
## [1] "PLASTIC: 0 (0/18)"
```

```
denom <- sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_lineage_num
num <- sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_lineage_num_t
paste0("STATIC: ", num/denom, " (", num, "/", denom, ")")
```

```
## [1] "STATIC: 0 (0/58)"
```

5.7 What fraction of poison execution increases occur in unexpressed phenotype (as cryptic

```
summary_data$frac_unexpressed_hitchhiker_inc <- summary_data$dominant_lineage_num_times_hitchhike
summary_data$frac_expressed_hitchiker_inc <- summary_data$dominant_lineage_num_times_hitchhike_in

ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0 & conditio
  geom_density() +
  facet_grid(
    condition~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  theme(
    legend.position="none"
  )
```




```
denom <- sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_
num <- sum(filter(summary_data, condition=="PLASTIC" & POISON_PENALTY==0.1)$dominant_li
paste0("PLASTIC: ", num/denom, " (", num, "/", denom, ")")
```

```
## [1] "PLASTIC: 0.0555555555555556 (1/18)"
```

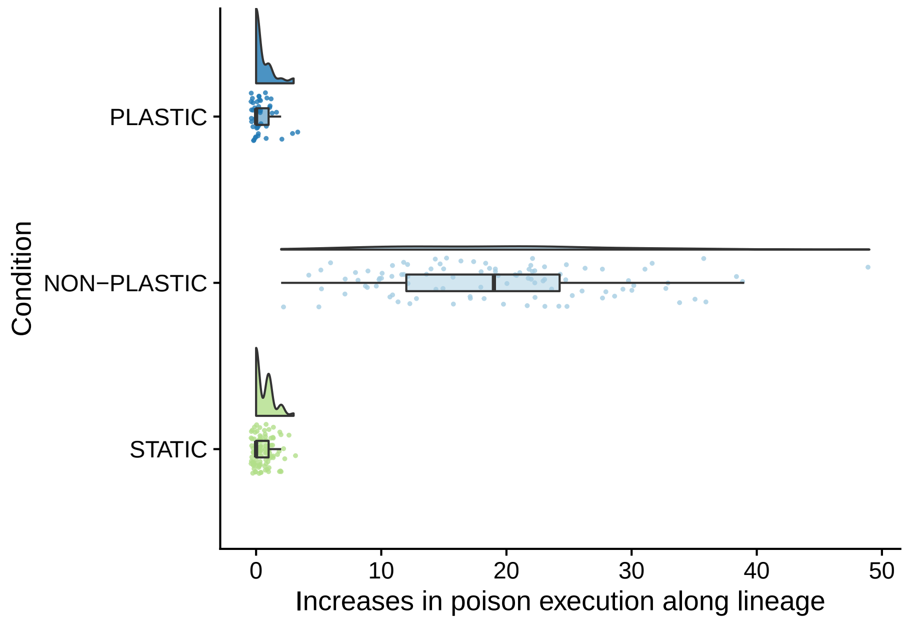
5.8 Manuscript figures

```
poison_penalty <- 0.1
```

5.8.1 Total poison execution increases along lineage.

```
poison_increases_fig <- ggplot(
    filter(summary_data, POISON_PENALTY==poison_penalty),
    aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases, fill=c
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
```

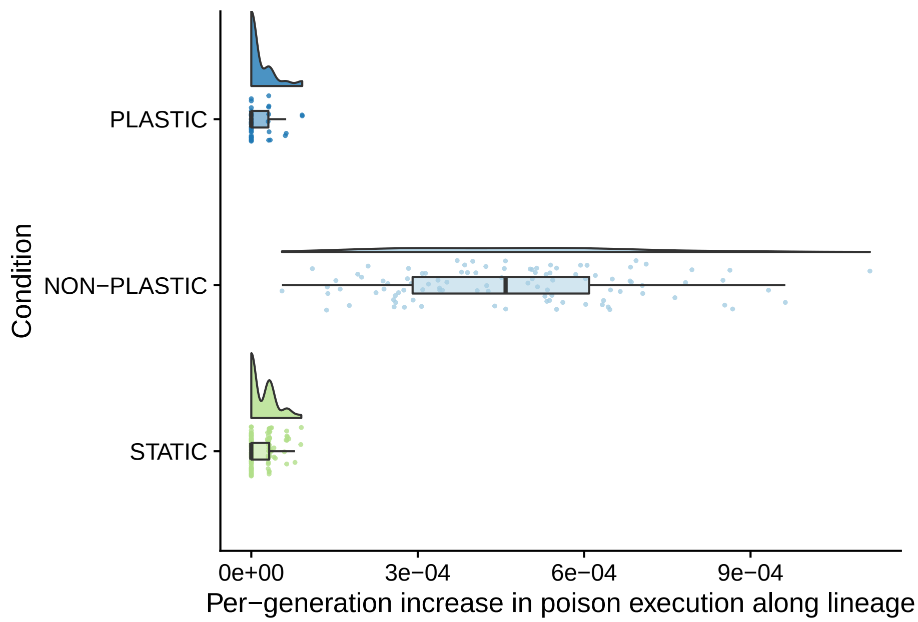
```
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Increases in poison execution along lineage",
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
  coord_flip()

poison_increases_fig
```



5.8.2 Per-generation poison execution increases along lineage

