

Supplemental Material

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

2021-02-26

Contents

1	Introduction	5
2	Validation experiment	9
2.1	Overview	9
2.2	Analysis dependencies	10
2.3	Setup	10
2.4	Evolution of phenotypic plasticity	12
3	Evolutionary change	15
3.1	Overview	15
3.2	Analysis dependencies	15
3.3	Setup	16
3.4	Evolution of phenotypic plasticity	17
3.5	Average generation	19
3.6	Selective sweeps	21
3.7	Phenotypic volatility along dominant lineage	26
3.8	Mutation accumulation along the dominant lineage	35
3.9	Depth of MRCA	52
3.10	Manuscript figures	53
4	Evolution and maintenance of novel traits	67
4.1	Overview	67
4.2	Analysis dependencies	68
4.3	Setup	69
4.4	Evolution of phenotypic plasticity	71
4.5	Final dominant novel task performance	72
4.6	Final population novel task performance	75
4.7	Population-level novel tasks discovered	77
4.8	Novel tasks along lineage of final dominant genotype	80
4.9	Extra task performance over time	99
4.10	Manuscript figures	101
5	Genetic hitchhiking	107
5.1	Overview	107

5.2	Analysis dependencies	107
5.3	Setup	108
5.4	Evolution of phenotypic plasticity	110
5.5	Hitchhiking instruction execution	112
5.6	Characterizing mutations that increase hitchhiker instruction execution	123
5.7	Manuscript figures	136

Chapter 1

Introduction

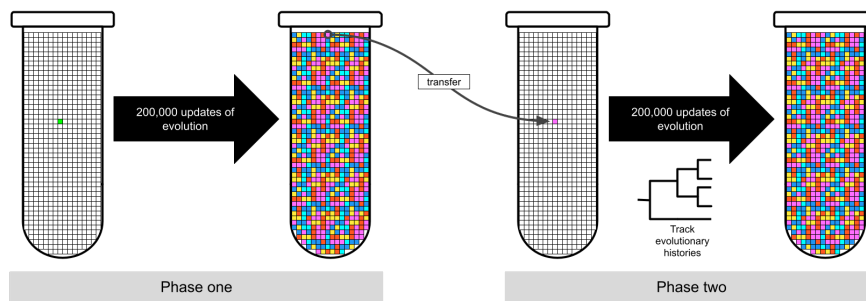


Figure 1.1: Experimental design overview

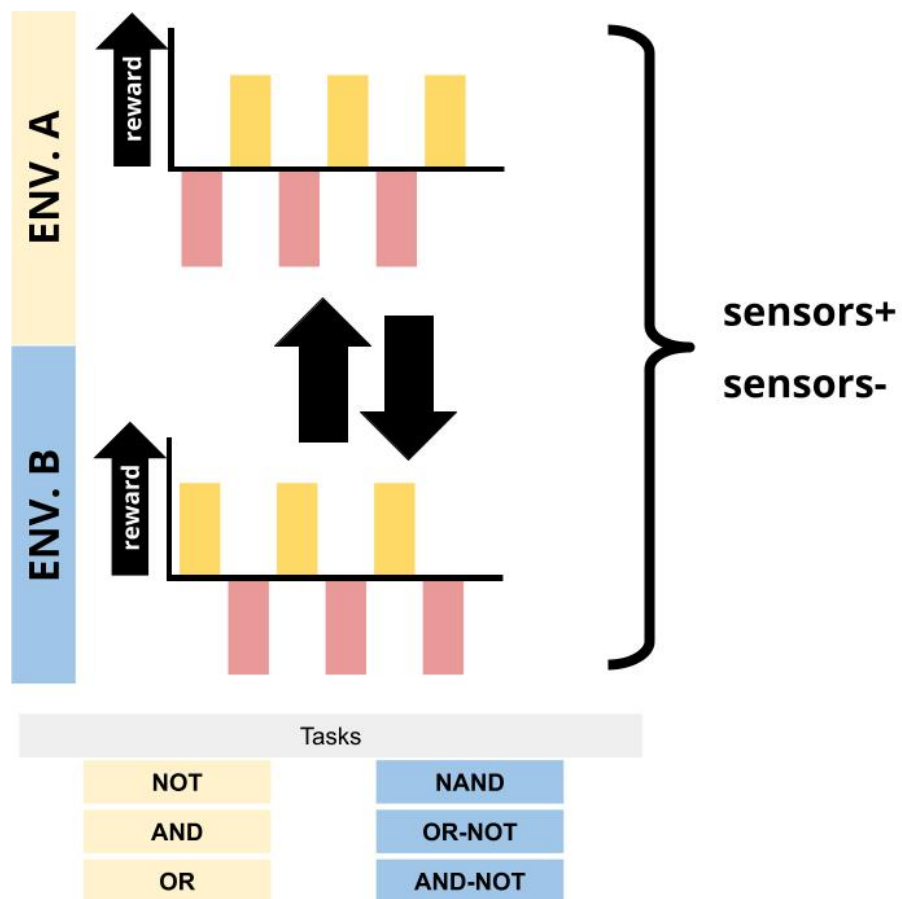


Figure 1.2: Fluctuating environment

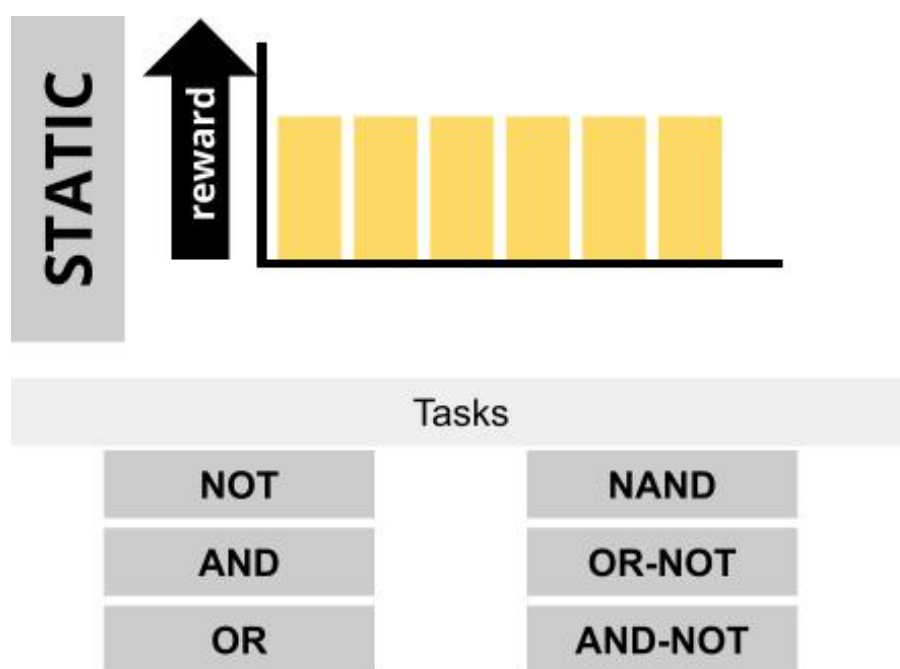


Figure 1.3: Static environment

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_sensors)
    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=="False"))
    )
  }
}

```

```

    )
  }
}

observed_plasticity <- pivot_longer(
  observed_plasticity,
  cols=c("plastic", "plastic_adaptive", "plastic_optimal", "plastic_nonadaptive", "nonplastic"),
  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=phenotype, y=phenotype_cnt)) +
  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"
  )

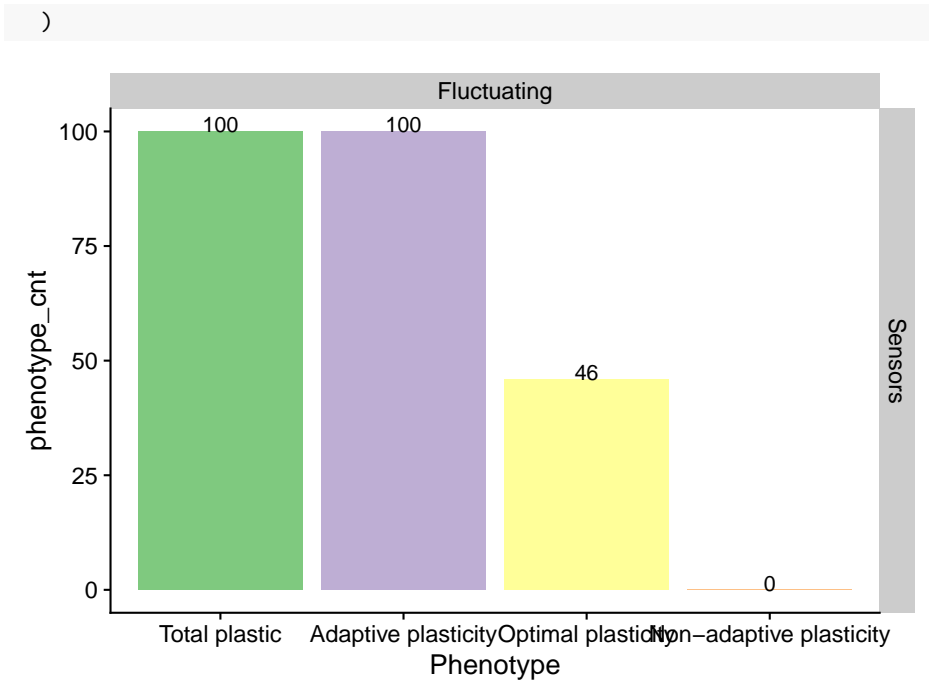
```



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 plasticity")
  ) +
  facet_grid(sensors~environment) +
  theme(
    legend.position="none"
```



Chapter 3

Evolutionary change

The effect of adaptive phenotypic plasticity on evolutionary change.

3.1 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
```

3.2 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 environ-

ment:

```
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_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"

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)

```

3.4 Evolution of phenotypic plasticity

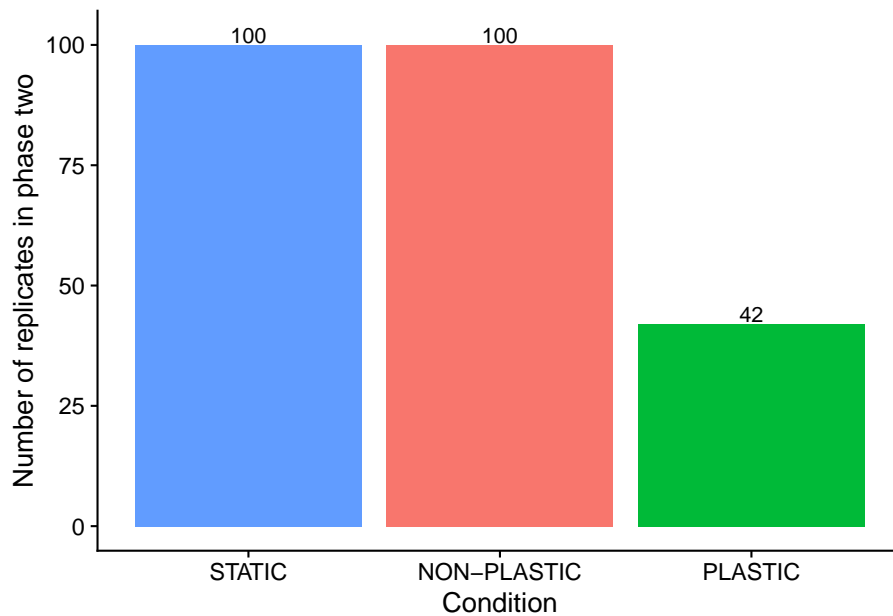
For sensor-enabled populations in fluctuating environments, we only transferred populations containing an optimally plastic genotype to phase-two.

```

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

```

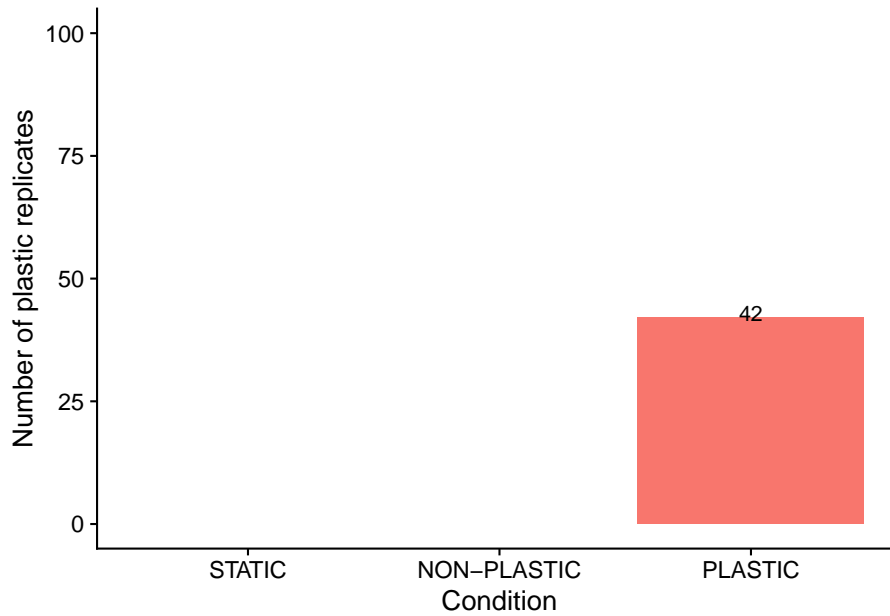
```
ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("Number of replicates in phase two") +
  theme(
    legend.position="none"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



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())
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)) +
ylab("Number of plastic replicates") +
ylim(0, 100) +
theme(
  legend.position="none"
)
```



3.5 Average generation

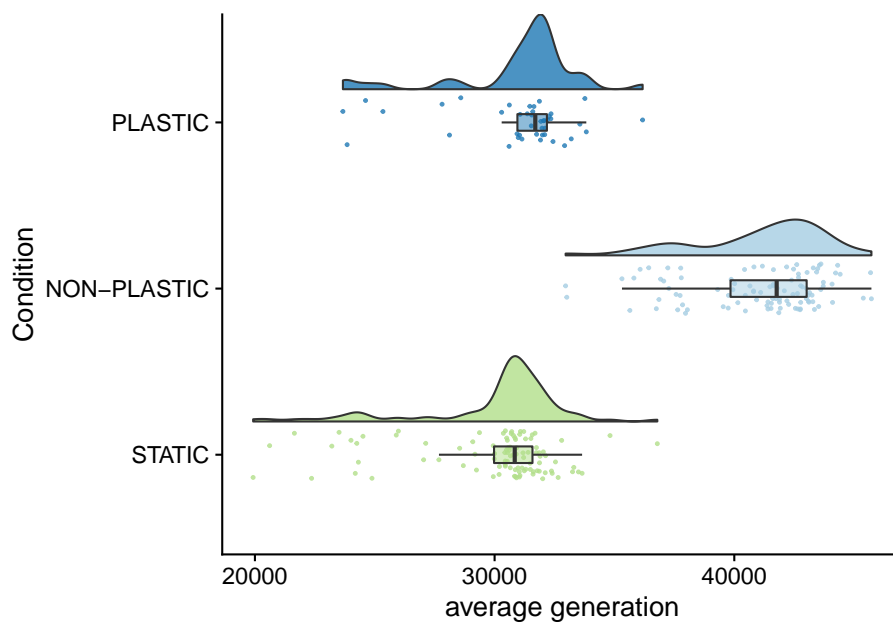
```
ggplot(summary_data, aes(x=condition, y=time_average_generation, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
 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

```

alpha = 0.5
) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
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



```

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

```

```
## [1] "PLASTIC median: 31697.65"
paste0(
  "STATIC median: ",
  median(filter(summary_data, condition=="STATIC")$time_average_generation)
)

## [1] "STATIC median: 30839.75"
paste0(
  "NON-PLASTIC median: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$time_average_generation)
)

## [1] "NON-PLASTIC median: 41768.65"
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
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
```

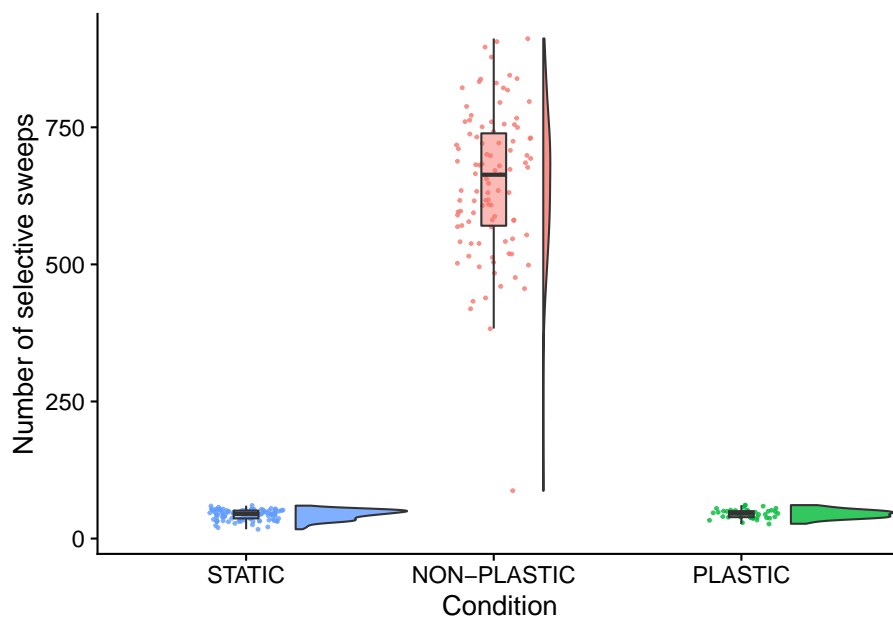
3.6 Selective sweeps

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_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: 45.5"

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

## [1] "STATIC: 45"

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

## [1] "NON-PLASTIC: 663.5"

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",
)

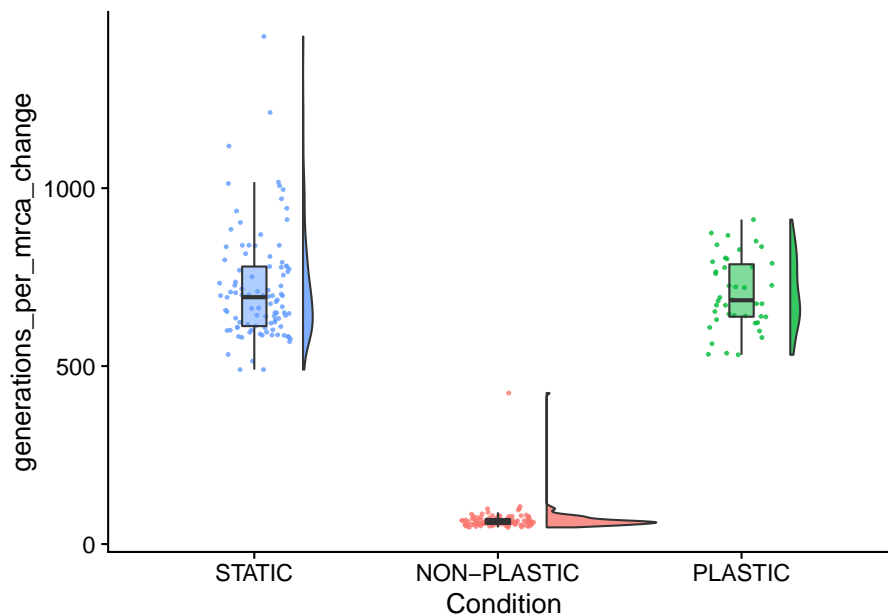
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  summary_data$phylo_mrca_changes and summary_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      1
##
## P value adjustment method: bonferroni

