Supplemental Material

Alexander Lalejini, Austin J. Ferguson, and Charles Ofria

2021-02-03

Contents

Chapter 1

Introduction

TODO

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.

```r
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:

```r
print(version)
```

```
##               _
## platform      x86_64-pc-linux-gnu
## arch          x86_64
## os            linux-gnu
## system        x86_64, linux-gnu
## status
## major         4
## minor         0.3
## year          2020
## month         10
## day           10
## svn rev       79318
## language      R
## version.string R version 4.0.3 (2020-10-10)
## nickname      Bunny-Wunnies Freak Out
```

2.3   Setup

```r
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)
```

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

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

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

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

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

```
    )
  }
}

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


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

2.4   Evolution of phenotypic plasticity

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

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

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

To what extent is the observed phenotypic plasticity adaptive?

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

)
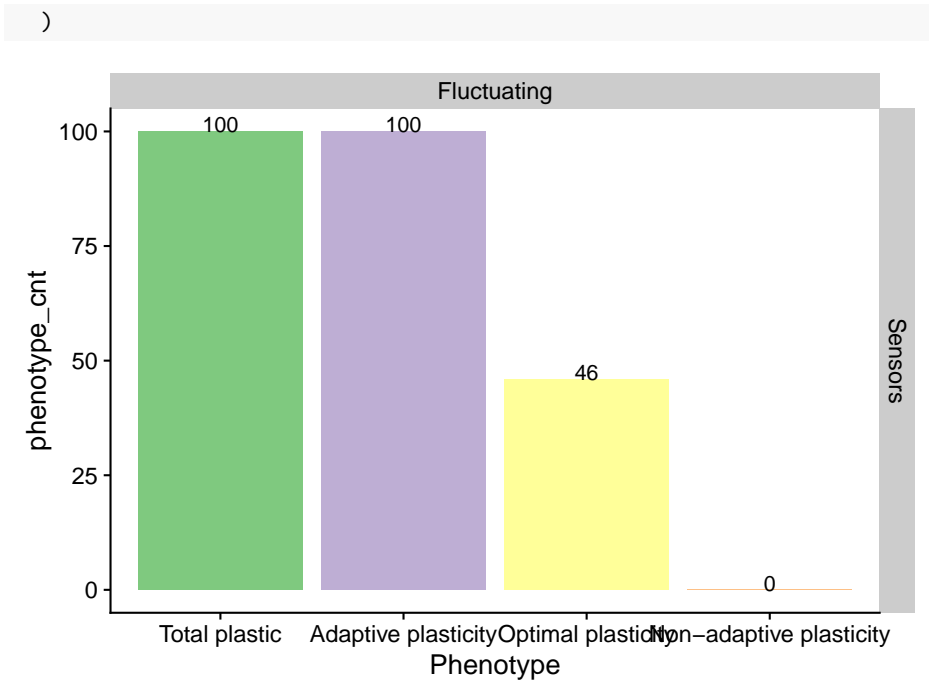
Chapter 3

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/2021-01-30-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)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9
```

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

```r
print(version)
```

```
##                 _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          0.3
## year           2020
## month          10
## day            10
## svn rev        79318
## language       R
## version.string R version 4.0.3 (2020-10-10)
## nickname       Bunny-Wunnies Freak Out
```

3.3 Setup

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

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

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

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

```r
  }
}

# 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"
)

###### time series #####
# time_series_data_loc <- paste0(working_directory, "data/time_series_u0-u200000.csv")
time_series_data_loc <- paste0(working_directory, "data/time_series_u95000-u105000.csv")

time_series_data <- read.csv(time_series_data_loc)
time_series_data$DISABLE_REACTION_SENSORS <- as.factor(time_series_data$DISABLE_REACTION_SENSORS)
time_series_data$chg_env <- time_series_data$chg_env == "True"
time_series_data$sensors <- time_series_data$DISABLE_REACTION_SENSORS == "0"

time_series_data$env_label <- mapply(
  env_label_fun,
```

```r
  time_series_data$chg_env
)
time_series_data$sensors_label <- mapply(
  sensors_label_fun,
  time_series_data$sensors
)
time_series_data$condition <- mapply(
  condition_label_fun,
  time_series_data$sensors,
  time_series_data$chg_env
)