```
poison_increases_per_gen_fig <- ggplot(
    filter(summary_data, POISON_PENALTY==poison_penalty),
    aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases_per_ge
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
```

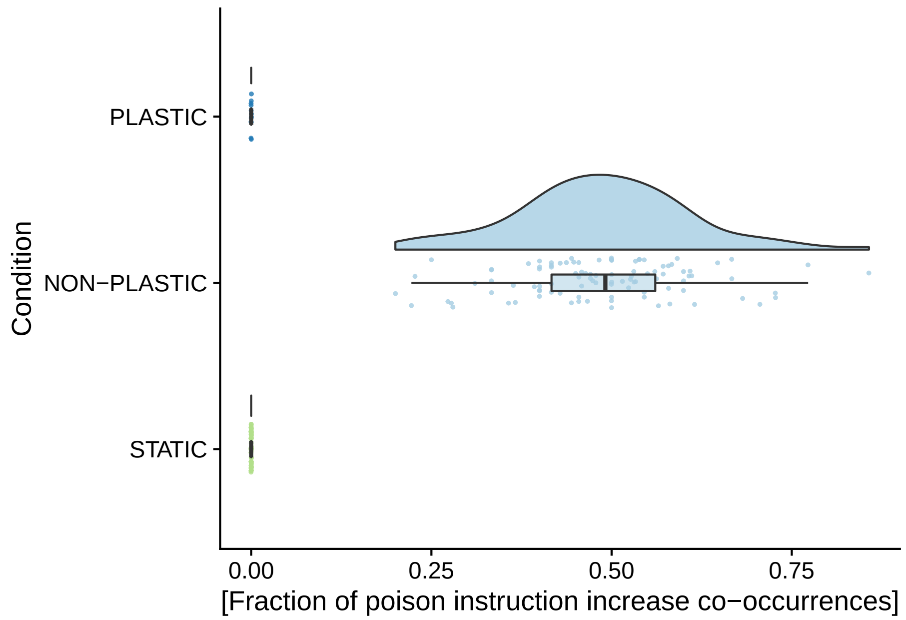
```
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="Per-generation increase in poison execution along lineage",
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
  coord_flip()

poison_increases_per_gen_fig
```



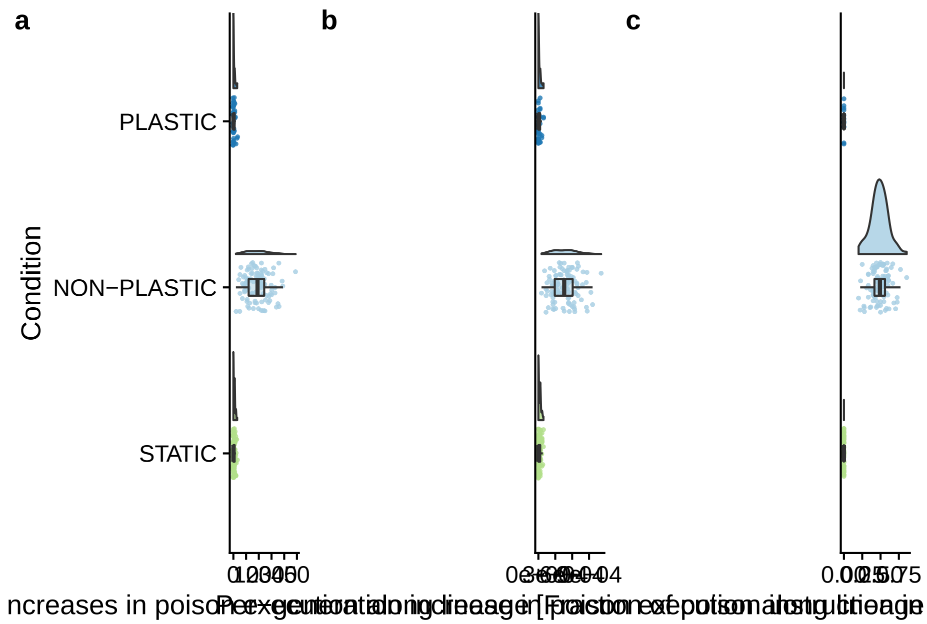
5.8.3 Co-occurrence

```
linked_trait_change_fig <- ggplot(
    filter(summary_data, POISON_PENALTY==poison_penalty & dominant_lineage_num_times_hitchhike_in
    aes(x=condition, y=frac_hitchhiking_linked_trait_change, fill=condition)
```

```
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order
  ) +
  scale_y_continuous(
    name="[Fraction of poison instruction increase co-occurrences]",
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
  coord_flip()
linked_trait_change_fig
```




```
grid <- plot_grid(
  poison_increases_fig,
  poison_increases_per_gen_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_blank(),a
  linked_trait_change_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_blank(),axis.t
  nrow=1,
  align="v",
  labels="auto"
)
save_plot(
   paste0(working_directory, "plots/", "poison-accumulation-panel.pdf"),
   grid,
   base_height=6,
   base_asp=3/1
)
grid
```


a
b
c
Condition
PLASTIC
NON−PLASTIC
STATIC

Chapter 6

Regulation in Avida

6.1 Overview

```
total_updates <- 200000
replicates <- 100

all_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")

# Relative location of data.
working_directory <- "experiments/2021-02-08-evo-dynamics/analysis/" # << For bookdown
# working_directory <- "./"                                            # << For local analysis
```

6.2 Analysis dependencies

Load all required R libraries.