```

3.6.1 Average number of generations between selective sweeps

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

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

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

## [1] "PLASTIC: 685.001780758557"

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

## [1] "STATIC: 693.676265008576"

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

## [1] "NON-PLASTIC: 62.0184902295191"

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

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
```

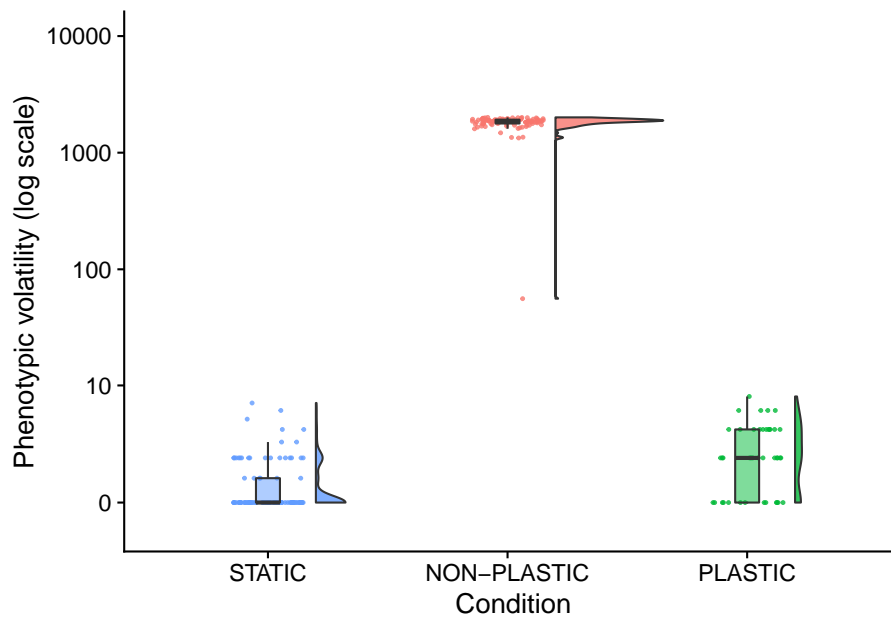
3.7 Phenotypic volatility along dominant lineage

```
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
  ) +
  scale_y_continuous(
```

```

name="Phenotypic volatility (log scale)",
trans="pseudo_log",
breaks=c(0, 10, 100, 1000, 10000),
limits=c(-1,10000)
) +
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_volatility)
)

```

```

)

## [1] "NON-PLASTIC: 1868"

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

```

3.7.1 Phenotypic volatility normalized by generations elapsed

```

summary_data$dominant_lineage_trait_volatility_per_generation <- summary_data$dominant_lineage_trait_volatility / summary_data$generations_elapsed

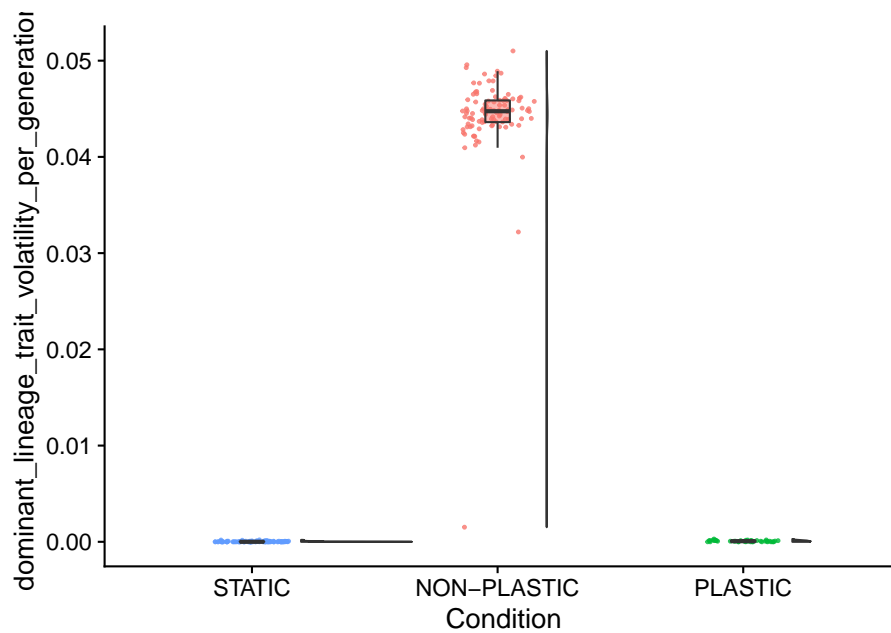
ggplot(summary_data, aes(x=condition, y=dominant_lineage_trait_volatility_per_generation)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8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

```

```

) +
geom_boxplot(
  width = .1,
  outlier.shape = NA,
  alpha = 0.5
) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
theme(
  legend.position="none"
)

```



```

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

```

```
## [1] "PLASTIC: 6.33339279717772e-05"
```

```

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

```

```
## [1] "STATIC: 0"
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_trait_volatility_per_generation)
)

## [1] "NON-PLASTIC: 0.0447440145638177"
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
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$dominant_lineage_trait_volatility_per_generation and summary_data$dominant_lineage_trait_volatility_per_lineage_step
##
##      NON-PLASTIC PLASTIC
## PLASTIC < 2e-16      -
## STATIC  < 2e-16    4.2e-06
##
## P value adjustment method: bonferroni
```

3.7.2 Phenotypic volatility normalized by lineage length

Lineage length = number of genotypes along the lineage.

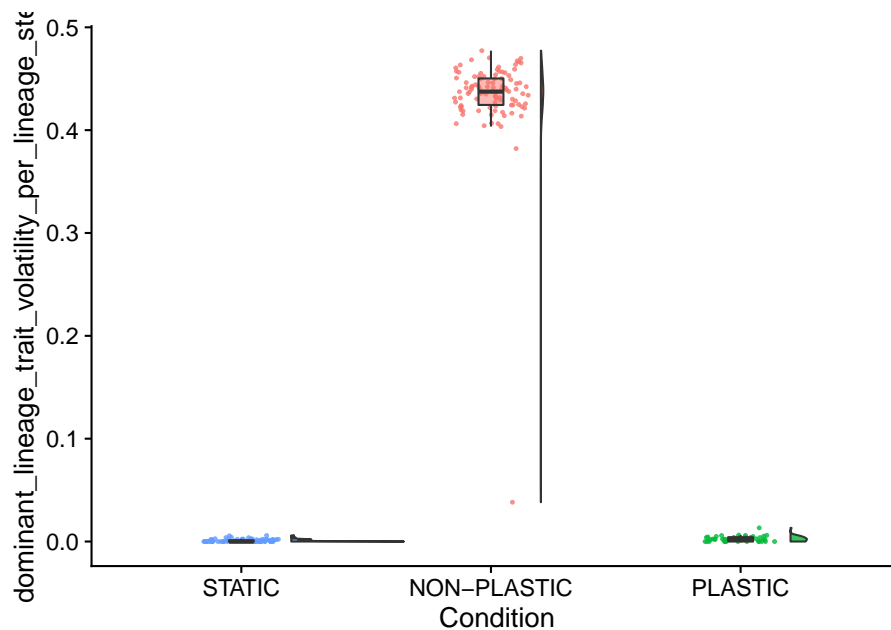
```
summary_data$dominant_lineage_trait_volatility_per_lineage_step <- summary_data$dominant_lineage_trait_volatility_per_generation / summary_data$dominant_lineage_length

ggplot(summary_data, aes(x=condition, y=dominant_lineage_trait_volatility_per_lineage_step)) +
  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
) +
theme(
legend.position="none"
)

```



```

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

```

```
## [1] "PLASTIC: 0.00224688783339238"
```

```

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

## [1] "STATIC: 0"

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

## [1] "NON-PLASTIC: 0.437482522172625"

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

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

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

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

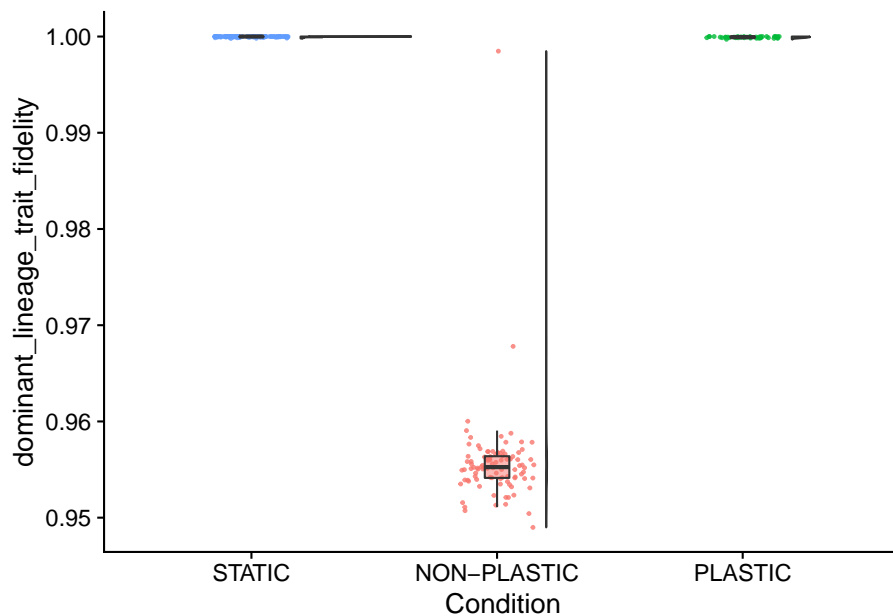
```

3.7.3 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 - summary_data$dominant_generation_died)

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



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

## [1] "PLASTIC: 0.999936666072028"

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

## [1] "STATIC: 1"

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

## [1] "NON-PLASTIC: 0.955255985436182"

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

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

3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE³⁵

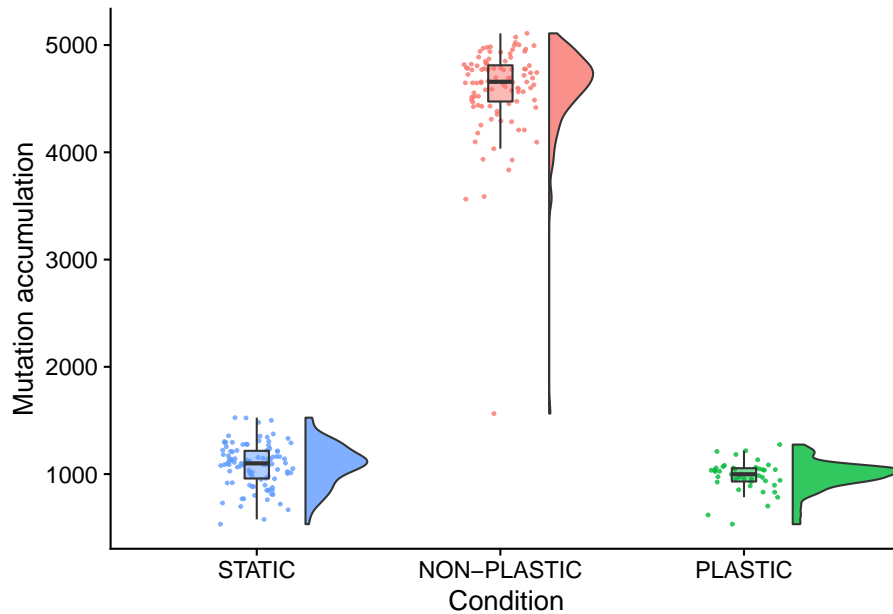
```
##
## data: dominant_lineage_trait_fidelity by condition
## Kruskal-Wallis chi-squared = 189.62, df = 2, p-value < 2.2e-16
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_fidelity,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)

##
## 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
```

3.8 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
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("Mutation accumulation") +
```

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



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

## [1] "PLASTIC: 998.5"

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

## [1] "STATIC: 1100"

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

## [1] "NON-PLASTIC: 4657.5"

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

3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE³⁷

```
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
```

3.8.1 Mutation accumulation normalized by generations elapsed

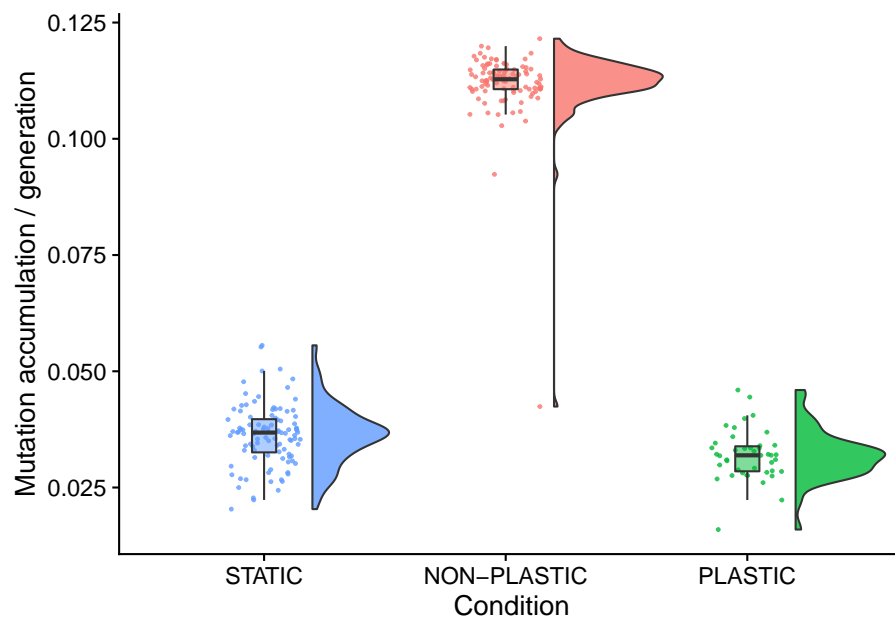
```
summary_data$mutations_per_generation <- summary_data$dominant_lineage_total_mut_cnt / summary_data$generations_elapsed

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 accumulation / generation") +
theme(
  legend.position="none"
)

```



```

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

```

```
## [1] "PLASTIC: 0.0319267181456982"
```

```

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

```

```
## [1] "STATIC: 0.0368157192941933"
```

```

paste0(
  "NON-PLASTIC: ",

```

3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE³⁹

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

## [1] "NON-PLASTIC: 0.112804526786948"

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

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
```

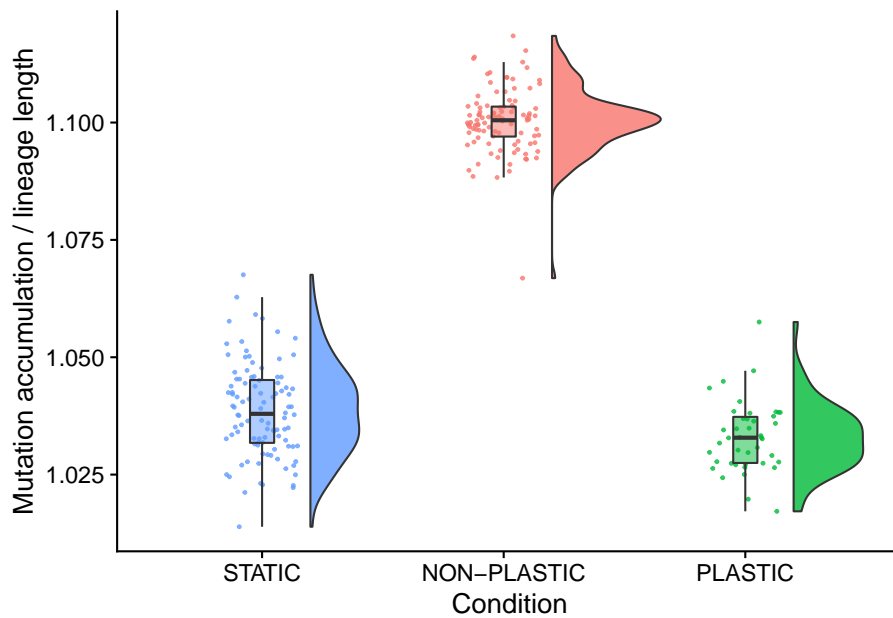
3.8.2 Mutation accumulation normalized by lineage length

```
summary_data$mutations_per_lineage_step <- summary_data$dominant_lineage_total_mut_cnt / summary_data$dominant_lineage_length
ggplot(summary_data, aes(x=condition, y=mutations_per_lineage_step, 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 accumulation / lineage length") +
theme(
  legend.position="none"
)

```



```

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

```

```
## [1] "PLASTIC: 1.0328599144651"
```

```

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

```


3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁴¹

```
## [1] "STATIC: 1.03794597464116"
paste0(
  "NON-PLASTIC: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$mutations_per_lineage_step)
)

## [1] "NON-PLASTIC: 1.10048311715591"
kruskal.test(
  formula=mutations_per_lineage_step~condition,
  data=summary_data
)

##
## Kruskal-Wallis rank sum test
##
## data: mutations_per_lineage_step by condition
## Kruskal-Wallis chi-squared = 178.92, df = 2, p-value < 2.2e-16
pairwise.wilcox.test(
  x=summary_data$mutations_per_lineage_step,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)

##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: summary_data$mutations_per_lineage_step and summary_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC <2e-16          -
## STATIC  <2e-16          0.0034
##
## P value adjustment method: bonferroni
```

3.8.3 Genotypic fidelity

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

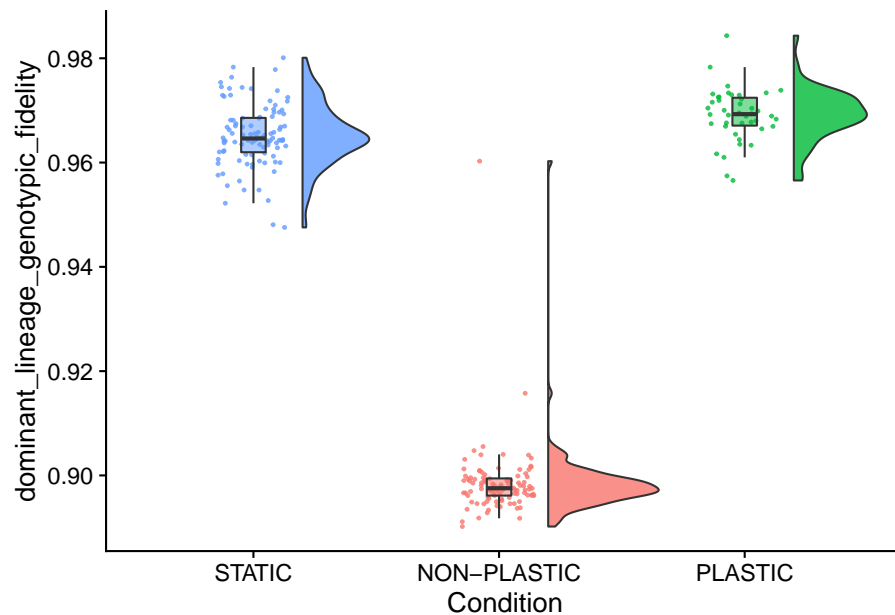
```
summary_data$dominant_lineage_genotypic_fidelity <- (summary_data$dominant_generation_born - summary_data$dominant_generation_parent) / summary_data$dominant_generation_born

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

```



```

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

```

```
## [1] "PLASTIC: 0.969286906891951"
```

3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁴³

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

## [1] "STATIC: 0.964620594632577"

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

## [1] "NON-PLASTIC: 0.89754902563783"

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",
)

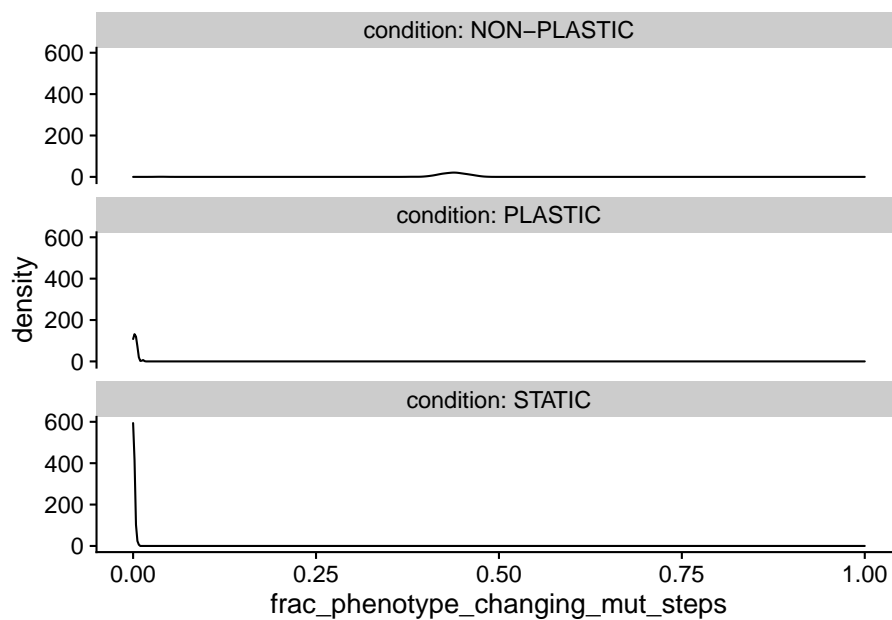
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: summary_data$dominant_lineage_genotypic_fidelity and summary_data$condition
##
##      NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      2e-04
##
## P value adjustment method: bonferroni
```