####### 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 transfered
populations containing an optimally plastic genotype to phase-two.

```r
summary_data_grouped = dplyr::group_by(summary_data, sensors, env_label, condition)
summary_data_group_counts = dplyr::summarize(summary_data_grouped, n=dplyr::n())
```
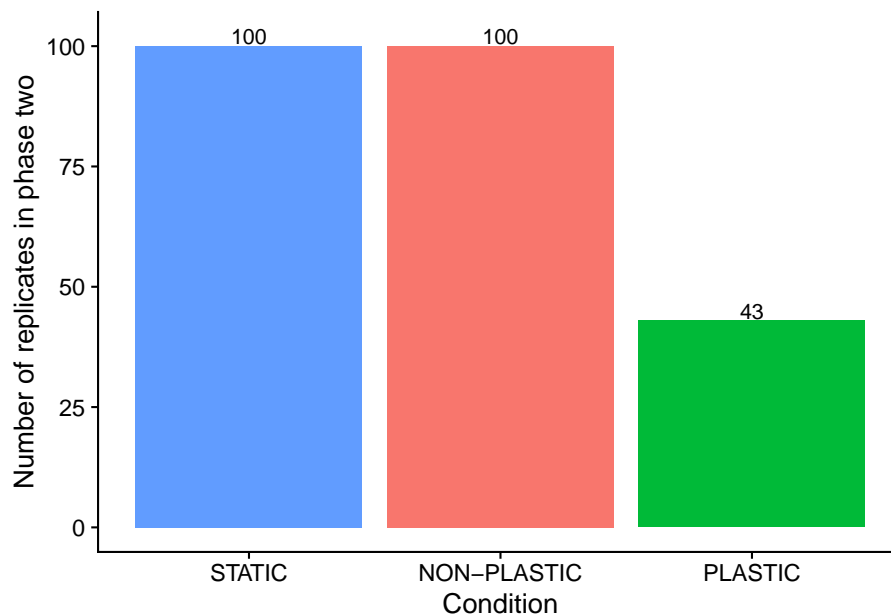
```
## `summarise()` has grouped output by 'sensors', 'env_label'. You can override using
```
```r
ggplot(summary_data_group_counts, aes(x=condition, y=n, fill=condition)) +
  geom_col(position=position_dodge(0.9)) +
  geom_text(aes(label=n, y=n+2)) +
  scale_x_discrete(
    name="Condition",
    limits=condition_order
  ) +
  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())
```

```
## `summarise()` has grouped output by 'condition'. 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") +
  theme(
    legend.position="none"
  )
```
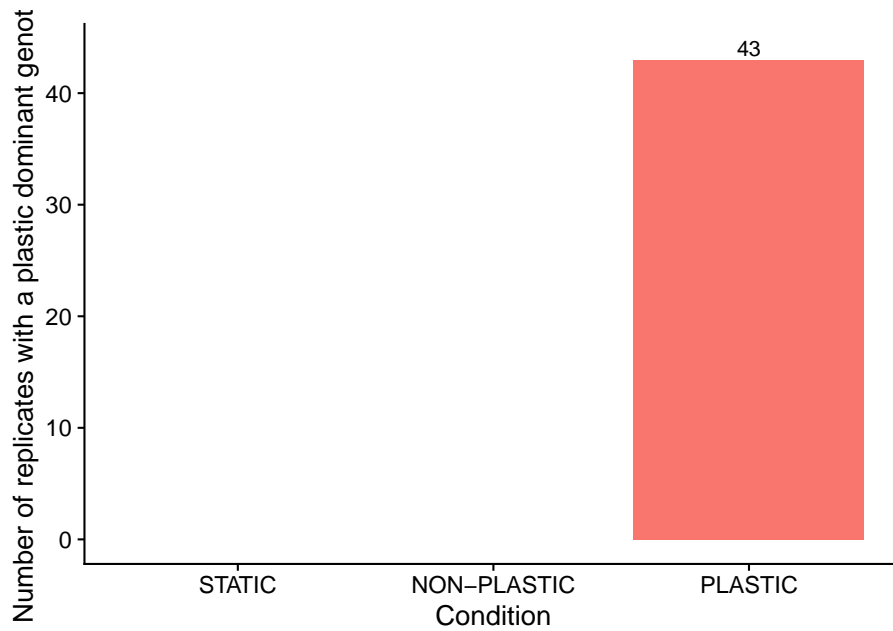
3.5 Average generation

```
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
  ) +
  ylab("average generation") +
```