```
library(ggplot2)
library(tidyverse)
library(cowplot)
library(RColorBrewer)
library(Hmisc)
library(boot)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9
```

These analyses were conducted/knitted with the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

6.3 Setup

```
trace_summary_data_loc <- paste0(working_directory, "data/trace_summary.csv")
trace_summary_data <- read.csv(trace_summary_data_loc, na.strings="NONE")

trace_summary_data$DISABLE_REACTION_SENSORS <- as.factor(trace_summary_data$DISABLE_REA
trace_summary_data$chg_env <- trace_summary_data$chg_env == "True"
trace_summary_data$sensors <- trace_summary_data$DISABLE_REACTION_SENSORS == "0"


env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
    return("Sensors")
  } else {
    return("No sensors")
  }
}
```

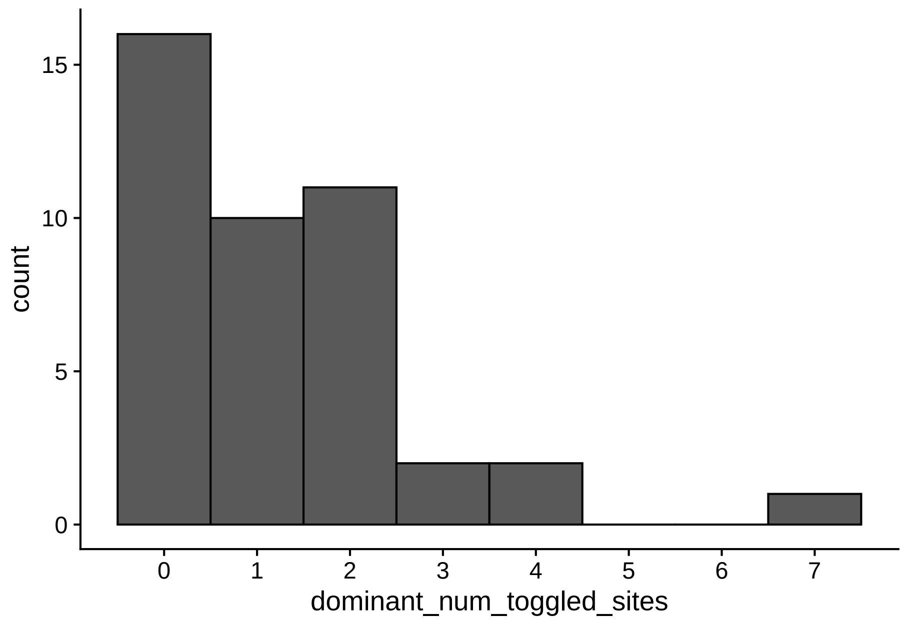
```
# note that this labeler makes assumptions about how we set up our experiment
condition_label_fun <- function(has_sensors, env_chg) {
  if (has_sensors && env_chg) {
    return("PLASTIC")
  } else if (env_chg) {
    return("NON-PLASTIC")
  } else {
    return("STATIC")
  }
}

trace_summary_data$env_label <- mapply(
  env_label_fun,
  trace_summary_data$chg_env
)
trace_summary_data$sensors_label <- mapply(
  sensors_label_fun,
  trace_summary_data$sensors
)
trace_summary_data$condition <- mapply(
  condition_label_fun,
  trace_summary_data$sensors,
  trace_summary_data$chg_env
)

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
```

6.4 How many instructions do plastic genomes toggle depending on environmental context?

```
ggplot(trace_summary_data, aes(x=dominant_num_toggled_sites)) +
    geom_histogram(
      binwidth=1,
      color="black"
    ) +
    scale_fill_brewer(
      palette="Paired"
    ) +
    scale_color_brewer(
      palette="Paired"
```

```
  ) +
  scale_x_continuous(
    breaks=seq(0, max(trace_summary_data$dominant_num_toggled_sites)+1)
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(paste0(working_directory, "plots/", "toggled-sites.png"))
```

```
## Saving 6.5 x 4.5 in image
```



6.5 What is the distrubution of toggled sequence sizes?

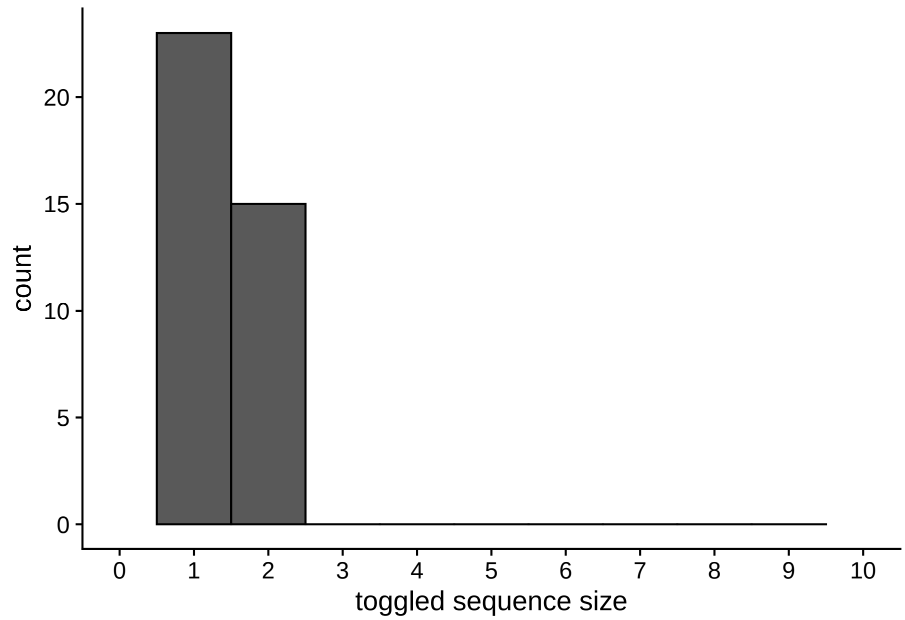
```
chunk_sizes <- data.frame(
  size=integer()
)
for (sizes in trace_summary_data$dominant_toggled_chunk_sizes) {
  if (sizes == "") { next }
  sizes <- unlist(lapply(str_split(sizes, ';'), as.integer))
  chunk_sizes <- rbind(chunk_sizes, data.frame(size=c(sizes)))
}
```