3.8.4 Characterizing variation along lineages

3.8.4.1 How many mutation-steps along the lineage result in phenotypic changes?

```
summary_data$frac_phenotype_changing_mut_steps <- summary_data$dominant_lineage_num_mut_steps / summary_data$dominant_lineage_num_steps

ggplot(filter(summary_data, dominant_lineage_num_mut_steps > 0), aes(x=frac_phenotype_changing_mut_steps)) +
  geom_density() +
  facet_wrap(
    ~condition,
    nrow=3,
    labeller=label_both
  ) +
  xlim(0, 1.0) +
  theme(
    legend.position="none"
  )
```

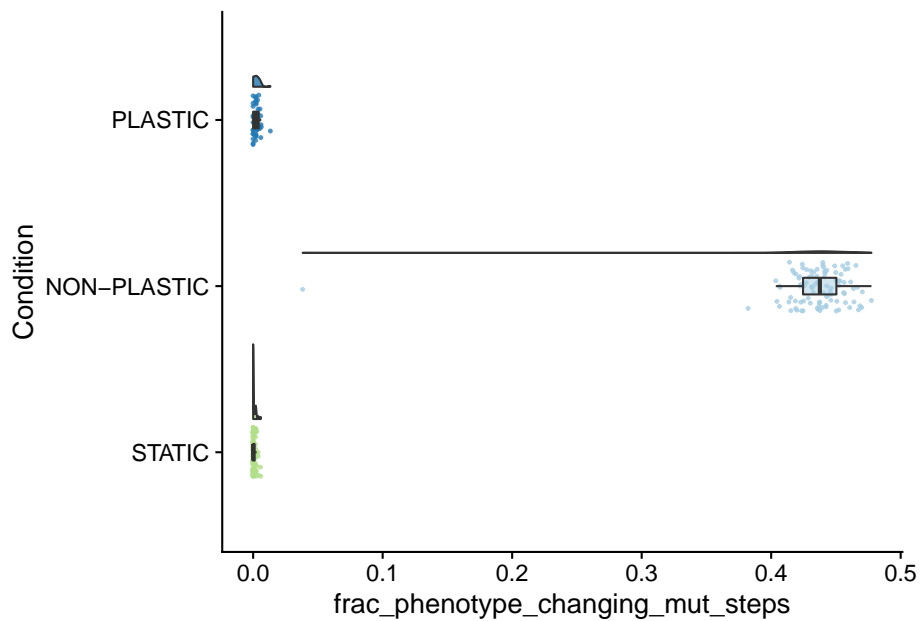


```
ggplot(summary_data, aes(x=condition, y=frac_phenotype_changing_mut_steps, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
```

3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁴⁵

```
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"
) +
ggsave(paste0(working_directory, "plots/", "frac_phenotype_changing_mutational_steps.png"))
```

Saving 6.5 x 4.5 in image



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

```
## [1] "PLASTIC: 0.00224941742616098"
```

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

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

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

```
## [1] "NON-PLASTIC: 0.437583018324547"
```

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

```
##
```

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

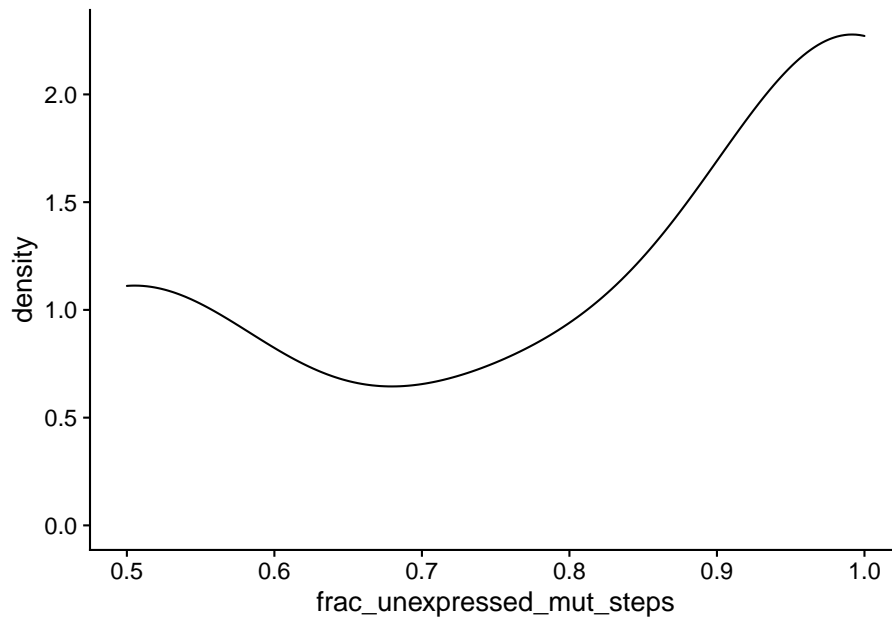
3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁴⁷

```
##
## 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
```

```
summary_data$frac_unexpressed_mut_steps <- summary_data$dominant_lineage_num_mut_steps_that_change_aggregately
summary_data$frac_expressed_mut_steps <- summary_data$dominant_lineage_num_mut_steps_that_change_aggregately

ggplot(filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change_aggregately > 0)) +
  geom_density() +
  theme(
    legend.position="none"
  )
```



```
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_1)
print(bo)
```

```
##
```

```
## ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
##
```

```
##
```

```
## Call:
```

```
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_steps_1)
##       0)$frac_unexpressed_mut_steps, statistic = samplemean, R = 10000)
```

```
##
```

```
##
```

```
## Bootstrap Statistics :
```

```
##      original      bias    std. error
```

```
## t1* 0.8247126 0.0003695402 0.03959875
```

```
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")
```


3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁴⁹

```
##
## Intervals :
## Level      Percentile
## 95%      ( 0.7471,  0.9023 )
## 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$dominant_lineage_num_mut_steps_that_change_unexpressed_phenotype)

## [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)
```

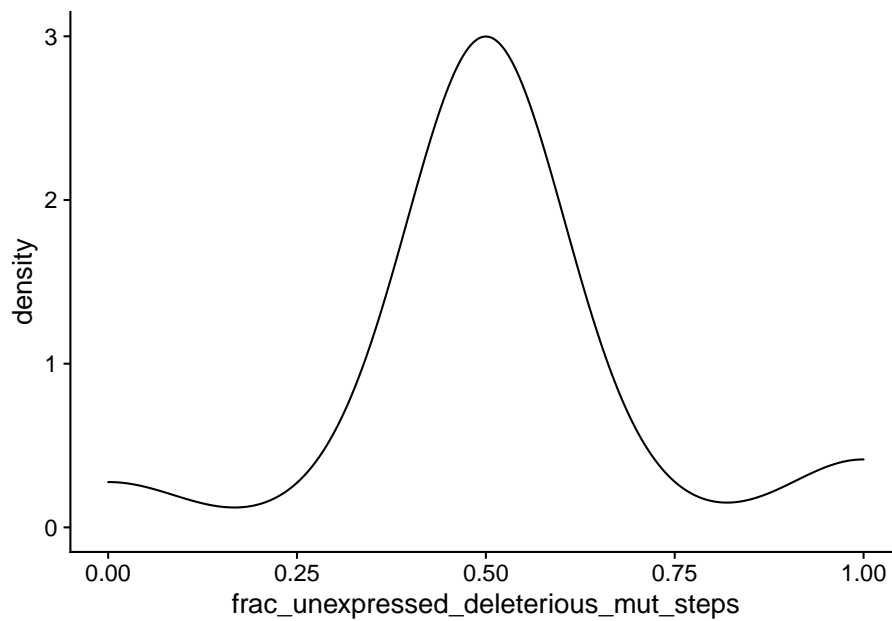
3.8.4.3 For PLASTIC populations, what fraction of mutations that affect the unexpressed phenotype are deleterious versus beneficial?

```
aggregate_frac_unexpressed_deleterious_mut_steps <- sum(plastic_summary_data$dominant_lineage_num_mut_steps_that_change_unexpressed_phenotype)
aggregate_frac_unexpressed_beneficial_mut_steps <- sum(plastic_summary_data$dominant_lineage_num_mut_steps_that_change_unexpressed_phenotype)
```

Deleterious

```
summary_data$frac_unexpressed_deleterious_mut_steps <- summary_data$dominant_lineage_num_mut_steps_that_change_unexpressed_phenotype / summary_data$dominant_lineage_num_mut_steps_that_change_aggregate_phenotype

ggplot(
  filter(summary_data, condition=="PLASTIC" & dominant_lineage_num_mut_steps_that_change_unexpressed_phenotype > 0)
  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_1)
print(bo)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_steps_1)
##       0)$frac_unexpressed_deleterious_mut_steps, statistic = samplemean,
##       R = 10000)
```

```
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.5172414 -0.0002850575 0.03952691
```

```
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 :
```

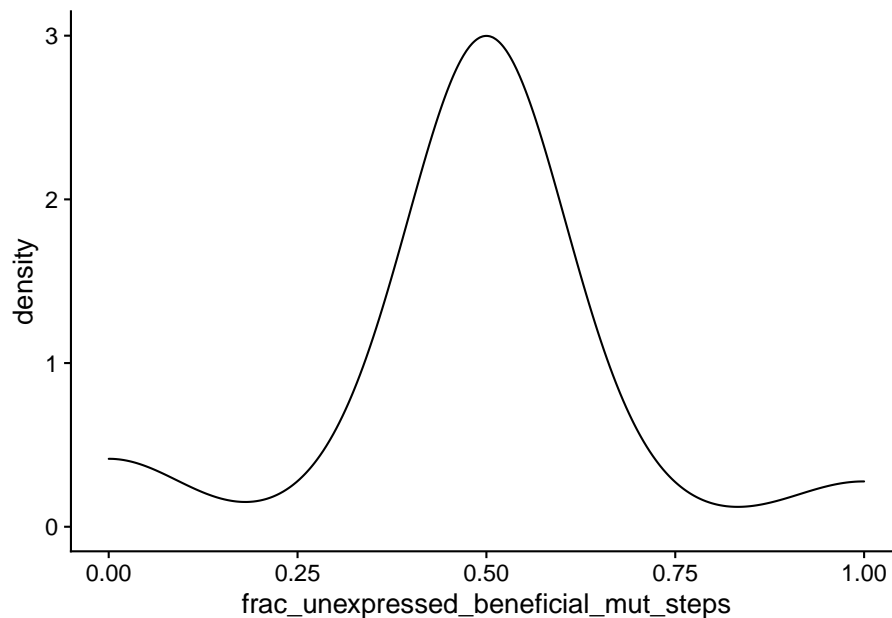
3.8. MUTATION ACCUMULATION ALONG THE DOMINANT LINEAGE⁵¹

```
## Level      Percentile
## 95%      ( 0.4391,  0.5954 )
## Calculations and Intervals on Original Scale
```

Beneficial

```
summary_data$frac_unexpressed_beneficial_mut_steps <- summary_data$dominant_lineage_num_mut_steps
```

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



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

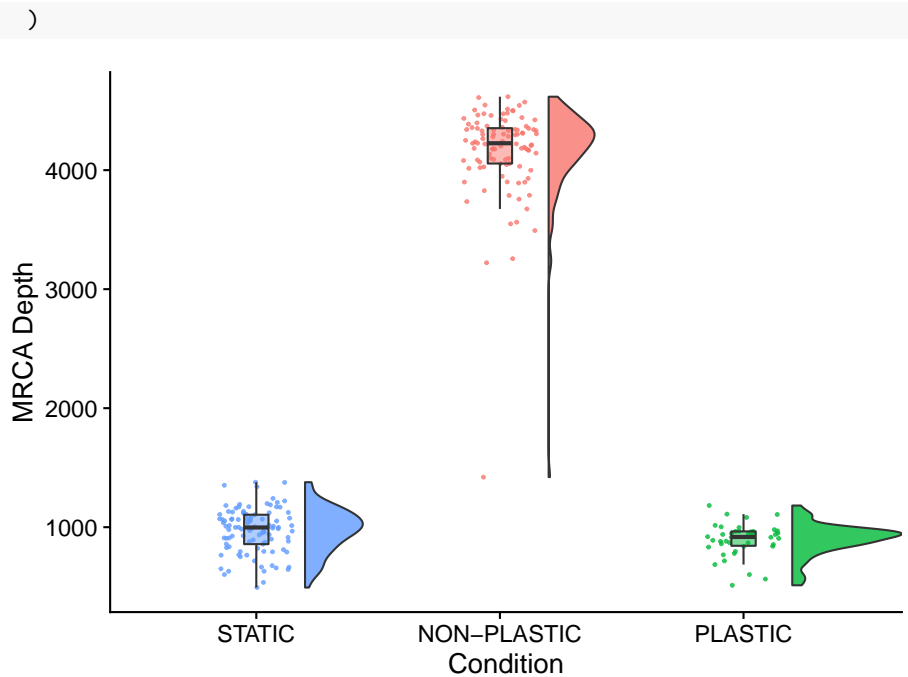
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = filter(summary_data, condition == "PLASTIC" & dominant_lineage_num_mut_steps_that_change_unexpressed),
##       nboot = 1000, R = 1000, statistic = samplemean,
```

```
##      R = 10000)
##
##
## Bootstrap Statistics :
##      original      bias      std. error
## t1* 0.4827586 0.0003794253 0.03928081
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%      ( 0.4034, 0.5575 )
## Calculations and Intervals on Original Scale
```

3.9 Depth of MRCA

```
ggplot(summary_data, aes(x=condition, y=phylo_mrca_depth, 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("MRCA Depth") +
  theme(
    legend.position="none"
```



3.10 Manuscript figures

Figures styled for the paper.

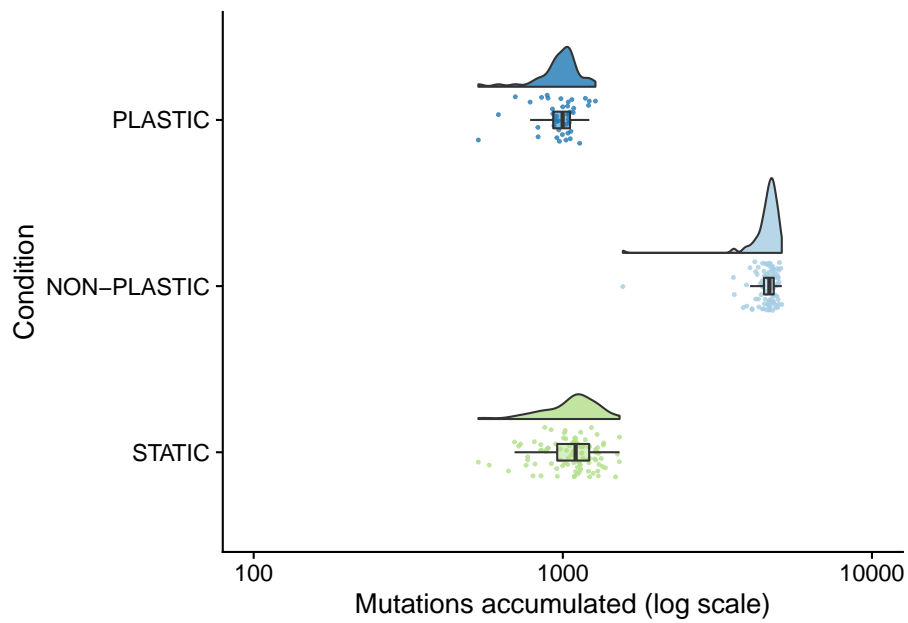
3.10.1 Total evolutionary change panel

Selective sweeps, mutation accumulation, phenotypic volatility.

Mutation accumulation:

```
# dominant_lineage_total_mut_cnt or mutations_per_lineage_step?
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="Mutations accumulated (log scale)",
    trans="log10",
    breaks=c(100, 1000, 10000),
    limits=c(100, 10000)
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "mutation-accumulation.pdf"),
    width=5,
    height=4
  )
mutation_count_fig
```



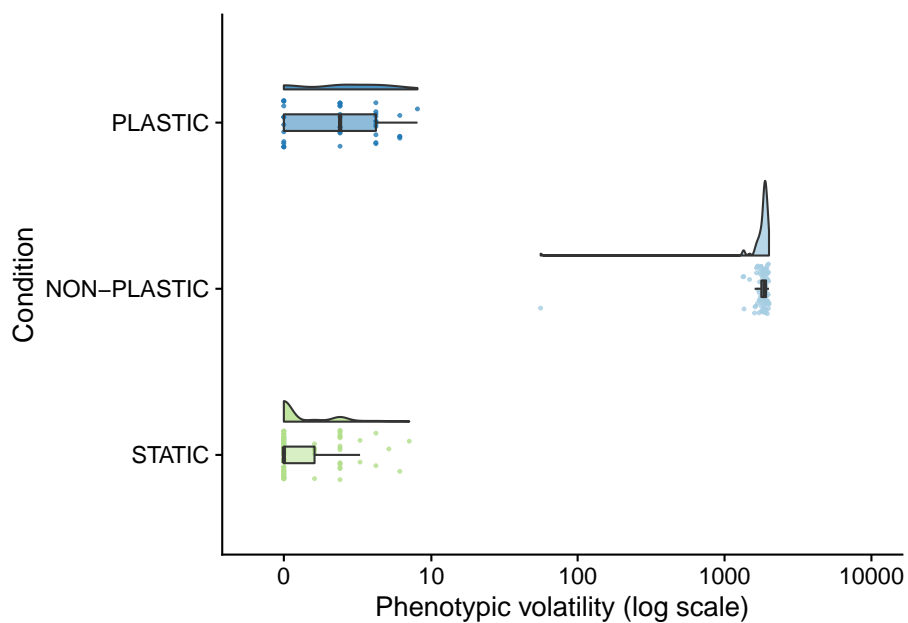
Phenotypic volatility:

```
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
  )
```

```

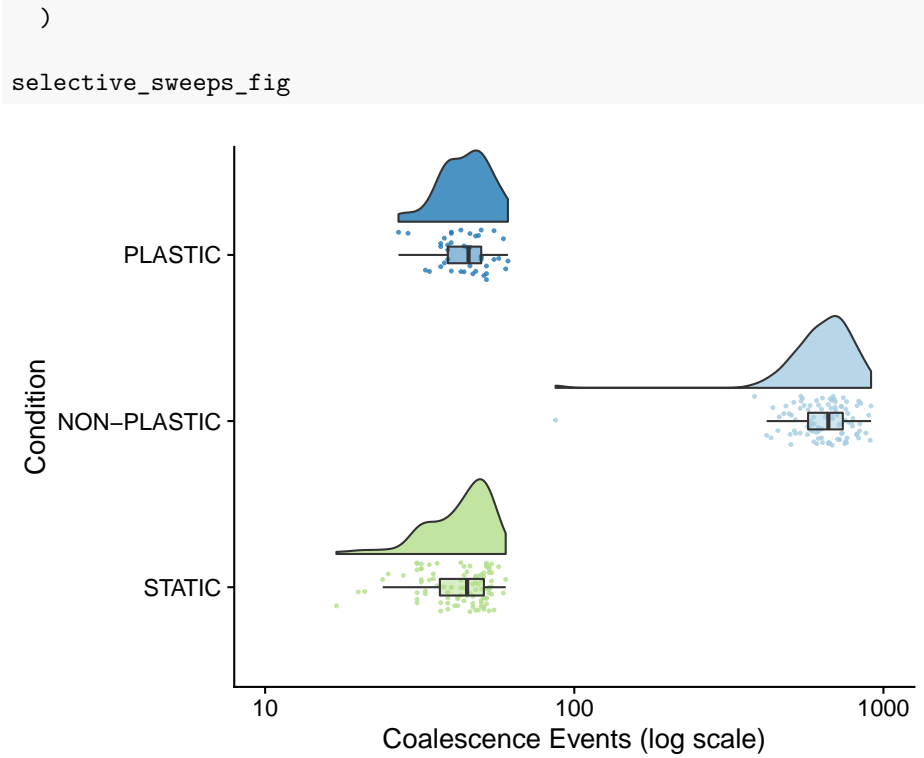
) +
scale_y_continuous(
  name="Phenotypic volatility (log scale)",
  trans="pseudo_log",
  breaks=c(0, 10, 100, 1000, 10000),
  limits=c(-1,10000)
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
coord_flip() +
theme(
  legend.position="none"
) +
ggsave(
  paste0(working_directory, "plots/", "phenotypic-volatility.pdf"),
  width=4,
  height=4
)
phenotypic_volatility_fig

```



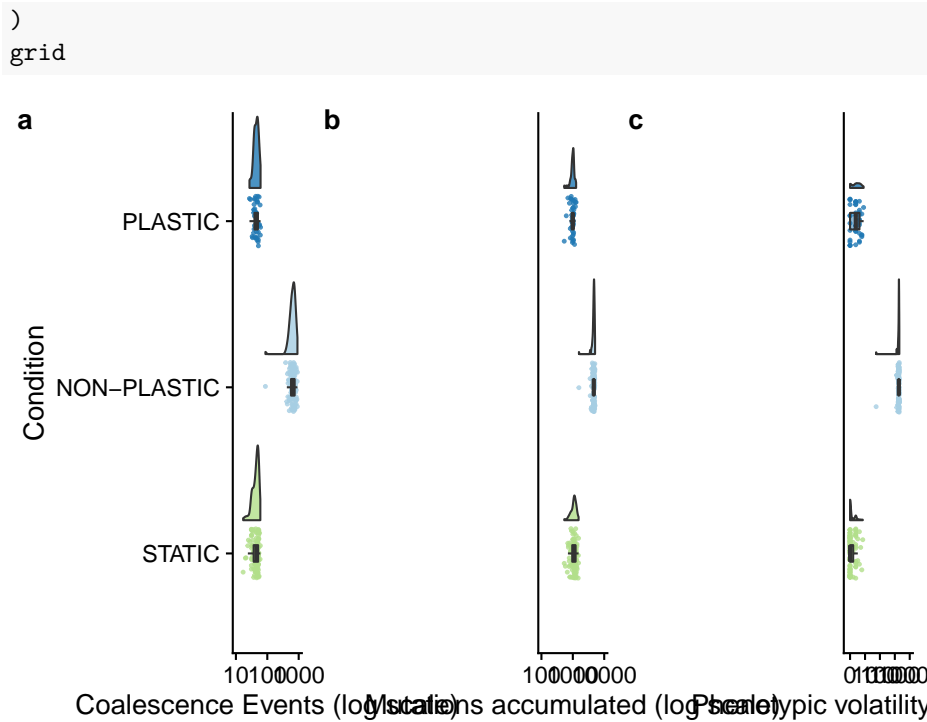
Selective sweeps:

```
selective_sweeps_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
  ) +
  scale_y_continuous(
    name="Coalescence Events (log scale)",
    trans="log10",
    breaks=c(10, 100, 1000),
    limits=c(10, 1000)
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "selective-sweeps.pdf"),
    width=4,
    height=4
  )
```



All together:

```
grid <- plot_grid(
  selective_sweeps_fig + theme(
    legend.position="none"
  ),
  mutation_count_fig + theme(
    legend.position="none",
    axis.ticks.y=element_blank(),
    axis.text.y=element_blank(),
    axis.title.y=element_blank()
  ),
  phenotypic_volatility_fig + theme(
    legend.position="none",
    axis.ticks.y=element_blank(),
    axis.text.y=element_blank(),
    axis.title.y=element_blank()
  ),
  nrow=1,
  align="v",
  labels="auto"
```

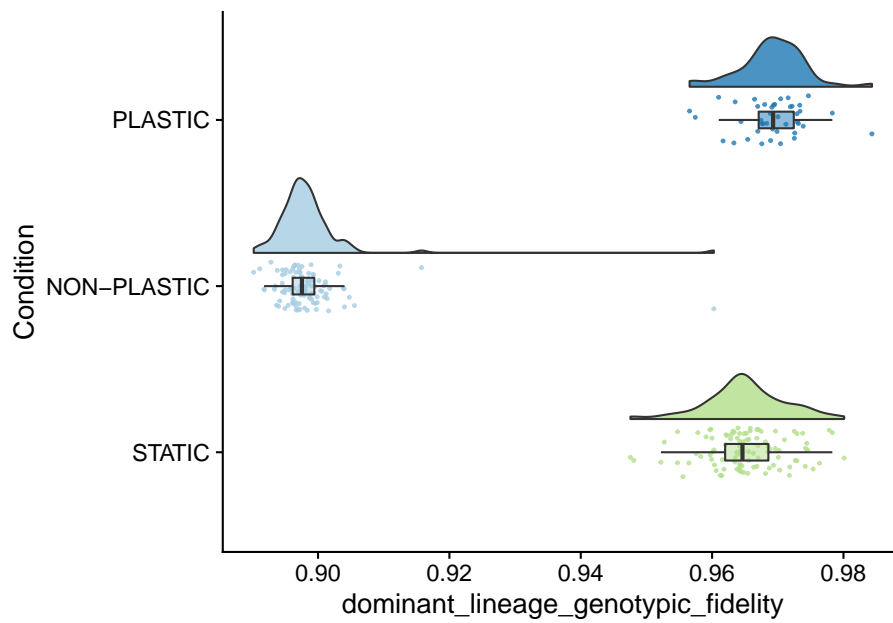


```
save_plot(
  paste0(working_directory, "plots/", "evolutionary-dynamics.pdf"),
  grid,
  base_height=6,
  base_asp=2.5
)
save_plot(
  paste0(working_directory, "plots/", "evolutionary-dynamics.png"),
  grid,
  base_height=6,
  base_asp=2.5
)
```

3.10.2 Rate of evolutionary change panel

```
# dominant_lineage_total_mut_cnt or mutations_per_lineage_step?
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_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/", "genotypic-fidelity.png"),
    width=5,
    height=4
  )
genotypic_fidelity_fig
```



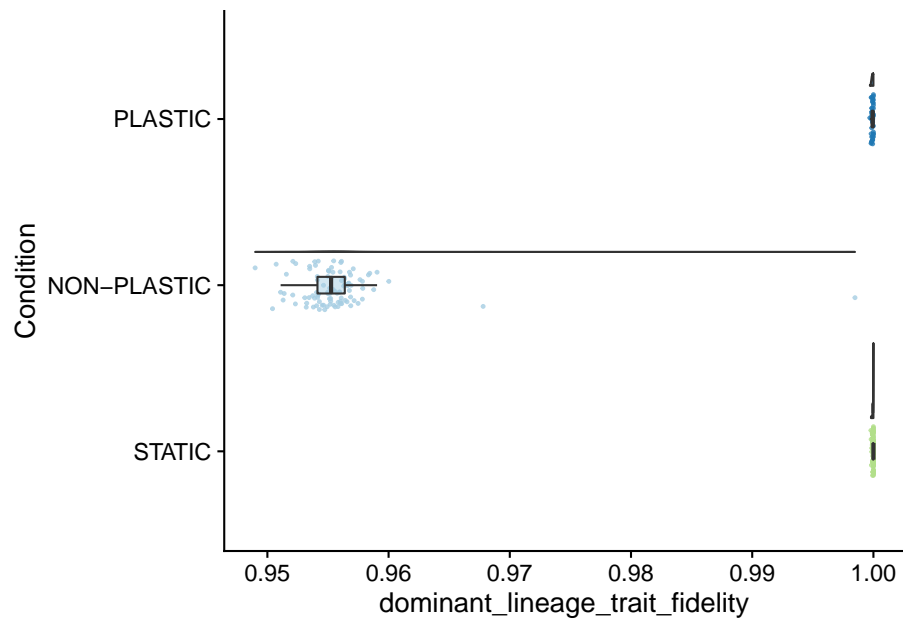
Phenotypic fidelity:

```
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_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
coord_flip() +
theme(
  legend.position="none"
) +
ggsave(
  paste0(working_directory, "plots/", "phenotypic-fidelity.png"),
  width=4,
  height=4
)
phenotypic_fidelity_fig

```



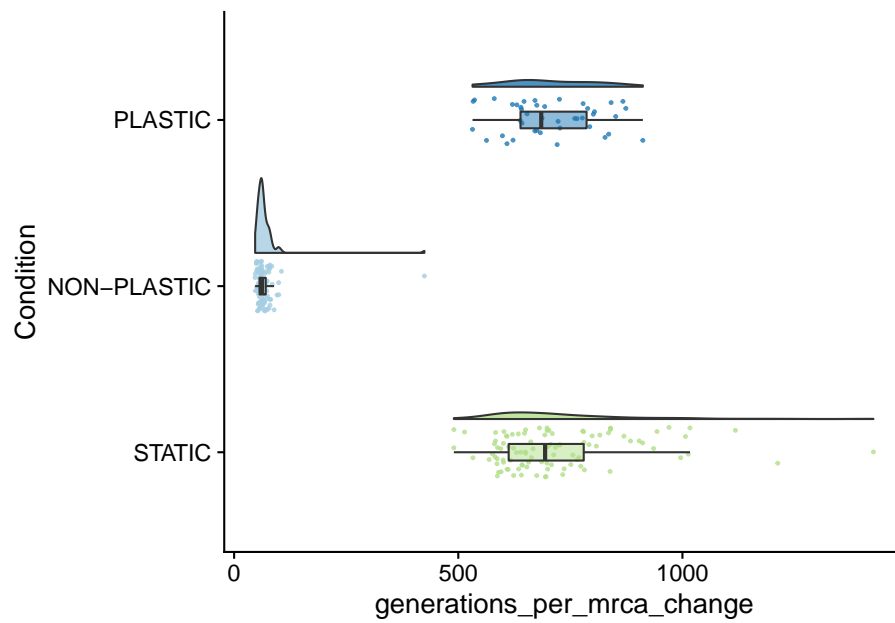
rate of selective sweeps:

```

selective_sweeps_rate_fig <- ggplot(
  summary_data,
  aes(x=condition, y=generations_per_mrca_change, fill=condition)
) +

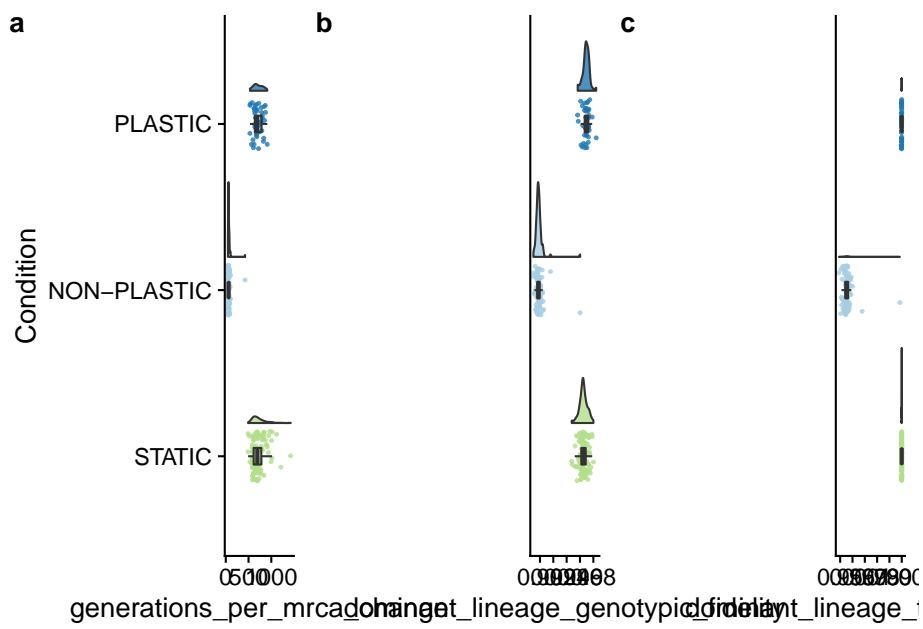
```

```
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_fill_brewer(  
  palette="Paired"  
) +  
scale_color_brewer(  
  palette="Paired"  
) +  
coord_flip() +  
theme(  
  legend.position="none"  
) +  
ggsave(  
  paste0(working_directory, "plots/", "generations-between-selective-sweeps.png"),  
  width=4,  
  height=4  
)  
  
selective_sweeps_rate_fig
```



All together:

```
grid <- plot_grid(
  selective_sweeps_rate_fig + theme(
    legend.position="none"
  ),
  genotypic_fidelity_fig + theme(
    legend.position="none",
    axis.ticks.y=element_blank(),
    axis.text.y=element_blank(),
    axis.title.y=element_blank()
  ),
  phenotypic_fidelity_fig + theme(
    legend.position="none",
    axis.ticks.y=element_blank(),
    axis.text.y=element_blank(),
    axis.title.y=element_blank()
  ),
  nrow=1,
  align="v",
  labels="auto"
)
grid
```

```

save_plot(
  paste0(working_directory, "plots/", "rate-evolutionary-dynamics.png"),
  grid,
  base_height=6,
  base_asp=2.5
)

```


Chapter 4

Evolution and maintenance of novel traits

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

4.1 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")
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(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      _
## arch          x86_64-pc-linux-gnu
## 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

```
##### 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"
)

##### 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_REACTION_SENSORS == "True")
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_task_value)

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())

```

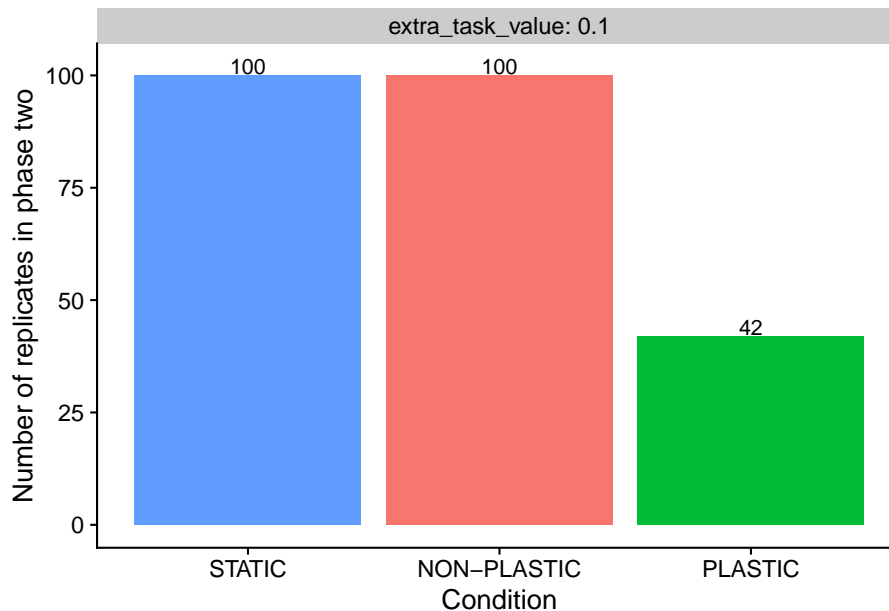
```
dir.create(paste0(working_directory, "plots"), showWarnings=FALSE)
```

4.4 Evolution of phenotypic plasticity

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

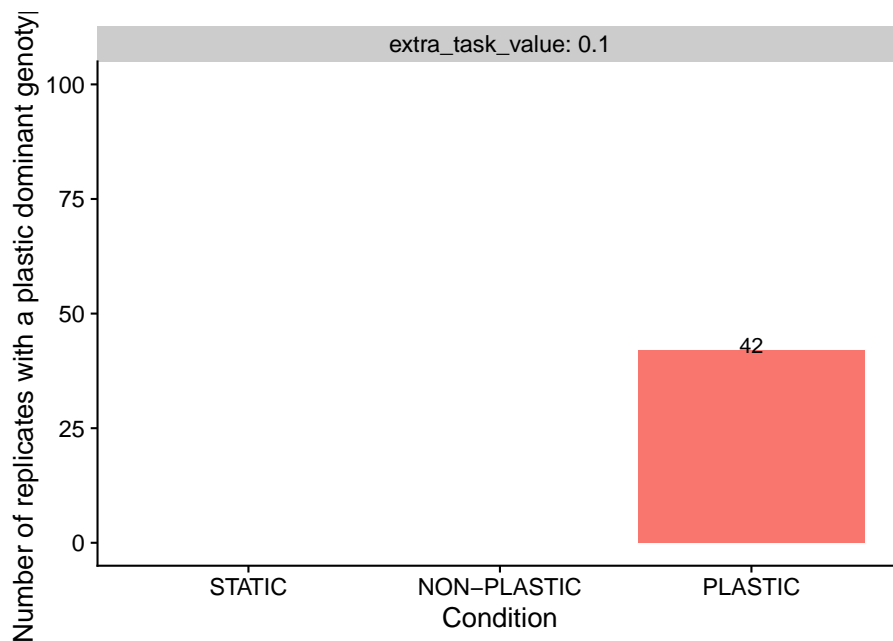
```
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition, extra_task_value)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
```

```
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
  ) +
  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_value)
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
  ) +
  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"
  )
)
```



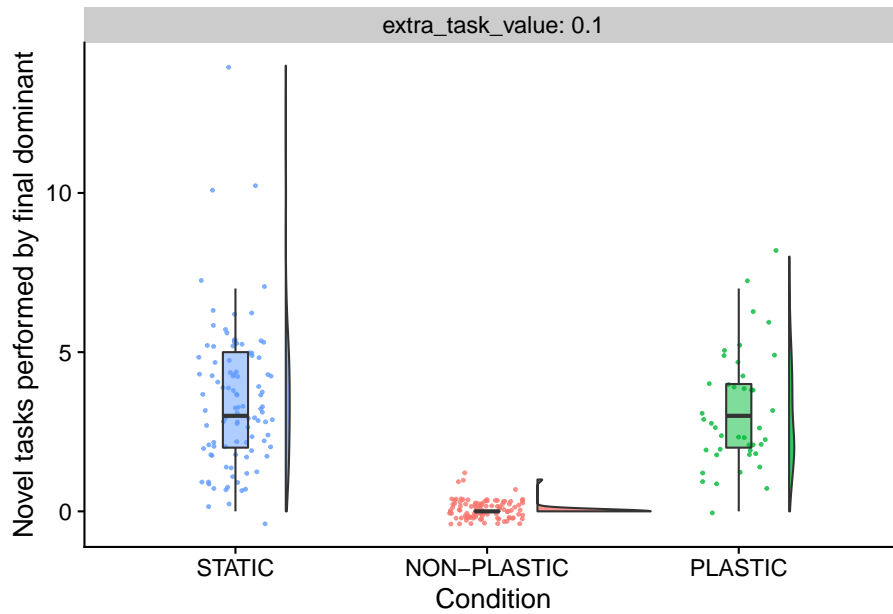
4.5 Final dominant novel task performance

How many novel tasks do final dominant genotypes perform?