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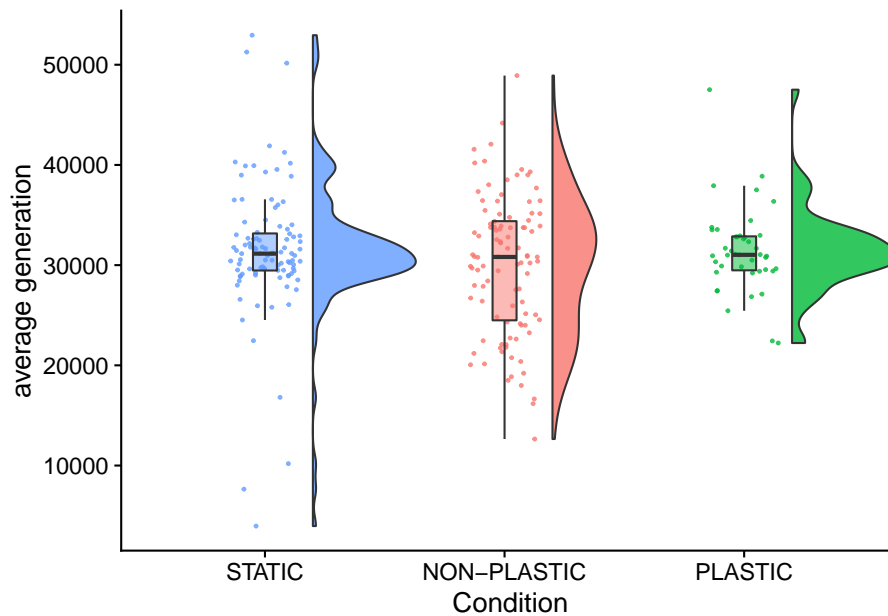


```
median(filter(summary_data, condition=="PLASTIC")$time_average_generation)
```

```
## [1] 31028.6
```
```
median(filter(summary_data, condition=="STATIC")$time_average_generation)
```

```
## [1] 31147.5
```
```
median(filter(summary_data, condition=="NON-PLASTIC")$time_average_generation)
```

```
## [1] 30817.95
```
```
kruskal.test(
  formula=time_average_generation~condition,
  data=summary_data
)
```

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

3.6 Phenotypic volatility along dominant lineage

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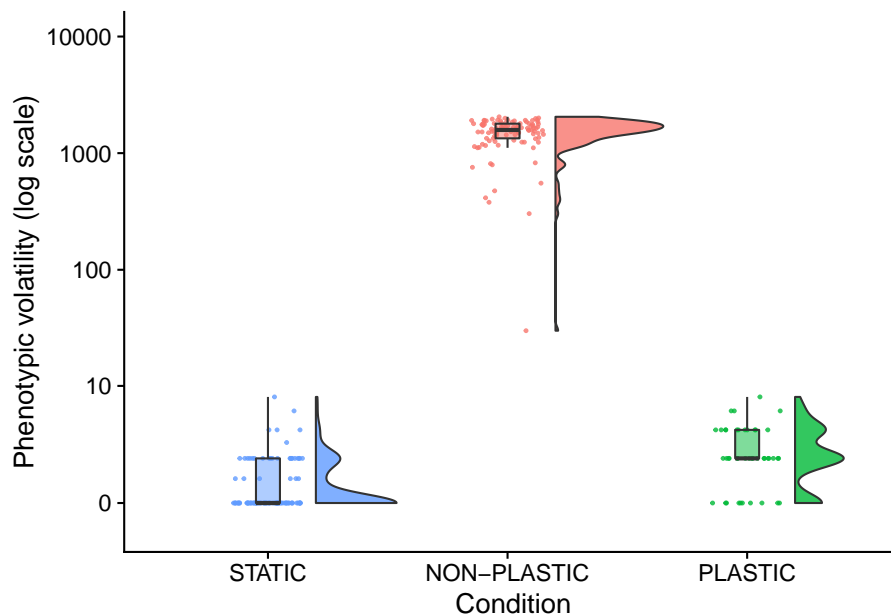
```
median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_trait_volatility)
```

```
## [1] 2
```