```
ggplot(chunk_sizes, aes(x=size)) +
    geom_histogram(
      binwidth=1,
      color="black"
    ) +
    scale_fill_brewer(
      palette="Paired"
    ) +
    scale_color_brewer(
      palette="Paired"
    ) +
    scale_x_continuous(
      name="toggled sequence size",
      breaks=seq(0, 10),
      limits=c(0, 10)
    ) +
    theme(
      legend.position="none"
    ) +
    ggsave(paste0(working_directory, "plots/", "toggled-chunk-sizes.png"))
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



count
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
toggled sequence size

Chapter 7

Evolutionary change (variable length genomes)

7.1 Overview

```
total_updates <- 200000
replicates <- 100

all_traits <- c("not","nand","and","ornot","or","andnot")
traits_set_a <- c("not", "and", "or")
traits_set_b <- c("nand", "ornot", "andnot")

# Relative location of data.
working_directory <- "experiments/2021-01-30-evo-dynamics/analysis/" # << For bookdown
# working_directory <- "./"                                          # << For local analysis
```

7.2 Analysis dependencies

Load all required R libraries.

```
library(ggplot2)
library(tidyverse)
library(cowplot)
library(RColorBrewer)
library(Hmisc)
library(boot)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9
```

These analyses were conducted/knitted with the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname       Lost Library Book
```

7.3 Setup

```
summary_data_loc <- paste0(working_directory, "data/aggregate.csv")
summary_data <- read.csv(summary_data_loc, na.strings="NONE")

summary_data$DISABLE_REACTION_SENSORS <- as.factor(summary_data$DISABLE_REACTION_SENSO
summary_data$chg_env <- summary_data$chg_env == "True"
summary_data$dominant_plastic_odd_even <- as.factor(summary_data$dominant_plastic_odd_e
summary_data$sensors <- summary_data$DISABLE_REACTION_SENSORS == "0"
summary_data$is_plastic <- summary_data$dominant_plastic_odd_even == "True"

env_label_fun <- function(chg_env) {
  if (chg_env) {
    return("Fluctuating")
  } else {
    return("Constant")
  }
}

sensors_label_fun <- function(has_sensors) {
  if (has_sensors) {
    return("Sensors")
  } else {
    return("No sensors")
  }
```

```
}

# note that this labeler makes assumptions about how we set up our experiment
condition_label_fun <- function(has_sensors, env_chg) {
  if (has_sensors && env_chg) {
    return("PLASTIC")
  } else if (env_chg) {
    return("NON-PLASTIC")
  } else {
    return("STATIC")
  }
}

summary_data$env_label <- mapply(
  env_label_fun,
  summary_data$chg_env
)
summary_data$sensors_label <- mapply(
  sensors_label_fun,
  summary_data$sensors
)
summary_data$condition <- mapply(
  condition_label_fun,
  summary_data$sensors,
  summary_data$chg_env
)

condition_order = c(
  "STATIC",
  "NON-PLASTIC",
  "PLASTIC"
)

####### misc #######
# Configure our default graphing theme
theme_set(theme_cowplot())
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
```

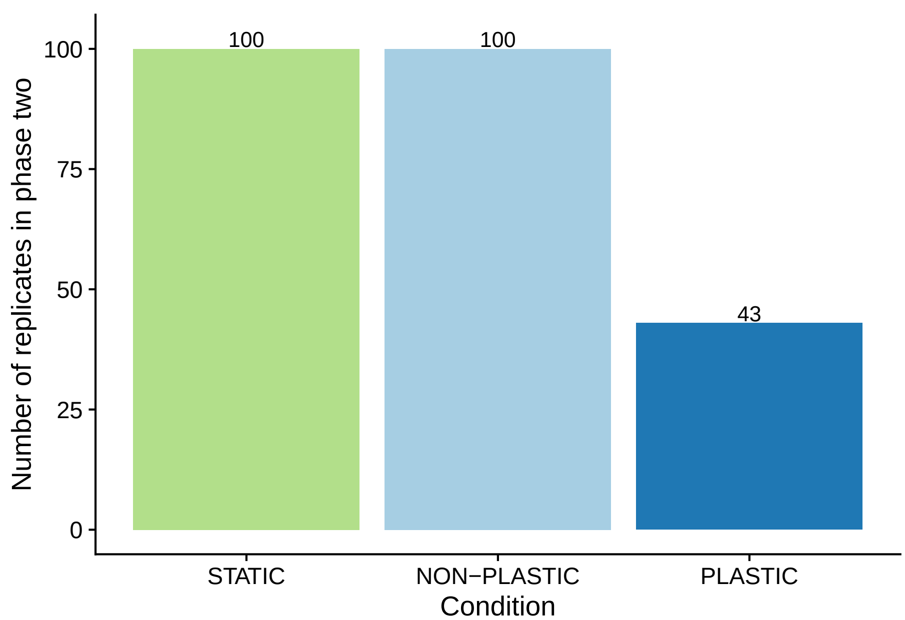
7.4 Evolution of phenotypic plasticity

For sensor-enabled populations in fluctuating environments, we only transfered populations containing an optimally plastic genotype to phase-two.