```
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  
) +  
ylab("Novel tasks performed by final dominant") +  
facet_wrap(  
  ~extra_task_value,  
  labeller=label_both  
) +  
theme(  
  legend.position="none"  
) +  
ggsave(  
  paste0(working_directory, "plots/dominant-extra-tasks.pdf"),  
  width=15,  
  height=10  
)
```



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

## [1] "PLASTIC median: 3"

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

## [1] "STATIC median: 3"

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

## [1] "NON-PLASTIC median: 0"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_extra_tasks~condition,
  data=dom_task_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=dom_task_data$dominant_extra_tasks,
  g=dom_task_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)

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

4.6 Final population novel task performance

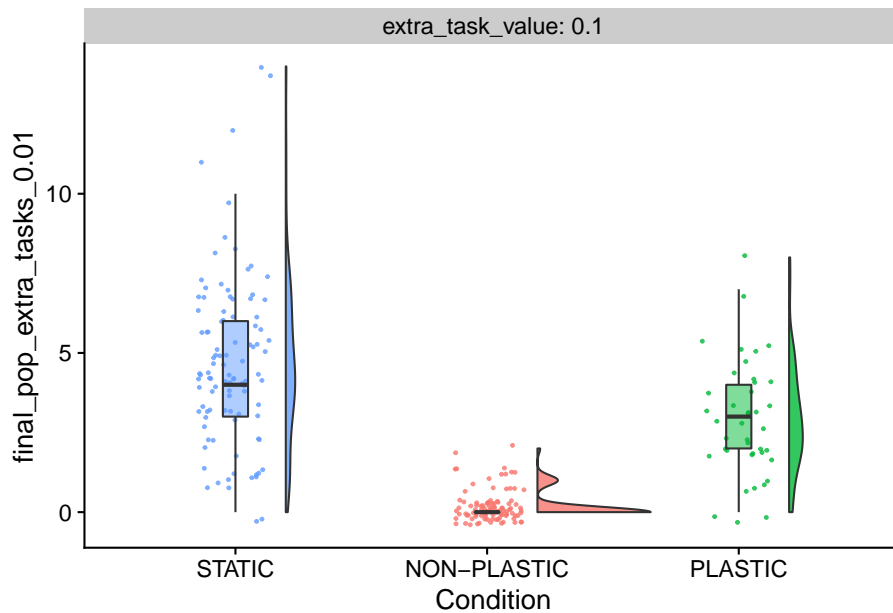
How many novel tasks are performed across the final population (1% of organisms must perform to count)?

```
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
) +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
)

```



```

paste0(
  "PLASTIC median: ",
  median(filter(summary_data, condition=="PLASTIC")$final_pop_extra_tasks_0.01)
)

```

```
## [1] "PLASTIC median: 3"
```

```

paste0(
  "STATIC median: ",
  median(filter(summary_data, condition=="STATIC")$final_pop_extra_tasks_0.01)
)

```

```
## [1] "STATIC median: 4"
```

```

paste0(
  "NON-PLASTIC median: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$final_pop_extra_tasks_0.01)
)

## [1] "NON-PLASTIC median: 0"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=final_pop_extra_tasks_0.01~condition,
  data=dom_task_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=dom_task_data$final_pop_extra_tasks_0.01,
  g=dom_task_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)

##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: dom_task_data$final_pop_extra_tasks_0.01 and dom_task_data$condition
##
##      NON-PLASTIC PLASTIC
## PLASTIC < 2e-16      -
## STATIC  < 2e-16      0.00016
##
## P value adjustment method: bonferroni

```

4.7 Population-level novel tasks discovered

```

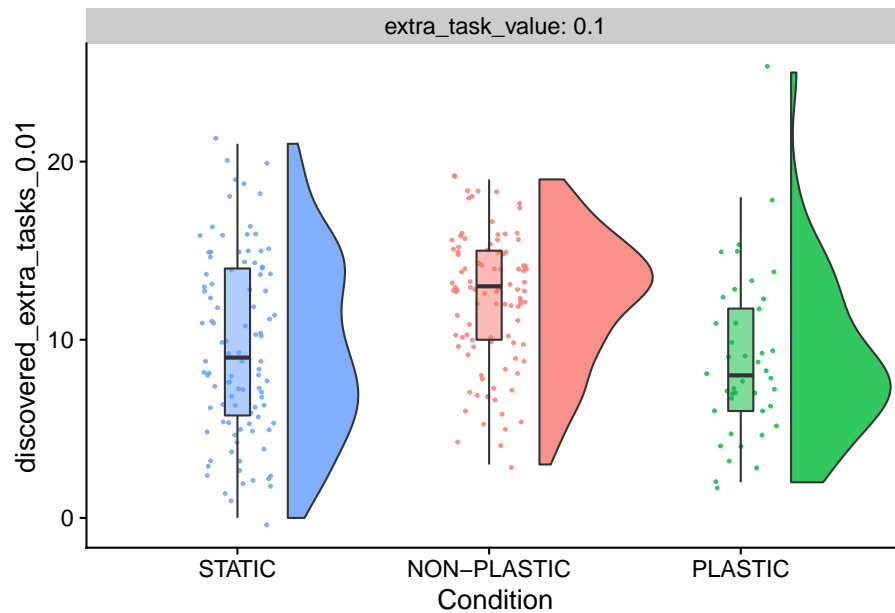
ggplot(summary_data, aes(x=condition, y=discovered_extra_tasks_0.01, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +

```

```

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
) +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
)

```



```

paste0(
  "PLASTIC median: ",

```

```

    median(filter(summary_data, condition=="PLASTIC")$discovered_extra_tasks_0.01)
  )

## [1] "PLASTIC median: 8"
paste0(
  "STATIC median: ",
  median(filter(summary_data, condition=="STATIC")$discovered_extra_tasks_0.01)
)

## [1] "STATIC median: 9"
paste0(
  "NON-PLASTIC median: ",
  median(filter(summary_data, condition=="NON-PLASTIC")$discovered_extra_tasks_0.01)
)

## [1] "NON-PLASTIC median: 13"
reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=discovered_extra_tasks_0.01~condition,
  data=dom_task_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=dom_task_data$discovered_extra_tasks_0.01,
  g=dom_task_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)

##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: dom_task_data$discovered_extra_tasks_0.01 and dom_task_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC 2.4e-05      -
## STATIC  0.00035     1.00000
##

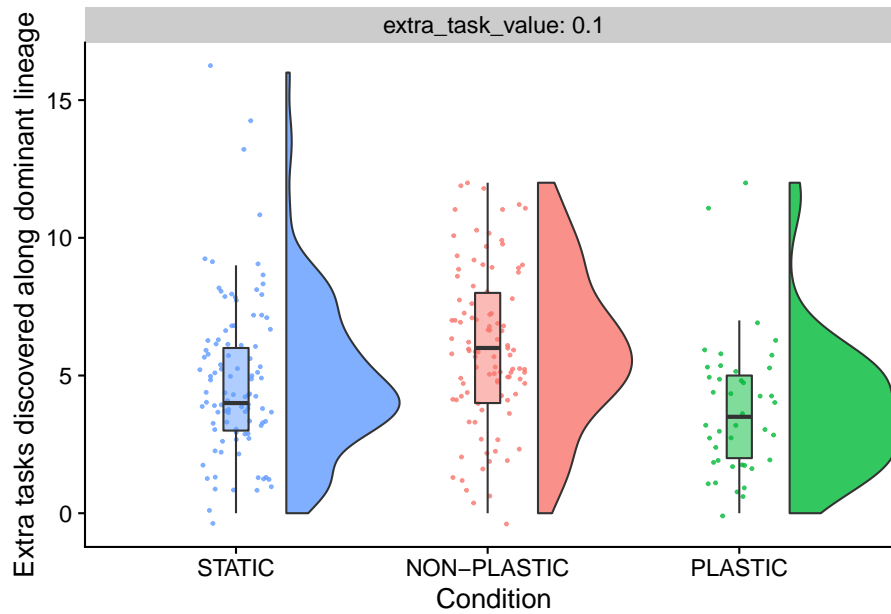
```

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

4.8 Novel tasks along lineage of final dominant genotype

```
ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_discovered, 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
  ) +
  ylab("Extra tasks discovered along dominant lineage") +
  facet_wrap(
    ~extra_task_value,
    labeller=label_both
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-extra-tasks-discovered.pdf"),
    width=15,
    height=10
  )
)
```


4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE81



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

## [1] "PLASTIC median: 3.5"

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

## [1] "STATIC median: 4"

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

## [1] "NON-PLASTIC median: 6"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_discovered~condition,
  data=dom_task_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=dom_task_data$dominant_lineage_extra_traits_discovered,
  g=dom_task_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)

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

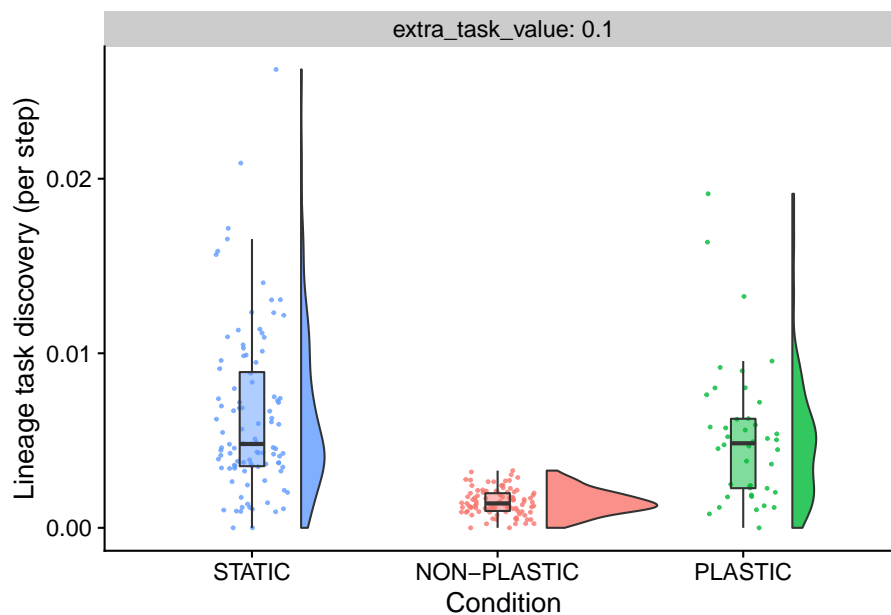
4.8.1.1 Novel traits discovered per step

This isn't totally fair to non-plastic lineages because they're continuously re-adapting, so they have more genotypes along lineage.

```
summary_data$dominant_lineage_extra_traits_discovered_per_step <- summary_data$dominant_lineage_extra_traits_discovered_per_step
ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_discovered_per_step)) +
  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
  ) +
```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE83

```
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
ylab("Lineage task discovery (per step)" ) +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
)
```



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

```
## [1] "PLASTIC median: 0.00484428434398198"
```

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

```
## [1] "STATIC median: 0.00480194844967106"
```

```

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

## [1] "NON-PLASTIC median: 0.00139827576402932"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_discovered_per_step~condition,
  data=dom_task_data
)

##
##  Kruskal-Wallis rank sum test
##
## data:  dominant_lineage_extra_traits_discovered_per_step by condition
## Kruskal-Wallis chi-squared = 106.72, df = 2, p-value < 2.2e-16

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

##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  dom_task_data$dominant_lineage_extra_traits_discovered_per_step and dom_task_
##
##           NON-PLASTIC PLASTIC
## PLASTIC 9.7e-11      -
## STATIC  < 2e-16      0.67
##
## P value adjustment method: bonferroni

```

4.8.2 Novel tasks discovered per generation

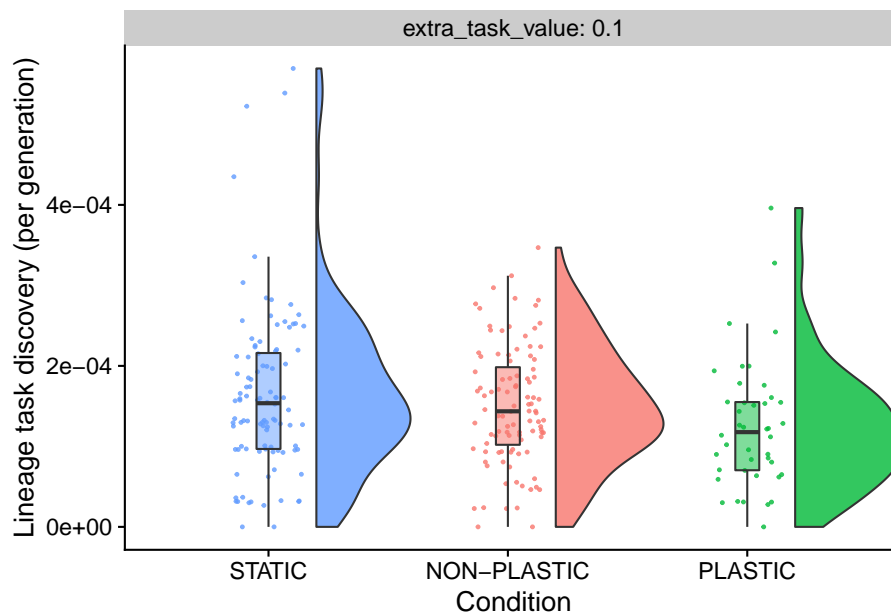
```

summary_data$dominant_lineage_extra_traits_discovered_per_generation <- summary_data$d
ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_discovered_per_g
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +

```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE85

```
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("Lineage task discovery (per generation)") +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
)
)
```



```

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

## [1] "PLASTIC median: 0.000117695011124939"

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

## [1] "STATIC median: 0.00015363220504867"

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

## [1] "NON-PLASTIC median: 0.00014358046266055"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_discovered_per_generation~condition,
  data=dom_task_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

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

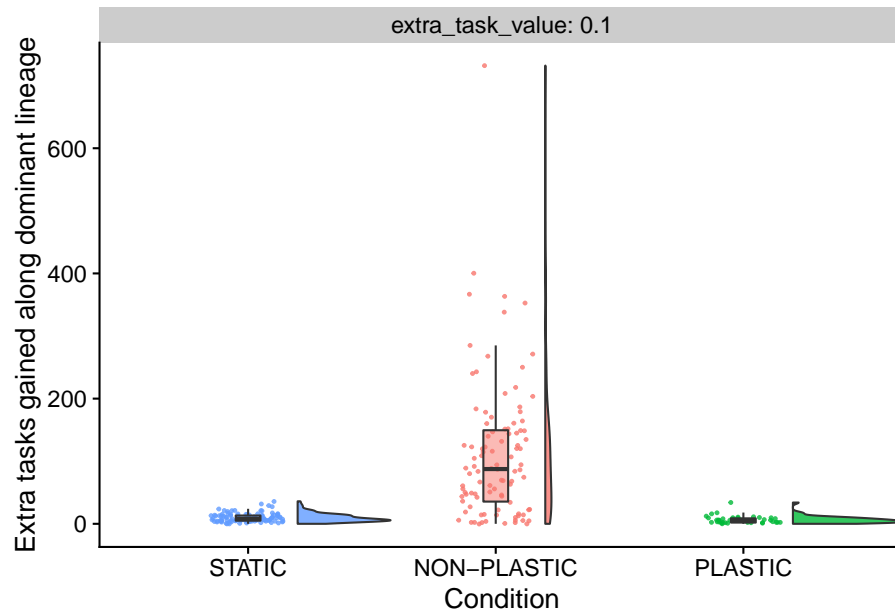
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: dom_task_data$dominant_lineage_extra_traits_discovered_per_generation and dom_task_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC 0.092          -

```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE⁸⁷

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

```
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("Extra tasks gained along dominant lineage") +
  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
  )
```

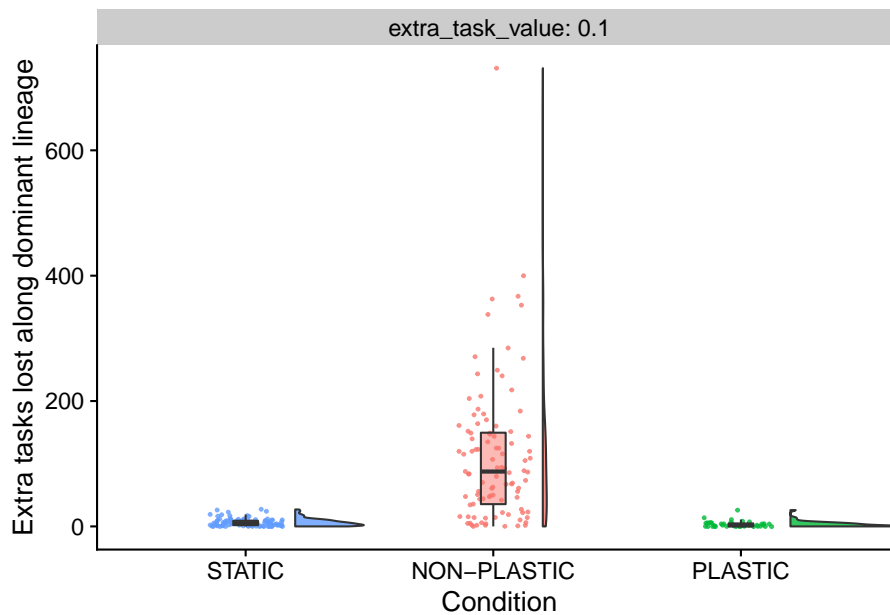


4.8.4 Novel tasks lost

```
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
  ) +
  ylab("Extra tasks lost along dominant lineage") +
  facet_wrap(
```


4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE89

```
~extra_task_value,  
  labeller=label_both  
) +  
theme(  
  legend.position="none"  
) +  
ggsave(  
  paste0(working_directory, "plots/dominant-lineage-extra-tasks-lost.pdf"),  
  width=15,  
  height=10  
)
```



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

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

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

```
## [1] "STATIC median: 5"
```

```

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

## [1] "NON-PLASTIC median: 87.5"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_lost~condition,
  data=dom_task_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=dom_task_data$dominant_lineage_extra_traits_lost,
  g=dom_task_data$condition,
  p.adjust.method="bonferroni",
  conf.int=TRUE,
  conf.level=0.95
)

##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  dom_task_data$dominant_lineage_extra_traits_lost and dom_task_data$condition
##
##           NON-PLASTIC PLASTIC
## PLASTIC 2.7e-16      -
## STATIC  < 2e-16      0.0024
##
## P value adjustment method: bonferroni

```

4.8.4.1 Novel traits lost per step

Again, not totally fair to non-plastic lineages.