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

```
## [1] 0
```

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

```
## [1] 1580
```

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

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

```
pairwise.wilcox.test(
  x=summary_data$dominant_lineage_trait_volatility,
  g=summary_data$condition,
  p.adjust.method="bonferroni",
)
```
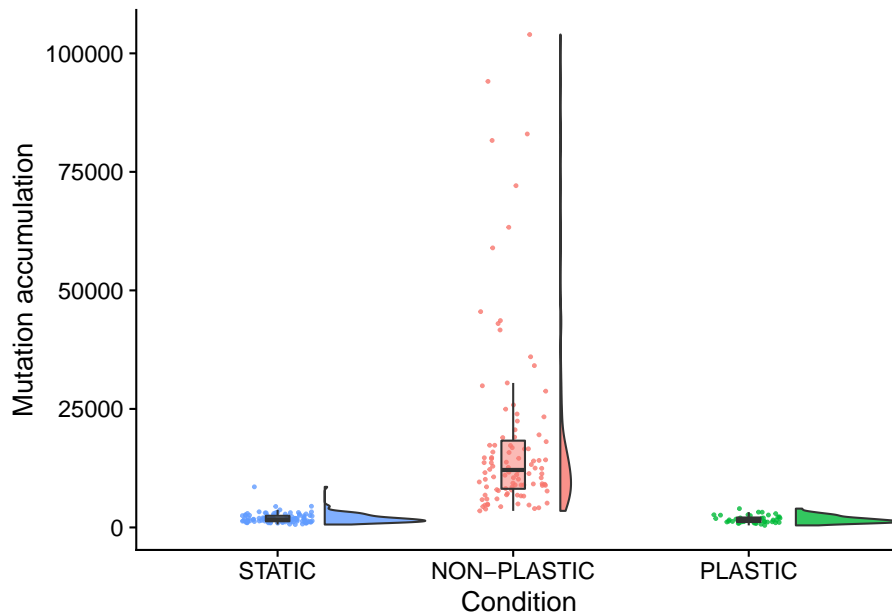
```
## 
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
## 
## data:  summary_data$dominant_lineage_trait_volatility and summary_data$condition
## 
##         NON-PLASTIC PLASTIC
## PLASTIC < 2e-16     -
## STATIC  < 2e-16     5.2e-08
## 
## P value adjustment method: bonferroni
```

3.7 Mutation accumulation along the dominant

```r
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"
  )
```

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

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

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

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

```
## P value adjustment method: bonferroni
median(filter(summary_data, condition=="PLASTIC")$dominant_lineage_total_mut_cnt)
```

```
## [1] 1552
median(filter(summary_data, condition=="STATIC")$dominant_lineage_total_mut_cnt)
```

```
## [1] 1724.5
median(filter(summary_data, condition=="NON-PLASTIC")$dominant_lineage_total_mut_cnt)
```

```
## [1] 12123
```

3.7.1 Mutation accumulation normalized by lineage length

```
summary_data$mutations_per_lineage_step <- summary_data$dominant_lineage_total_mut_cnt
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"
  )
```
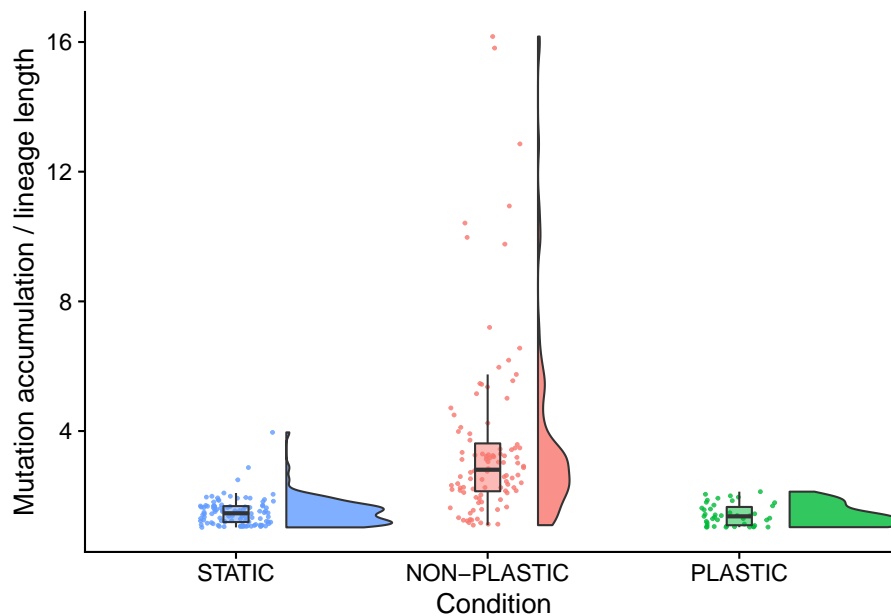
```
kruskal.test(
  formula=mutations_per_lineage_step~condition,
  data=summary_data
)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  mutations_per_lineage_step by condition
## Kruskal-Wallis chi-squared = 106.1, 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 8.9e-14      -
## STATIC  < 2e-16      0.64
##
```

```
## P value adjustment method: bonferroni
median(filter(summary_data, condition=="PLASTIC")$mutations_per_lineage_step)
```

```
## [1] 1.369198
median(filter(summary_data, condition=="NON-PLASTIC")$mutations_per_lineage_step)
```

```
## [1] 2.807136
median(filter(summary_data, condition=="STATIC")$mutations_per_lineage_step)
```

```
## [1] 1.464488
```