```
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
```

```
## `summarise()` has grouped output by 'sensors', 'env_label'. You can override using t
```

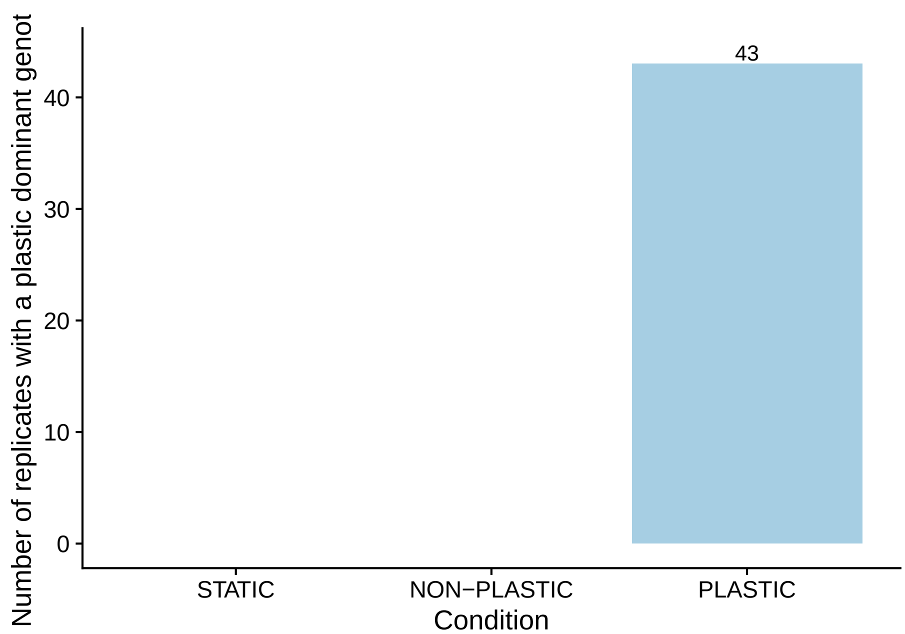
```
ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  ylab("Number of replicates in phase two") +
  theme(
    legend.position="none"
  )
```



We can confirm our expectation that the dominant genotypes in non-plastic conditions are not phenotypically plastic.

```
summary_data_grouped = dplyr::group_by(summary_data, condition, is_plastic)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
```

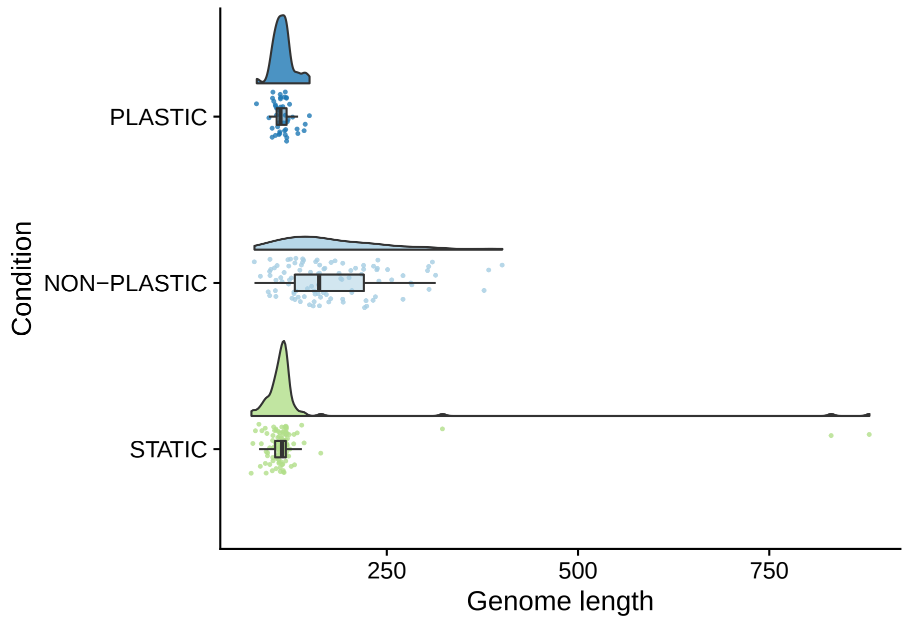
```r
ggplot(filter(summary_data_group_counts, is_plastic), aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  geom_text(aes(label=n, y=n+1)) +
  ylab("Number of replicates with a plastic dominant genotype") +
  theme(
    legend.position="none"
  )
```



7.5 Genome length

Single-instruction insertions and deletions were possible for this experiment, so genome size also evolved.