```

summary_data$dominant_lineage_extra_traits_lost_per_step <- summary_data$dominant_lineage_extra_traits_lost /
  ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_lost_per_step, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  )

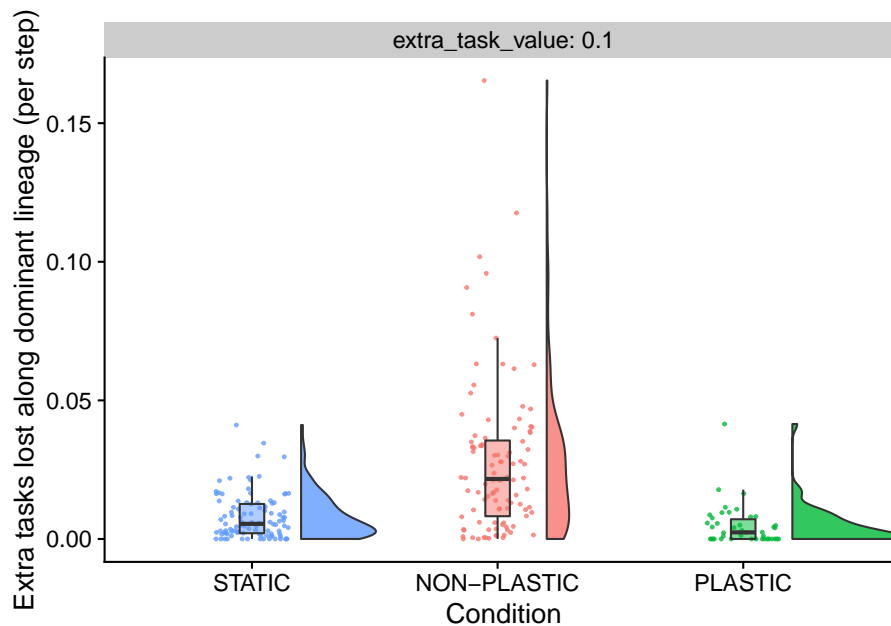
```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE91

```

) +
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("Extra tasks lost along dominant lineage (per step)" ) +
facet_wrap(
  ~extra_task_value,
  labeller=label_both
) +
theme(
  legend.position="none"
)

```



```

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

## [1] "PLASTIC median: 0.00238455242036334"

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

## [1] "STATIC median: 0.00544747485837901"

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

## [1] "NON-PLASTIC median: 0.0216427755153431"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_lost_per_step~condition,
  data=dom_task_data
)

##
## Kruskal-Wallis rank sum test
##
## data: dominant_lineage_extra_traits_lost_per_step by condition
## Kruskal-Wallis chi-squared = 65.779, df = 2, p-value = 5.204e-15

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

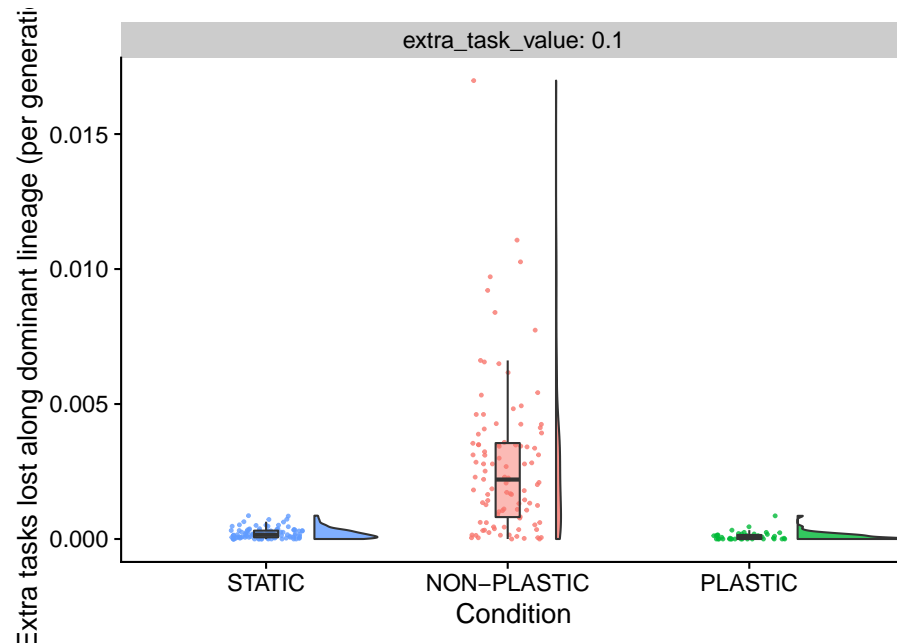
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: dom_task_data$dominant_lineage_extra_traits_lost_per_step and dom_task_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC 1.3e-10      -

```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE⁹³

```
## STATIC 1.7e-10 0.0092
##
## P value adjustment method: bonferroni
```

```
summary_data$dominant_lineage_extra_traits_lost_per_generation <- summary_data$dominant_lineage_e
ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_lost_per_generation, 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
  ) +
  ylab("Extra tasks lost along dominant lineage (per generation)") +
  facet_wrap(
    ~extra_task_value,
    labeller=label_both
  ) +
  theme(
    legend.position="none"
  )
)
```



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

## [1] "PLASTIC median: 6.25141973661864e-05"

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

## [1] "STATIC median: 0.000161396283669756"

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

## [1] "NON-PLASTIC median: 0.0022026054610079"

reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_lineage_extra_traits_lost_per_generation~condition,
  data=dom_task_data
)
```

4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE⁹⁵

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

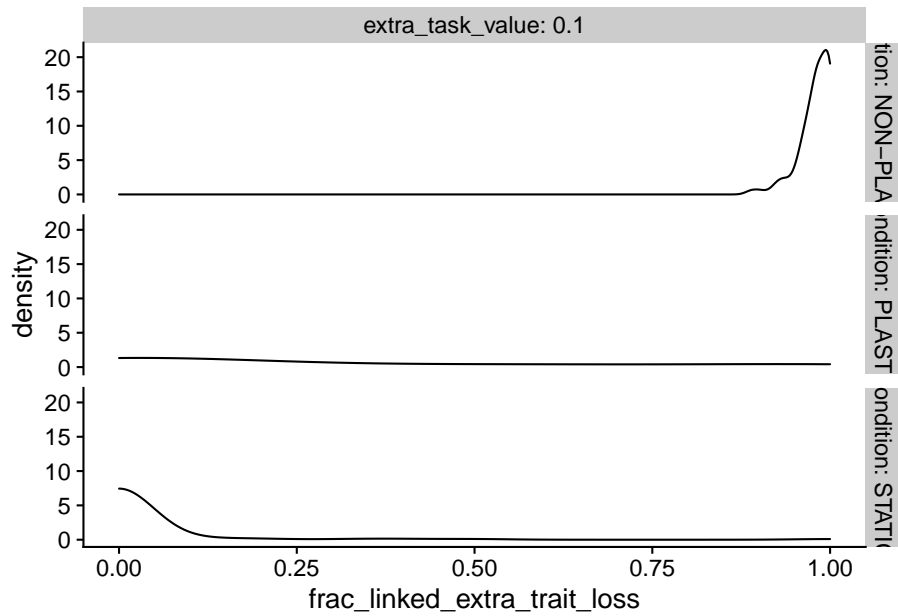
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: dom_task_data$dominant_lineage_extra_traits_lost_per_generation and dom_task_data$condi
##
##          NON-PLASTIC PLASTIC
## PLASTIC 1.1e-15      -
## STATIC  < 2e-16     0.0012
##
## P value adjustment method: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

4.8.4.3 How many instances of novel trait loss co-occur with changes in base phenotype?

Task loss linked with primary trait changes.

```
lost_traits_summary_data <- filter(summary_data, extra_task_value==0.1 & dominant_lineage_extra_t
lost_traits_summary_data$frac_linked_extra_trait_loss <- lost_traits_summary_data$dominant_lineag

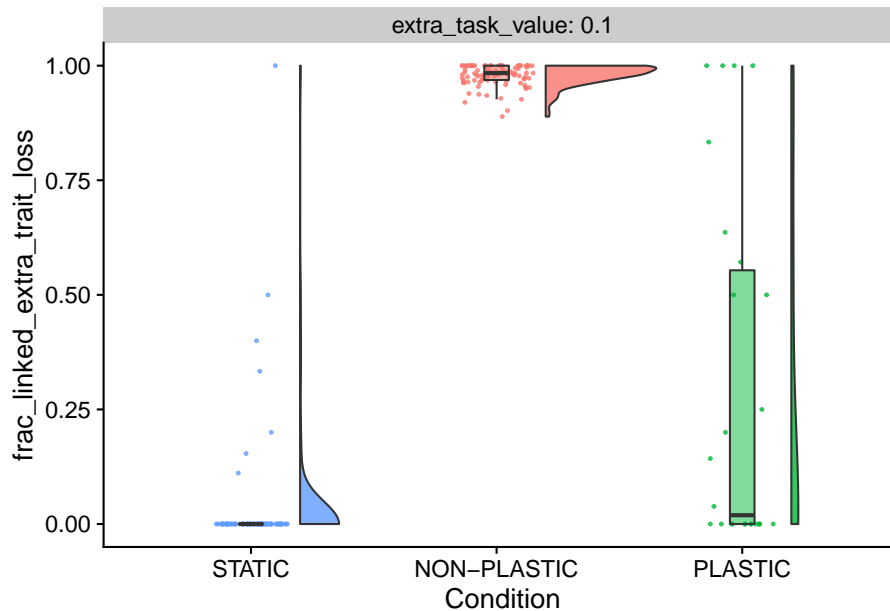
ggplot(lost_traits_summary_data, aes(x=frac_linked_extra_trait_loss)) +
  geom_density() +
  facet_grid(
    condition~extra_task_value,
    labeller=label_both
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-extra-tasks-lost-linkage.pdf"),
    width=15,
    height=10
  )
```



```
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
  ) +
  facet_wrap(
    ~extra_task_value,
    labeller=label_both
  ) +
  theme(
```


4.8. NOVEL TASKS ALONG LINEAGE OF FINAL DOMINANT GENOTYPE97

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



```
paste0(
  "PLASTIC median: ",
  median(filter(loss_traits_summary_data, condition=="PLASTIC")$frac_linked_extra_trait_loss)
)
```

```
## [1] "PLASTIC median: 0.0192307692307692"
```

```
paste0(
  "STATIC median: ",
  median(filter(loss_traits_summary_data, condition=="STATIC")$frac_linked_extra_trait_loss)
)
```

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

```
paste0(
  "NON-PLASTIC median: ",
  median(filter(loss_traits_summary_data, condition=="NON-PLASTIC")$frac_linked_extra_trait_loss)
)
```

```
## [1] "NON-PLASTIC median: 0.983803278688525"
```

```
kruskal.test(
  formula=frac_linked_extra_trait_loss~condition,
  data=loss_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$frac_linked_extra_trait_loss and lost_traits_summary_data$condition
##
##           NON-PLASTIC PLASTIC
## PLASTIC  1.9e-08      -
## STATIC   < 2e-16      1.8e-06
##
## P value adjustment method: bonferroni
sum(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_trait_loss)
## [1] 10998
sum(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_trait_loss)
## [1] 11229
aggregate_frac_linked_extra_trait_loss_nonplastic <- sum(filter(lost_traits_summary_data, condition=="NON-PLASTIC")$dominant_lineage_extra_trait_loss)
aggregate_frac_linked_extra_trait_loss_nonplastic
## [1] 0.9794283
sum(filter(lost_traits_summary_data, condition=="PLASTIC")$dominant_lineage_extra_trait_loss)
## [1] 29
sum(filter(lost_traits_summary_data, condition=="PLASTIC")$dominant_lineage_extra_trait_loss)
## [1] 142
aggregate_frac_linked_extra_trait_loss_plastic <- sum(filter(lost_traits_summary_data, condition=="PLASTIC")$dominant_lineage_extra_trait_loss)
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, condition=="STATIC")$dominant_lineage_extra_traits_lost)
aggregate_frac_linked_extra_trait_loss_nonplastic
```

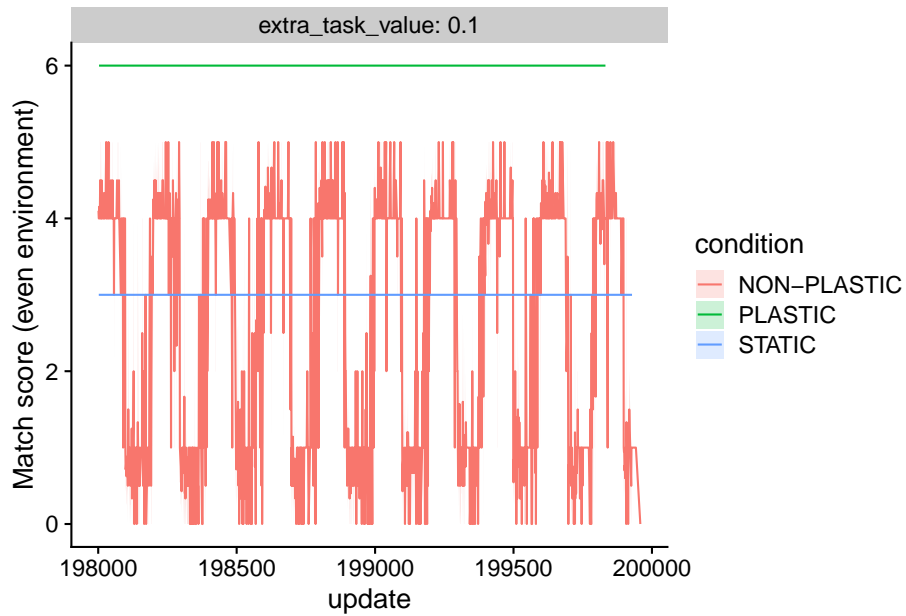
```
## [1] 0.02060222
```

4.9 Extra task performance over time

Match score over time

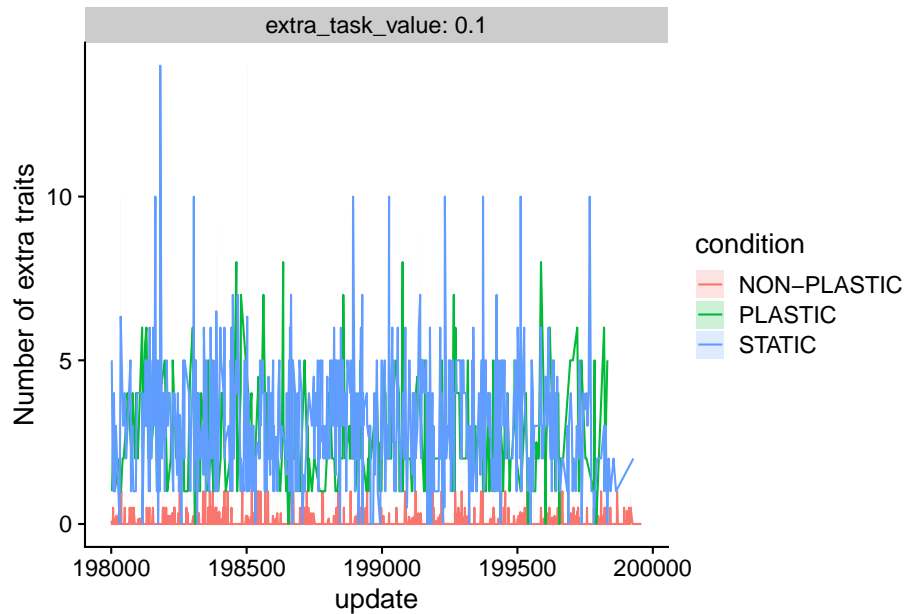
```
lineage_reward10 <- filter(lineage_time_series_data, extra_task_value=="0.1")
```

```
ggplot(filter(lineage_reward10, update>198000 & update<=200000), aes(x=update, y=match_score_even)) +
  stat_summary(fun="mean", geom="line") +
  stat_summary(
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    geom="ribbon",
    alpha=0.2,
    linetype=0
  ) +
  ylab("Match score (even environment)") +
  facet_wrap(
    ~extra_task_value,
    labeller=label_both
  ) +
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-match-score-even-val10.png"),
    width=15,
    height=10
  )
```



Extra tasks over time

```
ggplot(filter(lineage_reward10, update>198000 & update<=200000), aes(x=update, y=extra_tasks)) +
  stat_summary(fun="mean", geom="line") +
  stat_summary(
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    geom="ribbon",
    alpha=0.2,
    linetype=0
  ) +
  ylab("Number of extra traits") +
  facet_wrap(
    ~extra_task_value,
    labeller=label_both
  ) +
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-extra-traits-val10.png"),
    width=15,
    height=10
  )
```



4.10 Manuscript figures

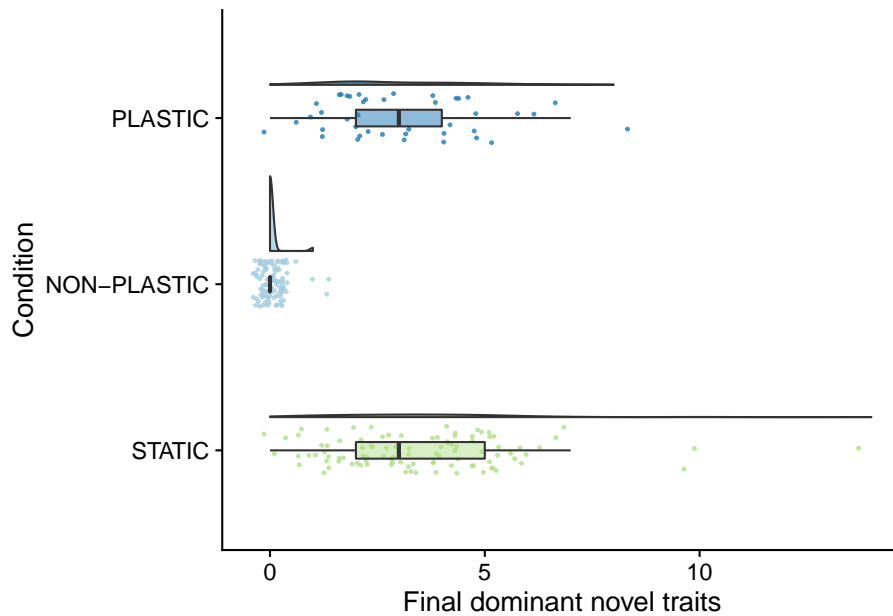
Final dominant extra tasks.

```
extra_task_reward_value=0.1
dominant_extra_tasks_fig <- ggplot(
  filter(summary_data, extra_task_value==extra_task_reward_value),
  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 dominant novel traits"
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
theme(
  legend.position="none"
) +
coord_flip()
dominant_extra_tasks_fig

```



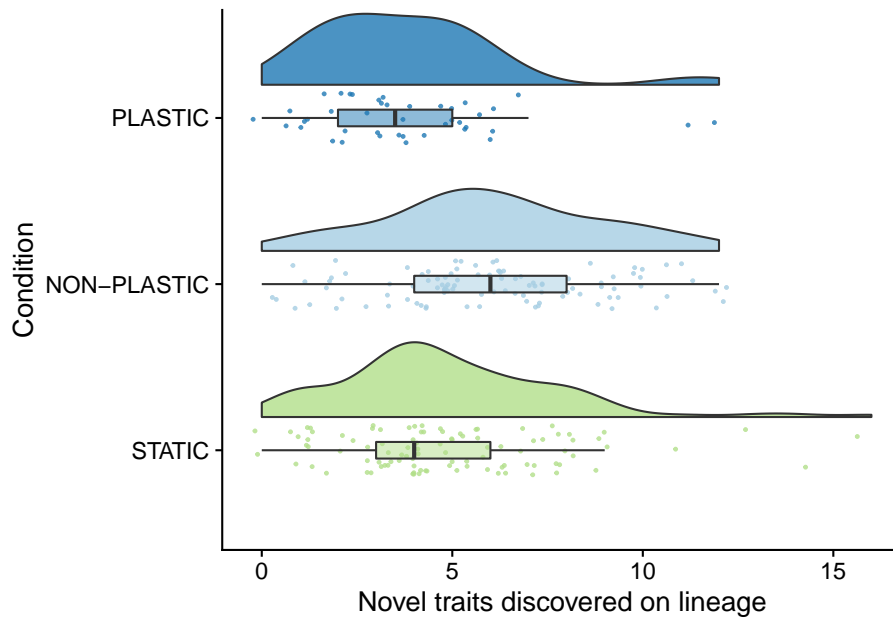
Final dominant lineage tasks discovered.

```

lineage_extra_tasks_discovered_fig <- ggplot(
  filter(summary_data, extra_task_value==extra_task_reward_value),
  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 traits discovered on lineage"  
) +  
scale_fill_brewer(  
  palette="Paired"  
) +  
scale_color_brewer(  
  palette="Paired"  
) +  
theme(  
  legend.position="none"  
) +  
coord_flip()  
lineage_extra_tasks_discovered_fig
```



Final dominant lineage tasks lost.

```
lineage_extra_tasks_lost_fig <- ggplot(
  filter(summary_data, extra_task_value == extra_task_reward_value),
  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
  )
```