3.8   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"
  )
```
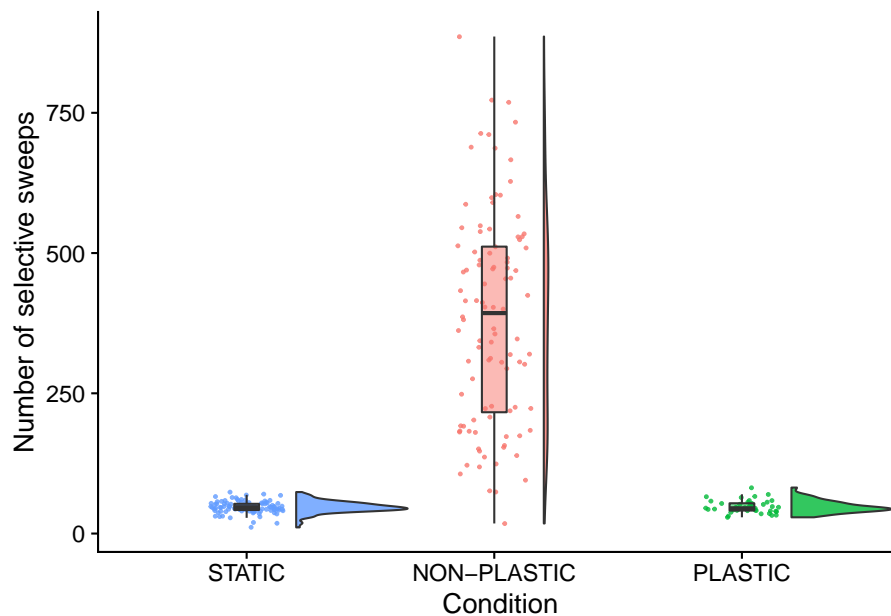
```r
kruskal.test(
  formula=phylo_mrca_changes~condition,
  data=summary_data
)
```

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

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

```
## [1] 45
median(filter(summary_data, condition=="STATIC")$phylo_mrca_changes)
```

```
## [1] 47
median(filter(summary_data, condition=="NON-PLASTIC")$phylo_mrca_changes)
```

```
## [1] 393
# low_mrca_change <- filter(summary_data, condition=="Non-plastic (fluctuating)" & phy
```

3.9 Genome length

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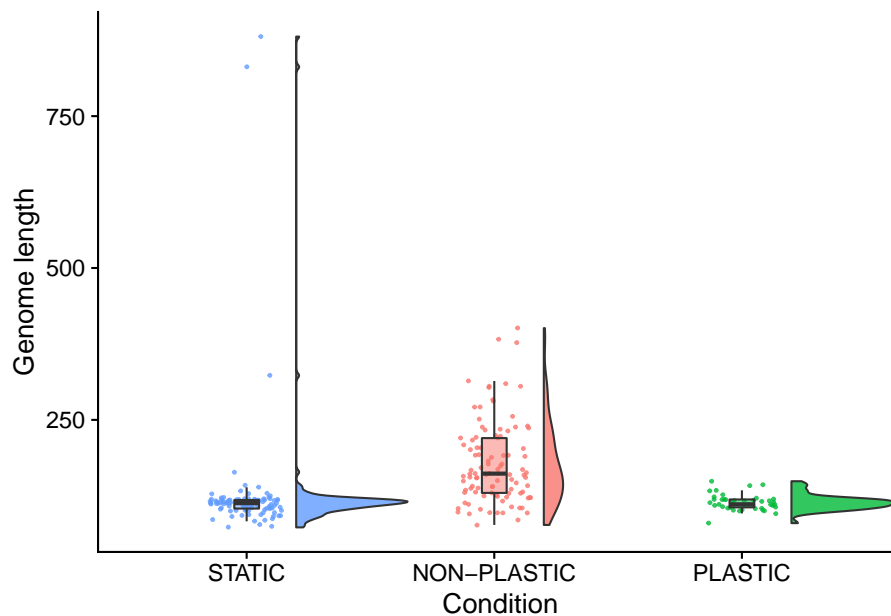
```r
kruskal.test(
  formula=dominant_genome_length~condition,
  data=summary_data
)
```

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

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

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