```
ggplot(summary_data, aes(x=condition, y=dominant_genome_length, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  ylab("Genome length") +
  theme(
    legend.position="none"
  )
```




```r
kruskal.test(
  formula=dominant_genome_length~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_genome_length by condition
## Kruskal-Wallis chi-squared = 82.798, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$dominant_genome_length,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_genome_length and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC 1.8e-10     -
## STATIC  < 2e-16     1
##
```

```
## P value adjustment method: bonferroni
```

```r
median(filter(summary_data, condition=="PLASTIC")$phylo_mrca_changes)
```

```
## [1] 45
```

```r
median(filter(summary_data, condition=="STATIC")$phylo_mrca_changes)
```

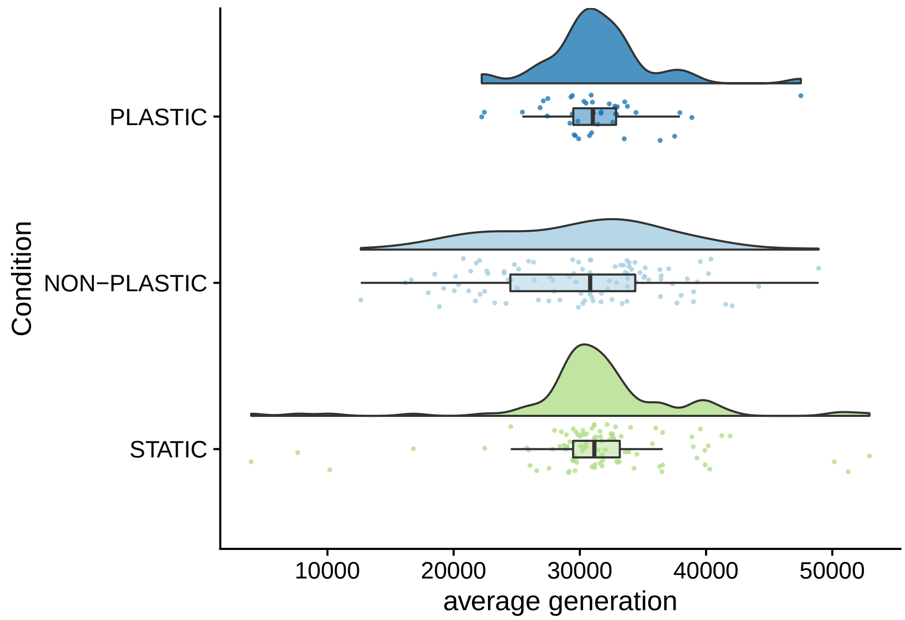
```
## [1] 47
```

```r
median(filter(summary_data, condition=="NON-PLASTIC")$phylo_mrca_changes)
```

```
## [1] 393
```

7.6 Average generation

```r
ggplot(summary_data, aes(x=condition, y=time_average_generation, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  ylab("average generation") +
  theme(
    legend.position="none"
  )
```




```r
median(filter(summary_data, condition=="PLASTIC")$time_average_generation)
```

```
## [1] 31028.6
```

```r
median(filter(summary_data, condition=="STATIC")$time_average_generation)
```

```
## [1] 31147.5
```

```r
median(filter(summary_data, condition=="NON-PLASTIC")$time_average_generation)
```

```
## [1] 30817.95
```

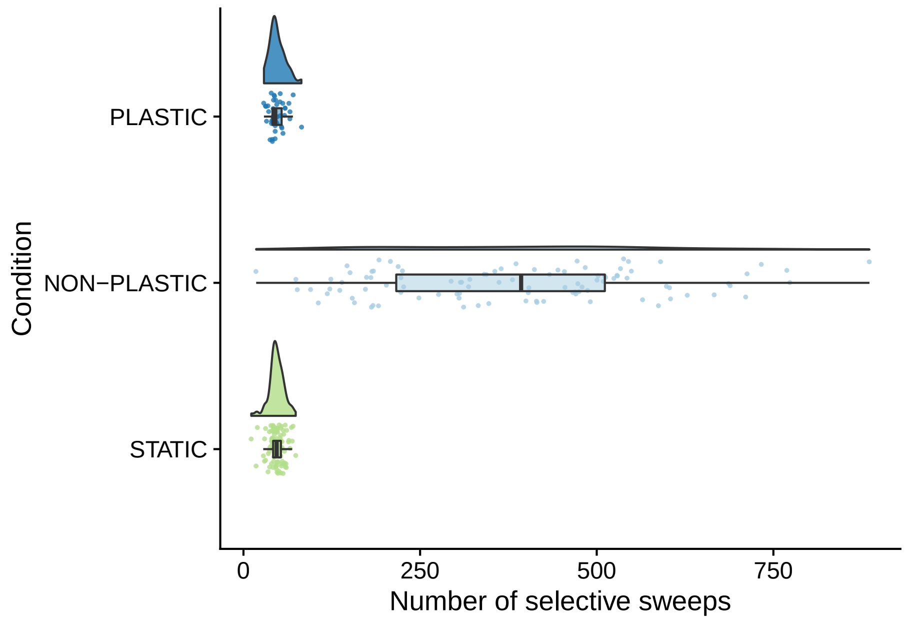
```r
kruskal.test(
  formula=time_average_generation~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  time_average_generation by condition
## Kruskal-Wallis chi-squared = 1.3804, df = 2, p-value = 0.5015
```

7.7 Coalescence events

The number of times the most recent common ancestor changes gives us the number of selective sweeps that occur during the experiment.