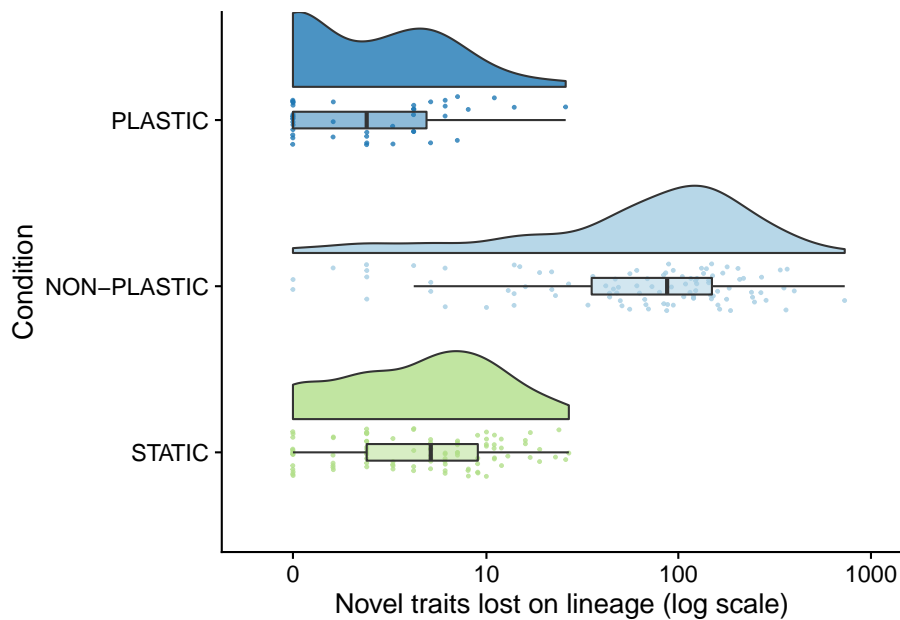


```

) +
scale_y_continuous(
  name="Novel traits lost on lineage (log scale)",
  trans="pseudo_log",
  breaks=c(0,10,100,1000),
  limits=c(-1,1000)
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
theme(
  legend.position="none"
) +
coord_flip()

```

lineage_extra_tasks_lost_fig



Pull it all together.

```

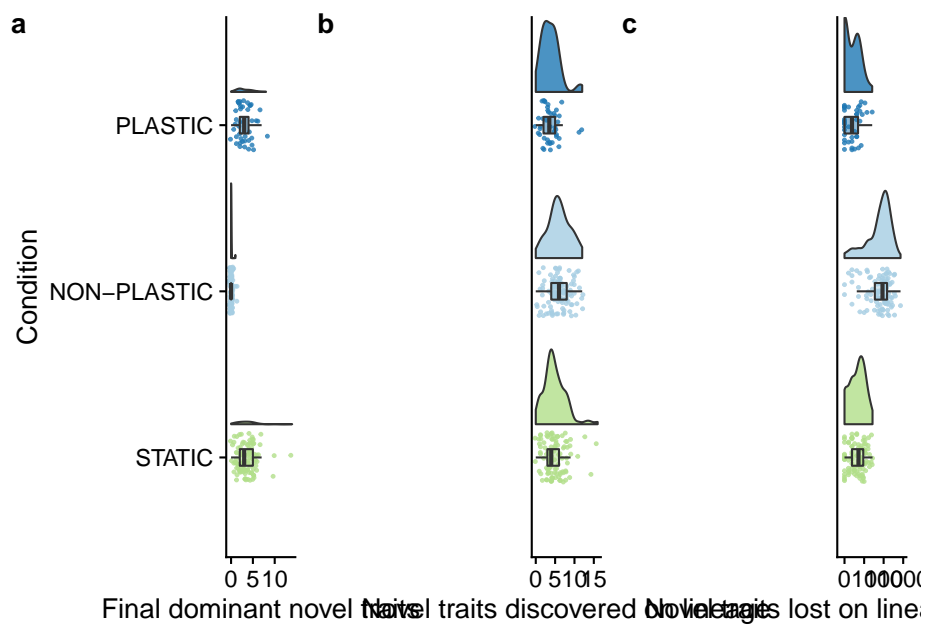
grid <- plot_grid(
  dominant_extra_tasks_fig,
  lineage_extra_tasks_discovered_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_bla

```

```

lineage_extra_tasks_lost_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_blank(),
nrow=1,
align="v",
labels="auto"
)
save_plot(
  paste0(working_directory, "plots/", "complex-traits-panel.pdf"),
  grid,
  base_height=6,
  base_asp=2.5
)
grid

```



Chapter 5

Genetic hitchhiking

The effect of adaptive phenotypic plasticity on (deleterious) genetic hitchhiking.

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)
library(boot)
library(fmsb)
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
```

5.3 Setup

```
##### 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$POISON_PENALTY <- as.factor(summary_data$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$env_label <- mapapply(
  env_label_fun,
  summary_data$chg_env
)
summary_data$sensors_label <- mapapply(
  sensors_label_fun,
  summary_data$sensors
)
summary_data$condition <- mapapply(
  condition_label_fun,
  summary_data$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_F
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 == "0"
lineage_time_series_data$POISON_PENALTY <- as.factor(lineage_time_series_data$POISON_VALUE)

lineage_time_series_data$env_label <- mapapply(

```

```

    env_label_fun,
    lineage_time_series_data$chg_env
  )
  lineage_time_series_data$sensors_label <- mapapply(
    sensors_label_fun,
    lineage_time_series_data$sensors
  )
  lineage_time_series_data$condition <- mapapply(
    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)

```

5.4 Evolution of phenotypic plasticity

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

```

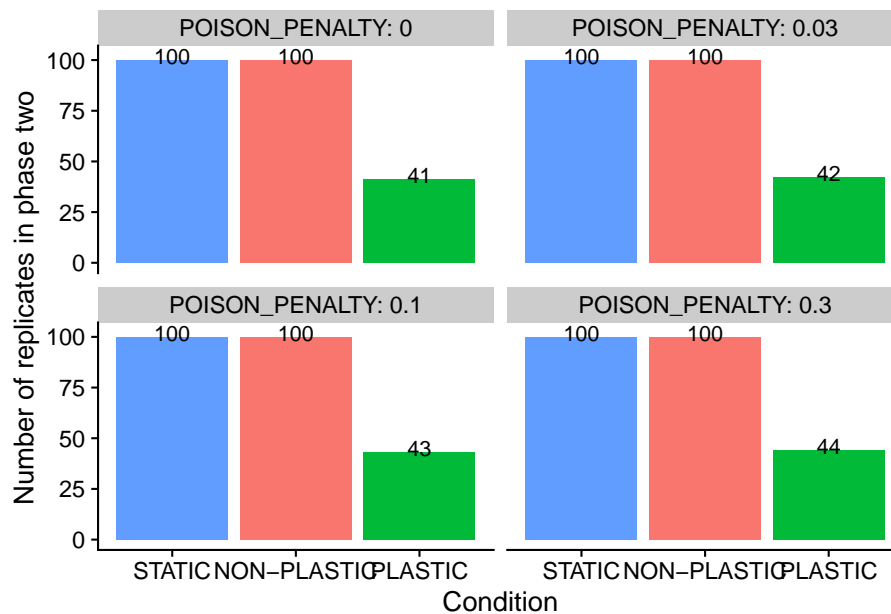
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition, POI)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())

```

```

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

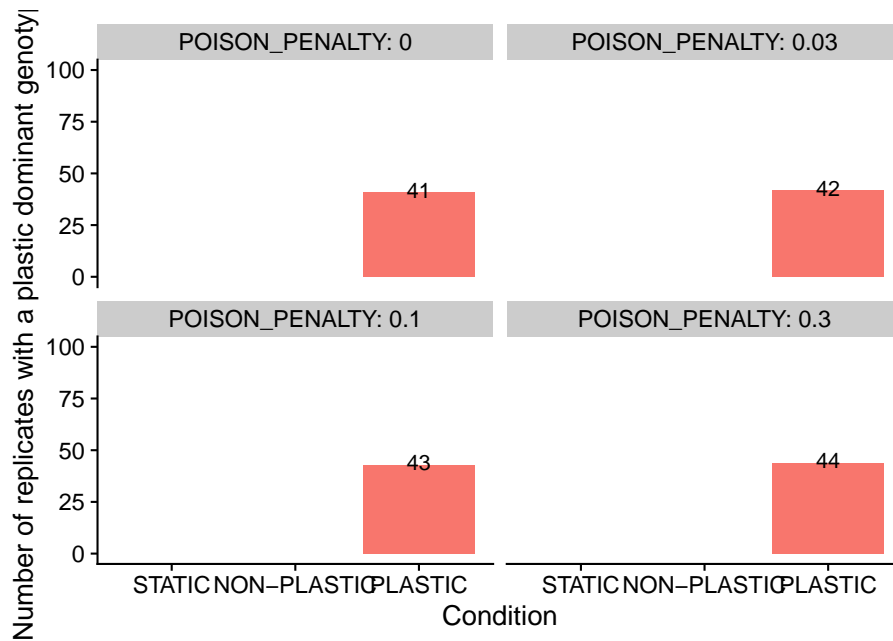
```



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 `.groups` argument.
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)) +
  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 Hitchhiking instruction execution

5.5.1 Number of replicates where final dominant genotype executes hitchhiker instruction

```
hitchiker_penalty <- 0.1
```

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



```

    "PLASTIC",
    "STATIC"
)

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 Final dominant genotype hitchhiker execution

```

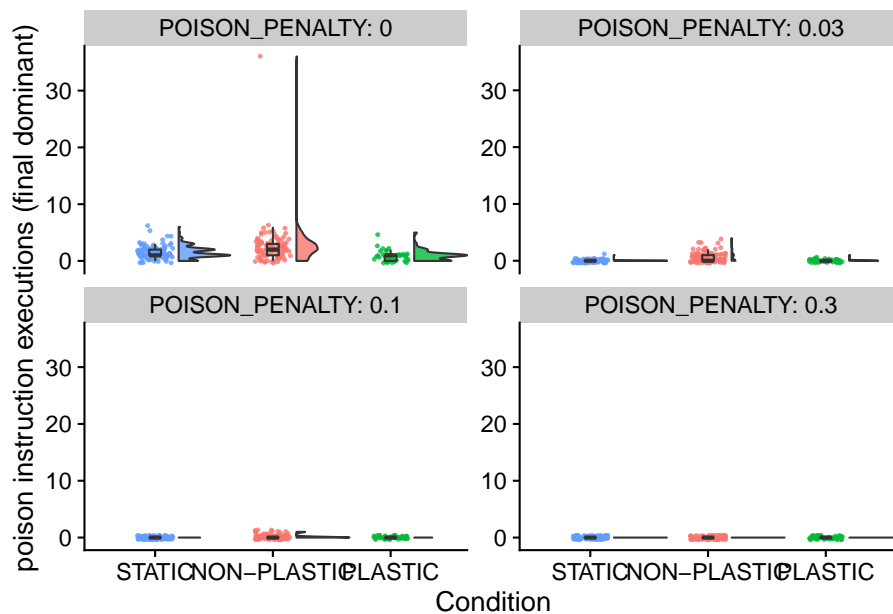
ggplot(summary_data, aes(x=condition, y=dominant_times_poison_executed, fill=condition)) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  ylab("poison instruction executions (final dominant)") +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both,
    scale="free_y"
  ) +

```

```

theme(
  legend.position="none"
) +
ggsave(
  paste0(working_directory, "plots/dominant-poison.pdf"),
  width=15,
  height=10
)

```



```

penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
    formula=dominant_times_poison_executed~condition,
    data=stat_data
  )
  print(
    kt
  )
  if (is.na(kt$p.value)) { next }
}

```

```

if (kt$p.value > 0.05) { next }
print(
  pairwise.wilcox.test(
    x=stat_data$dominant_times_poison_executed,
    g=stat_data$condition,
    p.adjust.method="bonferroni"
  )
)
}

## [1] "PENALTY: 0"
##
## Kruskal-Wallis rank sum test
##
## data: dominant_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 36.988, df = 2, p-value = 9.294e-09
##
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: stat_data$dominant_times_poison_executed and stat_data$condition
##
##      NON-PLASTIC PLASTIC
## PLASTIC 2.8e-07      -
## STATIC  0.00015     0.00198
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.03"
##
## Kruskal-Wallis rank sum test
##
## data: dominant_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 72.995, df = 2, p-value < 2.2e-16
##
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: stat_data$dominant_times_poison_executed and stat_data$condition
##
##      NON-PLASTIC PLASTIC
## PLASTIC 2.0e-06      -
## STATIC  2.8e-13      1
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"

```

```
##
## Kruskal-Wallis rank sum test
##
## data: dominant_times_poison_executed by condition
## Kruskal-Wallis chi-squared = 21.157, df = 2, p-value = 2.546e-05
##
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: stat_data$dominant_times_poison_executed and stat_data$condition
##
##          NON-PLASTIC PLASTIC
## PLASTIC 0.02034      -
## STATIC  0.00022      -
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.3"
##
## Kruskal-Wallis rank sum test
##
## data: dominant_times_poison_executed by condition
## Kruskal-Wallis chi-squared = NaN, df = 2, p-value = NA
```

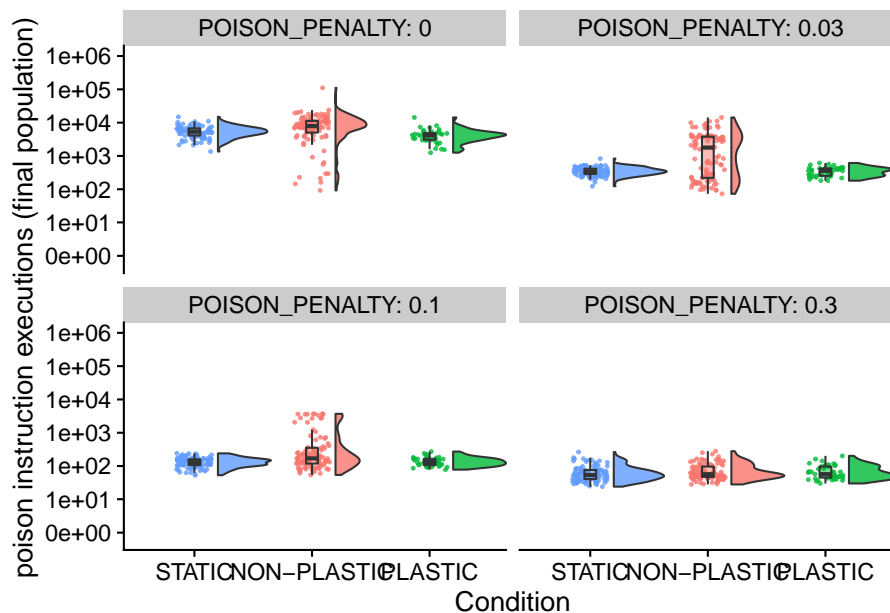
5.5.3 Hitchhiker instruction execution in final population

```
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(0,10,100,1000, 10000, 100000, 1000000),
  limits=c(-1,1000000)
) +
facet_wrap(
  ~POISON_PENALTY,
  labeller=label_both
) +
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
##
##

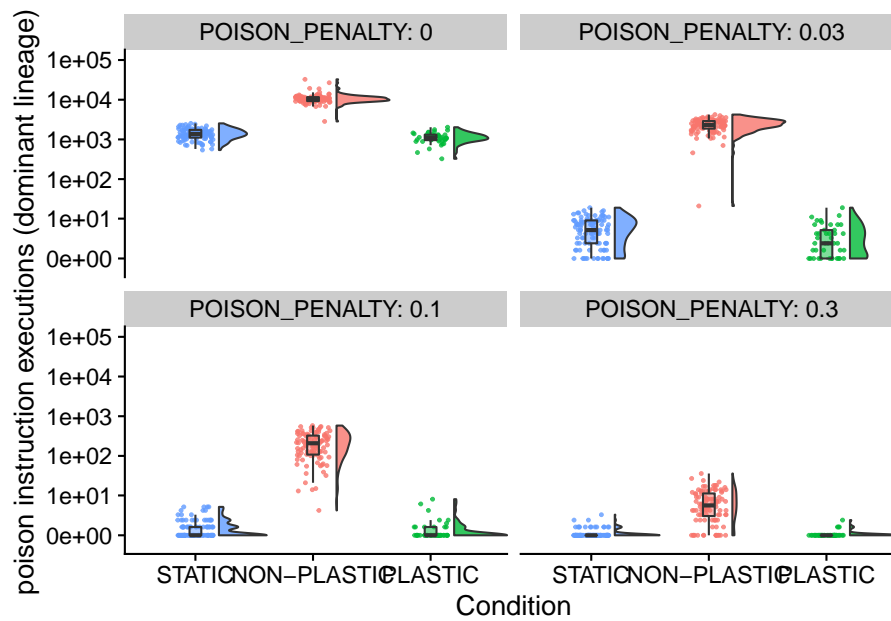
```

```
## 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
```

5.5.4 Hitchhiker instruction execution along final dominant lineage (cummulative)

```
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(0,10,100,1000,10000,100000),
    limits=c(-1,100000)
  ) +
  facet_wrap(
    ~POISON_PENALTY,
    labeller=label_both
  ) +
  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 hitchhiker instruction execution

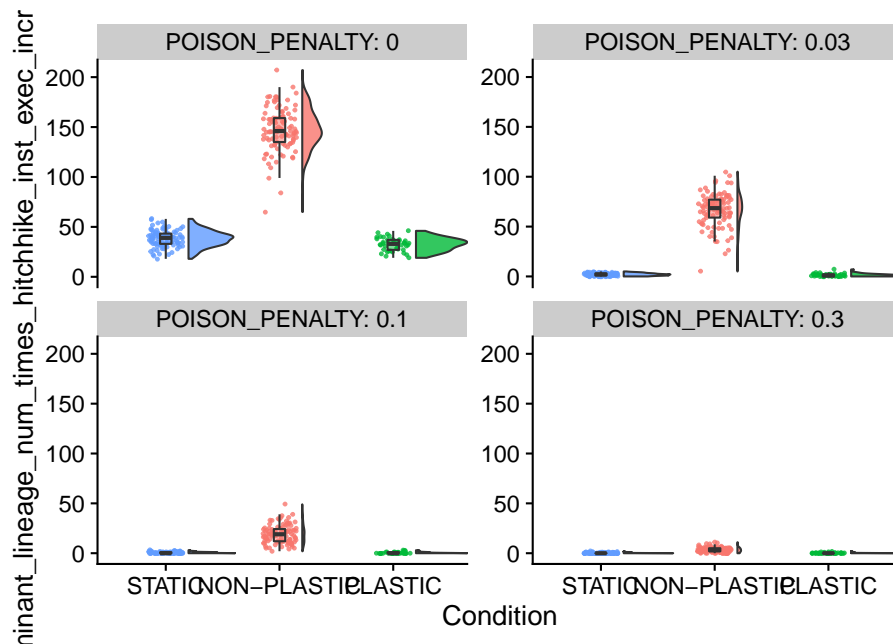
5.6.1 Number of offspring along dominant lineage with increase in hitchhiker instruction execution

```
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
) +
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-mutant"),
  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$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_data$condition
##
##      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$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_c
##
##          NON-PLASTIC PLASTIC
## PLASTIC <2e-16      -
## STATIC  <2e-16      0.03
##
## P value adjustment method: bonferroni
## [1] "PENALTY: 0.1"
##
## Kruskal-Wallis rank sum test
##
## data: dominant_lineage_num_times_hitchhike_inst_exec_increases by condition
## Kruskal-Wallis chi-squared = 185.34, df = 2, p-value < 2.2e-16
##
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: stat_data$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_c
##
##          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$dominant_lineage_num_times_hitchhike_inst_exec_increases and stat_c
##
##          NON-PLASTIC PLASTIC
## PLASTIC 7.8e-16      -
## STATIC  < 2e-16      0.86
##
## P value adjustment method: bonferroni
sum(filter(summary_data, condition=="NON-PLASTIC" & POISON_PENALTY==0.1)$dominant_lineage_num_times_hitchhike_inst_exec_increases)

```

5.6. CHARACTERIZING MUTATIONS THAT INCREASE HITCHHIKER INSTRUCTION EXECUTION127

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

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

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

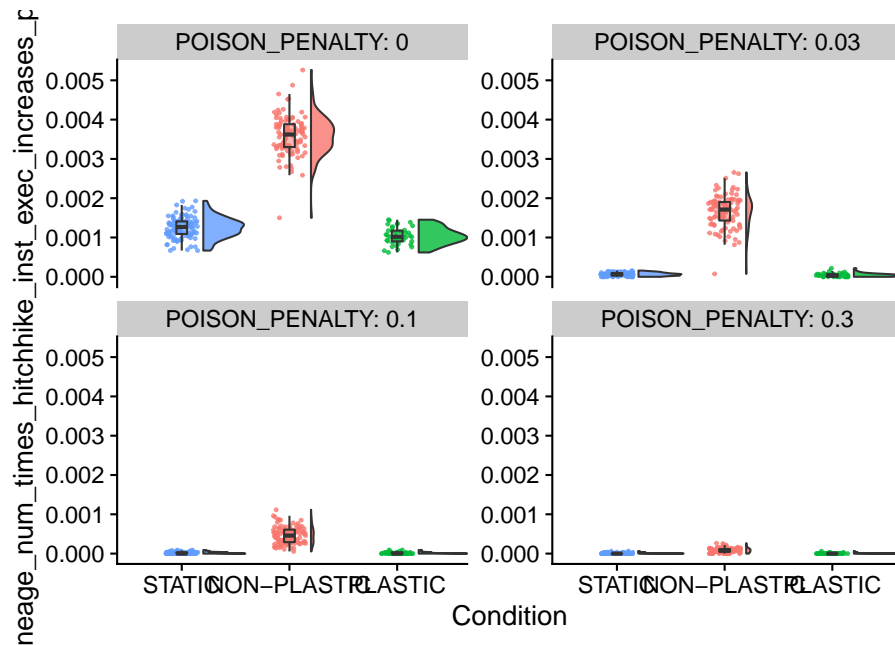
```
sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_lineage_num_times_hi
```

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

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

5.6.1.1 Normalized by generations