```
## P value adjustment method: bonferroni
median(filter(summary_data, condition=="PLASTIC")$phylo_mrca_changes)
```

```
## [1] 45
median(filter(summary_data, condition=="STATIC")$phylo_mrca_changes)
```

```
## [1] 47
median(filter(summary_data, condition=="NON-PLASTIC")$phylo_mrca_changes)
```

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

## 3.10   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"
  )
```
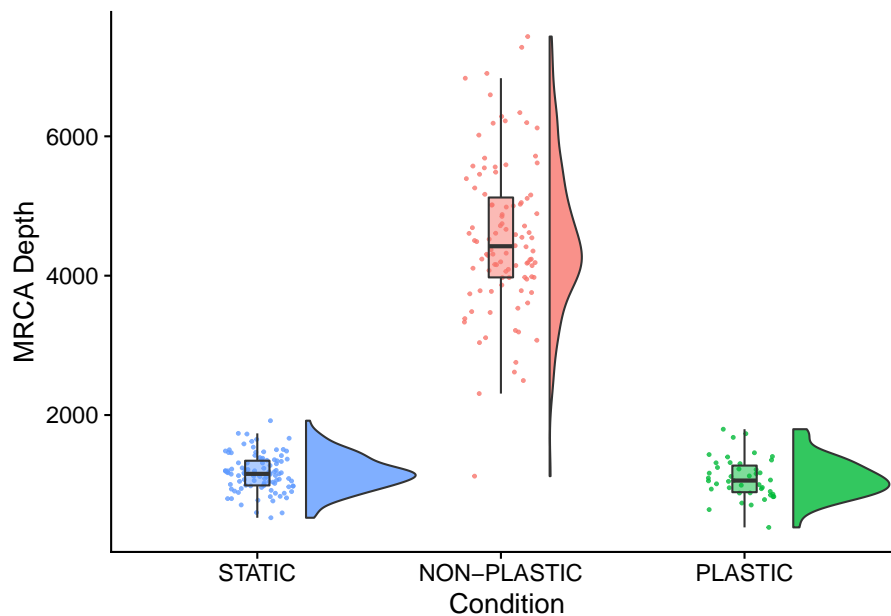
3.11 Phylogenetic diversity over time

```
// From (Faith 1992, reviewed in Winters et al., 2013), phylogenetic diversity is the sum of edge
// This calculates phylogenetic diversity for all extant taxa in the tree.
ggplot(time_series_data, aes(x=update, y=phylo_current_phylogenetic_diversity, fill=condition, co
  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("Phylogenetic diversity") +
  ggsave(
    paste0(working_directory, "plots/", "phylo-diversity-ot.png"),
    width=10,
    height=5
  )
```
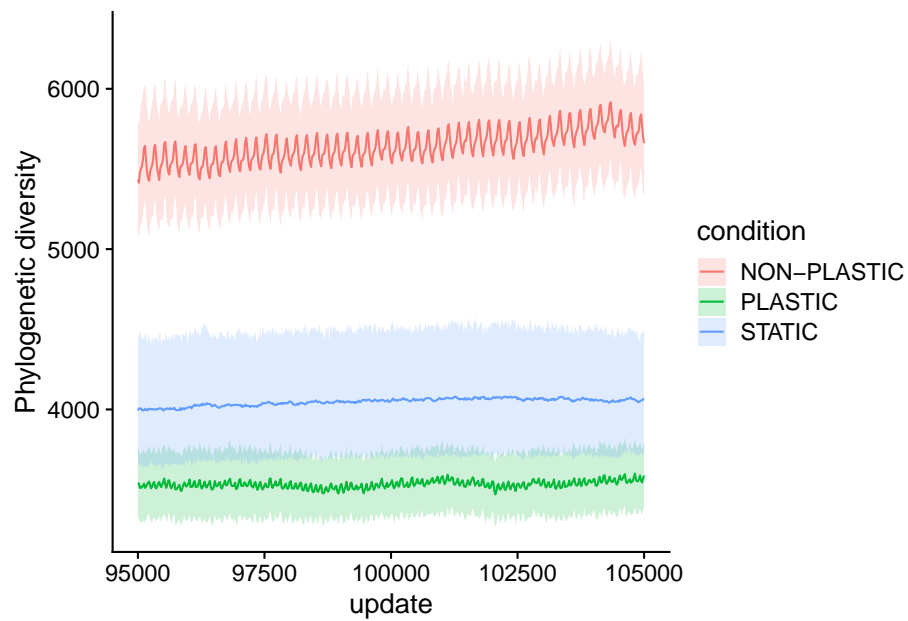
3.12   Number of extant taxa (genotypes) over

```r
ggplot(time_series_data, aes(x=update, y=phylo_num_taxa_extant, fill=condition, color=
  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