```
ggplot(summary_data, aes(x=condition, y=phylo_mrca_changes, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("Number of selective sweeps") +
  theme(
    legend.position="none"
  )
```




```
paste0(
  "PLASTIC: ",
  median(filter(summary_data, condition=="PLASTIC")$phylo_mrca_changes)
)
```

```
## [1] "PLASTIC: 45"
```

```
paste0(
  "STATIC: ",
  median(filter(summary_data, condition=="STATIC")$phylo_mrca_changes)
)
```

```
## [1] "STATIC: 47"
```

```
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$phylo_mrca_changes)
)
```

```
## [1] "NON-PLASTIC: 393"
```

```
kruskal.test(
  formula=phylo_mrca_changes~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  phylo_mrca_changes by condition
## Kruskal-Wallis chi-squared = 168.89, df = 2, p-value < 2.2e-16
```

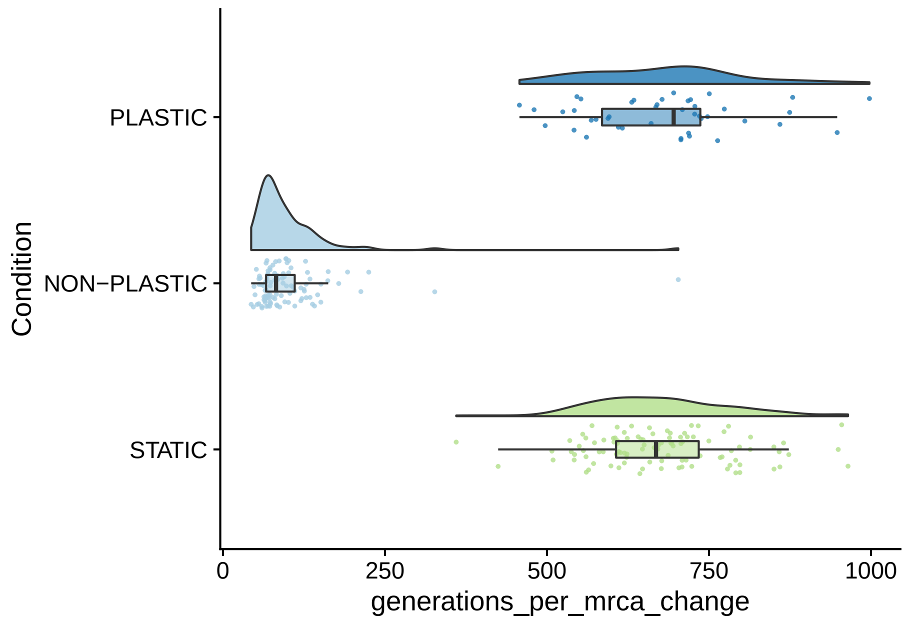
```
pairwise.wilcox.test(
  x=summary_data$phylo_mrca_changes,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$phylo_mrca_changes and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      1
##
## P value adjustment method: bonferroni
```

7.7.1 Average number of generations between selective sweeps

```
summary_data$generations_per_mrca_change <- summary_data$time_average_generation / summ

ggplot(summary_data, aes(x=condition, y=generations_per_mrca_change, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
```

```
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  )
```




```
paste0(
  "PLASTIC: ",
  median(filter(summary_data, condition=="PLASTIC")$generations_per_mrca_change)
)
```

```
## [1] "PLASTIC: 695.504761904762"
```

```
paste0(
  "STATIC: ",
  median(filter(summary_data, condition=="STATIC")$generations_per_mrca_change)
)
```

```
## [1] "STATIC: 668.25523255814"
```

```
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$generations_per_mrca_change)
)
```