```
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  
  ) +  
  facet_wrap(  
    ~POISON_PENALTY,  
    labeller=label_both,  
    scales="free_y"  
  ) +  
  theme(  
    legend.position="none"  
  )  
)
```



```

penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
    formula=dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation ~ condition,
    data=stat_data
  )
  print(
    kt
  )
  if (is.na(kt$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_per_generation,
      g=stat_data$condition,
      p.adjust.method="bonferroni"
    )
  )
}

```


5.6. CHARACTERIZING MUTATIONS THAT INCREASE HITCHHIKER INSTRUCTION EXECUTION129

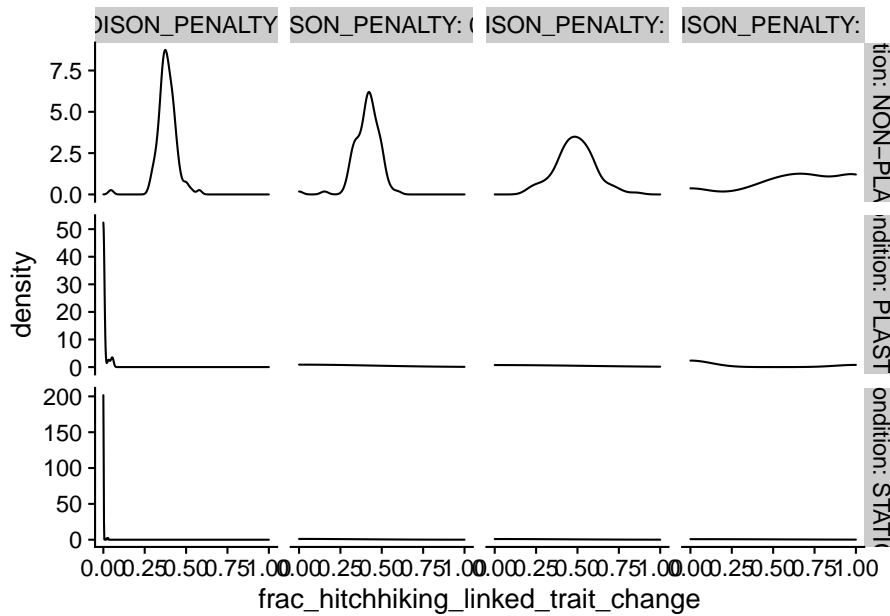
```
## [1] "PENALTY: 0"
##
## Kruskal-Wallis rank sum test
##
## data: dominant_lineage_num_times_hitchhike_inst_exec_increases_per_generation by condition
## Kruskal-Wallis chi-squared = 180.05, df = 2, p-value < 2.2e-16
##
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## 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 c
## 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
```

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

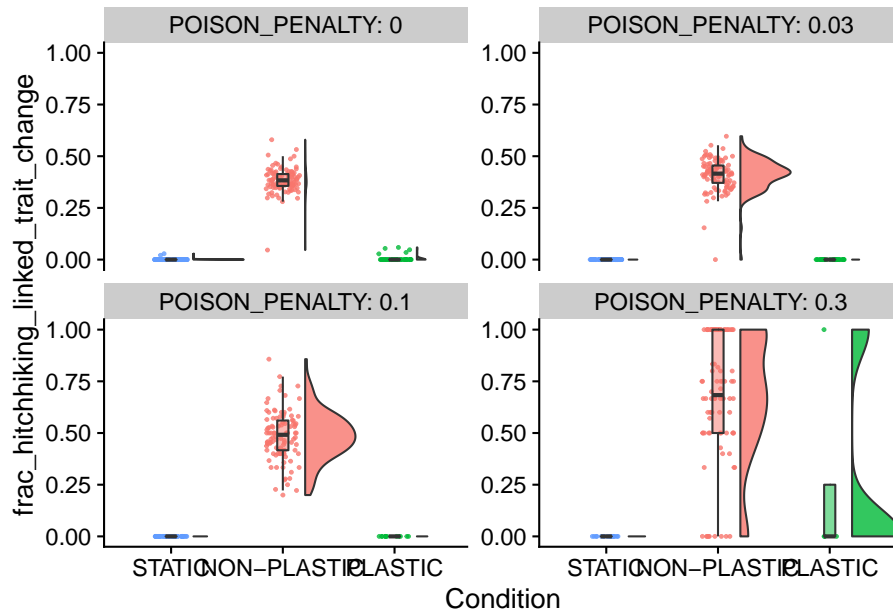
```
# Fraction of unexpressed vs expressed increases in hitchhiker instructions
summary_data$frac_hitchhiking_linked_trait_change <- summary_data$dominant_lineage_num_

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"
  ) +
  theme(
    legend.position="none"
  ) +
  ggsave(
    paste0(working_directory, "plots/dominant-lineage-frac_hitchhiking_linked_trait_cha
    width=15,
    height=10
  )
```



```
ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0 ), aes(x=frac_hitchhiking_linked_trait_change)) +
  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
  ) +
  facet_wrap(
    ~POISSON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
```

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



```
penalties <- levels(summary_data$POISON_PENALTY)
for (penalty in penalties) {
  stat_data <- filter(summary_data, POISON_PENALTY==penalty & dominant_lineage_num_time > 0)
  print(
    paste0(
      "PENALTY: ", penalty
    )
  )
  kt <- kruskal.test(
    formula=frac_hitchhiking_linked_trait_change~condition,
    data=stat_data
  )
  print(
    kt
  )
  if (is.na(kt$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,
```

5.6. CHARACTERIZING MUTATIONS THAT INCREASE HITCHHIKER INSTRUCTION EXECUTION133

```

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

## [1] "PLASTIC: 0 (0/18)"
denom <- sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_allele)
num <- sum(filter(summary_data, condition=="STATIC" & POISON_PENALTY==0.1)$dominant_allele)
paste0("STATIC: ", num/denom, " (", num, "/", denom, ")")

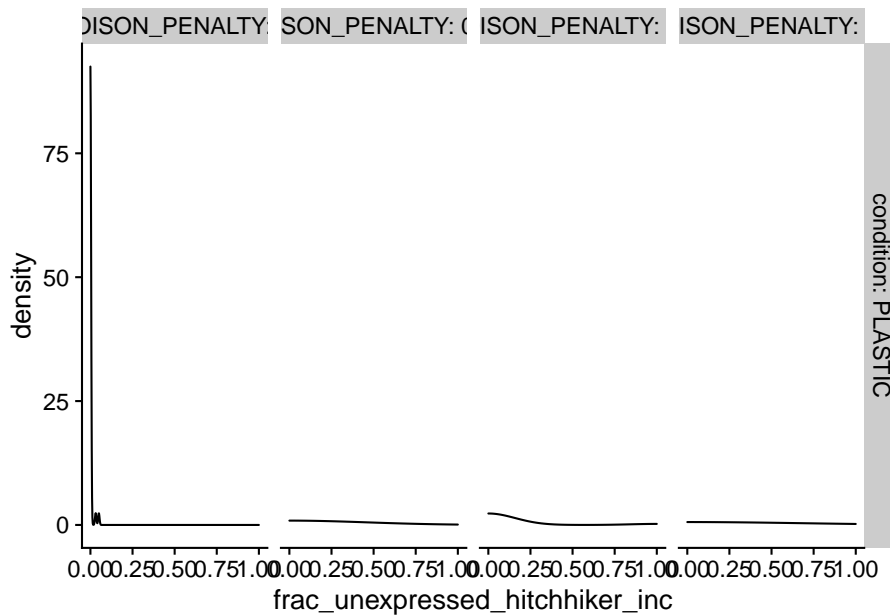
## [1] "STATIC: 0 (0/58)"

```

5.6.3 What about unexpressed vs expressed trait changes in plastic populations?

```
summary_data$frac_unexpressed_hitchhiker_inc <- summary_data$dominant_lineage_num_times_hitchhike_inst_exec_increases>0
summary_data$frac_expressed_hitchhiker_inc <- summary_data$dominant_lineage_num_times_hitchhike_inst_exec_increases>0

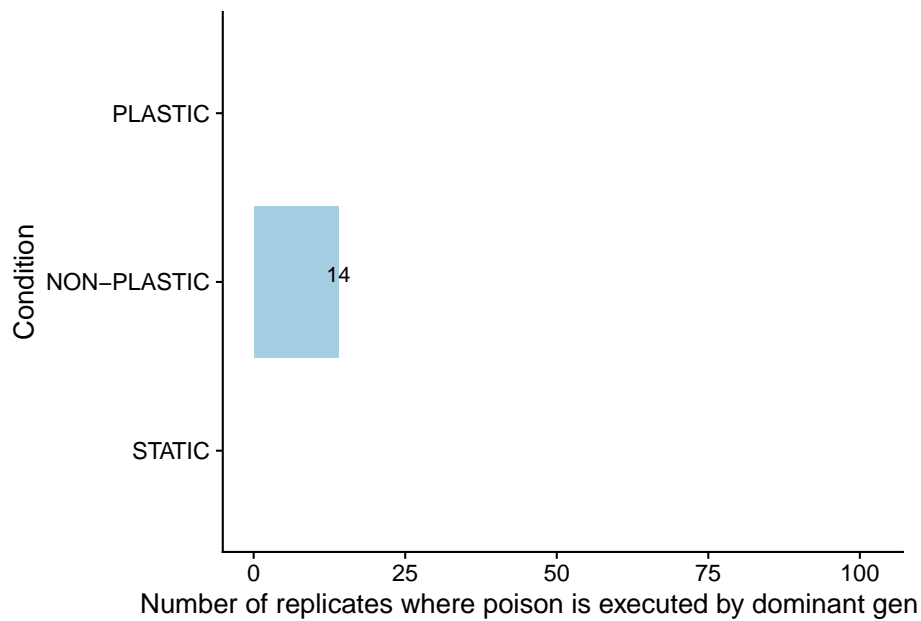
ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0 & condition=="PLASTIC")) +
  geom_density() +
  facet_grid(
    condition~POISON_PENALTY,
    labeller=label_both,
    scales="free_y"
  ) +
  theme(
    legend.position="none"
  )
```



```
# ggplot(filter(summary_data, dominant_lineage_num_times_hitchhike_inst_exec_increases>0 & condition=="PLASTIC")) +
#   geom_density() +
#   facet_grid(
#     condition~POISON_PENALTY,
#     labeller=label_both,
#     scales="free_y"
#   ) +
#   theme(
```

```
# legend.position="none"
# )
```

```
hitchiker_penalty <- 0.1
5.7 Manuscript Figures
ggplot(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="NON-PLASTIC")) +
  geom_bar() +
  geom_text(
    stat="count",
    mapping=aes(label=..count..),
    position=position_dodge(0.9),
    vjust=0
  ) +
  scale_y_continuous(
    name="Number of replicates where poison is executed by dominant genotype",
    limits=c(0, replicates+2)
  ) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order,
    labels=condition_order,
    breaks=condition_order
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  coord_flip() +
  theme(
    legend.position="none"
  )
)
```

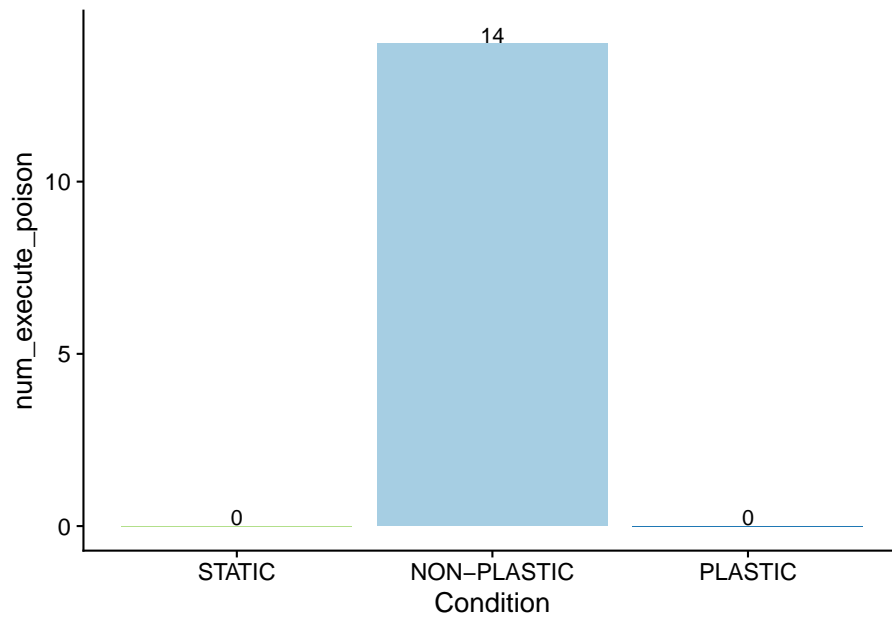
```
fig_data <- data.frame(
  num_execute_poison=c(
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="NON-PLASTIC" & domina
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="PLASTIC" & domina
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="STATIC" & domina
  )),
  num_execute_no_poison=c(
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="NON-PLASTIC" & do
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="PLASTIC" & domina
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="STATIC" & domina
  )),
  total=c(
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="NON-PLASTIC")$RANDOM
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="PLASTIC")$RANDOM
    length(filter(summary_data, POISON_PENALTY==hitchiker_penalty & condition=="STATIC" )$RANDOM
  )),
  condition=c(
    "NON-PLASTIC",
    "PLASTIC",
    "STATIC"
  )
)

ggplot(fig_data, aes(x=condition, y=num_execute_poison, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
```

```

geom_text(aes(label=num_execute_poison, y=num_execute_poison+0.25)) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
theme(
  legend.position="none"
)

```



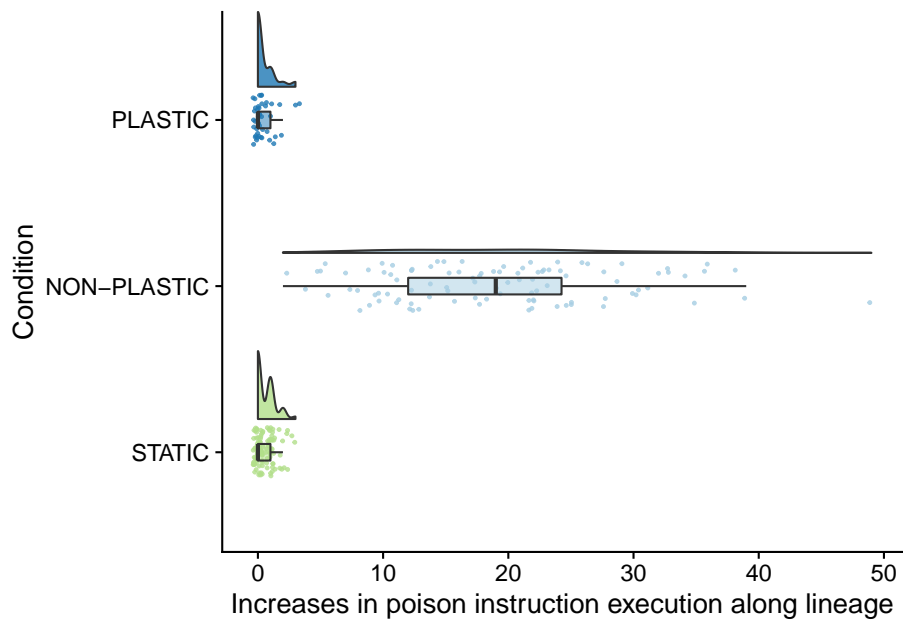
Number of poison increases along lineage.

```

poison_increases_fig <- ggplot(
  filter(summary_data, POISON_PENALTY==hitchiker_penalty),
  aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases, 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,  
  labels=condition_order  
) +  
scale_y_continuous(  
  name="Increases in poison instruction execution along lineage",  
) +  
scale_fill_brewer(  
  palette="Paired"  
) +  
scale_color_brewer(  
  palette="Paired"  
) +  
theme(  
  legend.position="none"  
) +  
coord_flip()  
  
poison_increases_fig
```

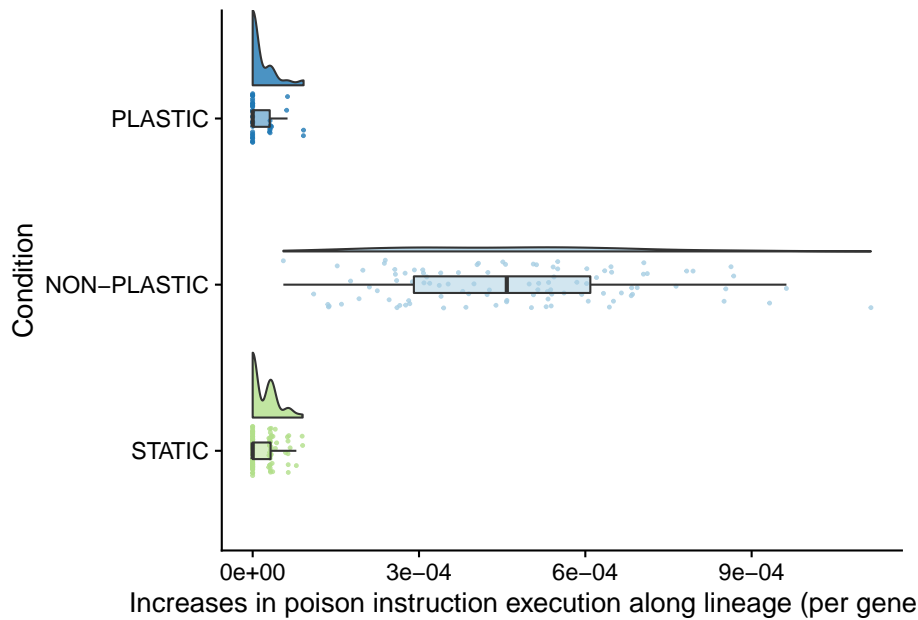


```
poison_increases_per_gen_fig <- ggplot(
  filter(summary_data, POISON_PENALTY==hitchhiker_penalty),
  aes(x=condition, y=dominant_lineage_num_times_hitchhike_inst_exec_increases_per_gen)
) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8
  ) +
  geom_point(
    mapping=aes(color=condition),
    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 instruction execution along lineage (per generation)",
  ) +
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
  coord_flip()
poison_increases_per_gen_fig

```



Pull it all together.

```

# grid <- plot_grid(
#   dominant_extra_tasks_fig,
#   lineage_extra_tasks_discovered_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_blank()),
#   lineage_extra_tasks_lost_fig + theme(axis.ticks.y=element_blank(),axis.text.y=element_blank()),
#   nrow=1,
#   align="v",
#   labels="auto"
# )

```

```
# save_plot(  
#   paste0(working_directory, "plots/", "complex-traits-panel.pdf"),  
#   grid,  
#   base_height=6,  
#   base_asp=2.5  
# )  
# grid
```