```
## [1] "NON-PLASTIC: 81.9208459944751"
```

```
kruskal.test(
  formula=generations_per_mrca_change~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  generations_per_mrca_change by condition
## Kruskal-Wallis chi-squared = 171.73, df = 2, p-value < 2.2e-16
```

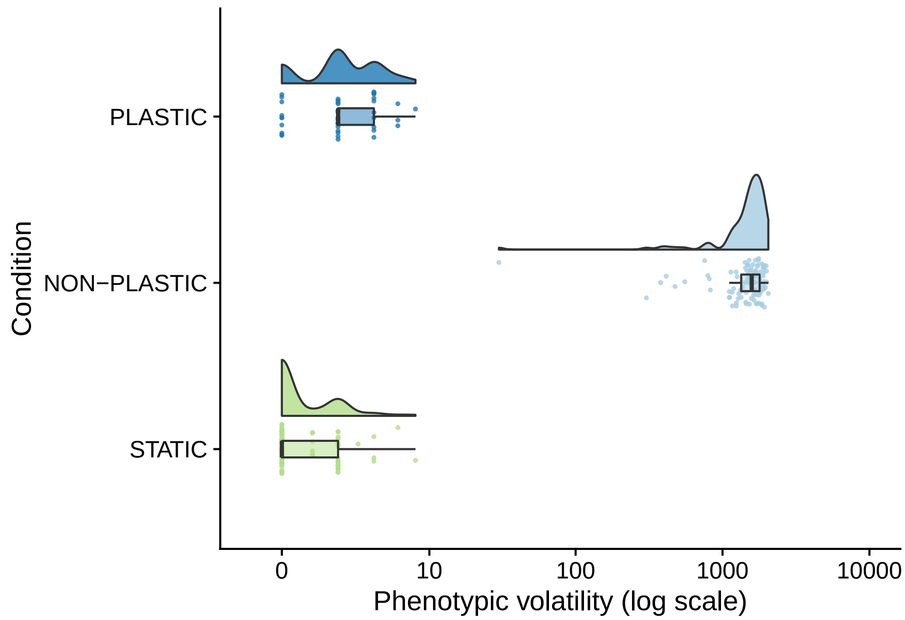
```
pairwise.wilcox.test(
  x=summary_data$generations_per_mrca_change,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$generations_per_mrca_change and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      1
##
## P value adjustment method: bonferroni
```

7.8 Phenotypic volatility along the dominant lineage

```
ggplot(summary_data, aes(x=condition, y=dominant_lineage_trait_volatility, fill=conditi
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
```

```
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  scale_y_continuous(
    name="Phenotypic volatility (log scale)",
    trans="pseudo_log",
    breaks=c(0, 10, 100, 1000, 10000),
    limits=c(-1,10000)
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  )
```




```
paste0(
  "PLASTIC: ",
  median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_trait_volatility)
)
```

```
## [1] "PLASTIC: 2"
```

```
paste0(
  "STATIC: ",
  median(filter(summary_data, condition=="STATIC")$dominant_lineage_trait_volatility)
)
```

```
## [1] "STATIC: 0"
```

```
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_trait_volatili
)
```

```
## [1] "NON-PLASTIC: 1580"
```

```
kruskal.test(
  formula=dominant_lineage_trait_volatility~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  dominant_lineage_trait_volatility by condition
## Kruskal-Wallis chi-squared = 191.98, df = 2, p-value < 2.2e-16
```

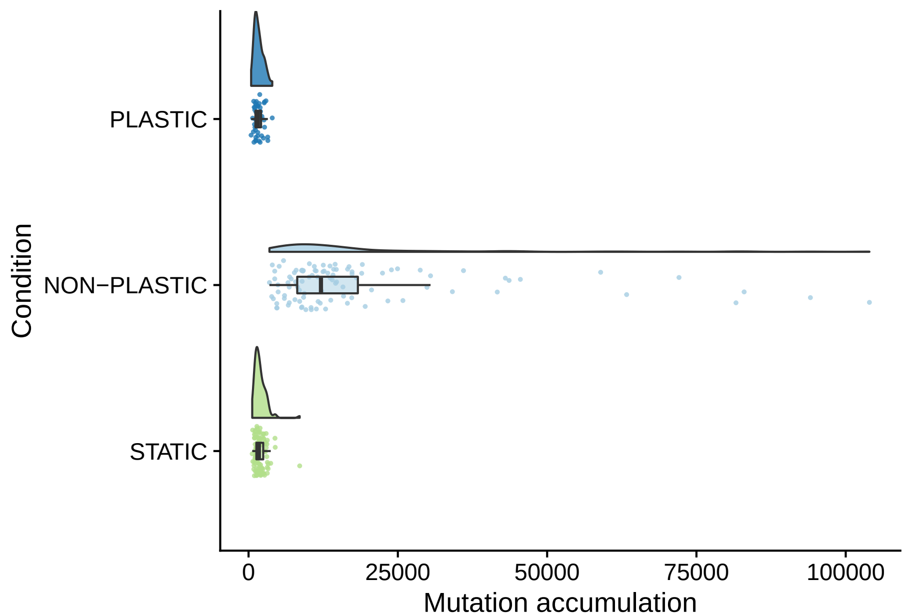
```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_volatility,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_trait_volatility and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     5.2e-08
##
## P value adjustment method: bonferroni
```

7.9 Mutation accumulation along the dominant lineage

```
ggplot(summary_data, aes(x=condition, y=dominant_lineage_total_mut_cnt, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  ylab("Mutation accumulation") +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
```

```r
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  )
```




```r
paste0(
  "PLASTIC: ",
  median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_total_mut_cnt)
)
```

```
## [1] "PLASTIC: 1552"
```

```r
paste0(
  "STATIC: ",
  median(filter(summary_data, condition=="STATIC")$dominant_lineage_total_mut_cnt)
)
```

```
## [1] "STATIC: 1724.5"
```

```r
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_total_mut_cnt)
)
```

```
## [1] "NON-PLASTIC: 12123"
```

```r
kruskal.test(
  formula=dominant_lineage_total_mut_cnt~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_total_mut_cnt by condition
## Kruskal-Wallis chi-squared = 174.38, df = 2, p-value < 2.2e-16
```

```r
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_total_mut_cnt,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$dominant_lineage_total_mut_cnt and summary_data$condition
##
##         NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.57
##
## P value adjustment method: bonferroni
```