  ) +
  ggsave(
    paste0(working_directory, "plots/", "phylo-num-extant-taxa-ot.png"),
    width=10,
    height=5
  )
```
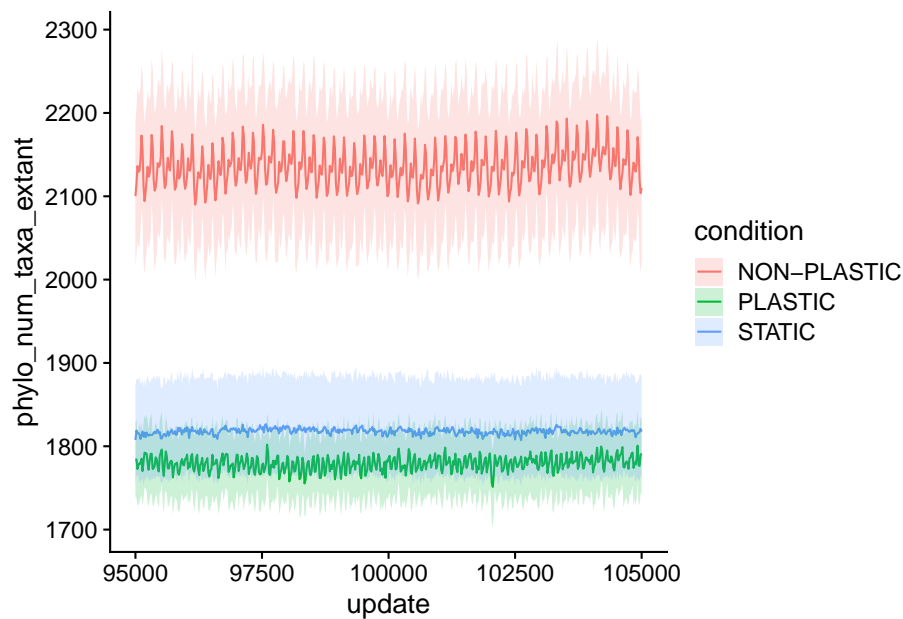
## 3.13   Evolutionary distinctiveness over time

```
/** This metric (from Isaac, 2007; reviewed in Winter et al., 2013) measures how
 * distinct @param tax is from the rest of the population, weighted for the amount of
 * unique evolutionary history that it represents.
 *
 * To quantify length of evolutionary history, this method needs @param time: the current
 * time, in whatever units time is being measured in when taxa are added to the systematics
 * manager. Note that passing a time in the past will produce innacurate results (since we
 * don't know what the state of the tree was at that time).
 *
 * Assumes the tree is all connected. Will return -1 if this assumption isn't met.
 */
```

```
ggplot(time_series_data, aes(x=update, y=phylo_mean_evolutionary_distinctiveness, fill=condition,
  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
  ) +
  ggsave(
```

```
    paste0(working_directory, "plots/", "phylo_mean_evolutionary_distinctiveness-ot.png
    width=10,
    height=5
)
```
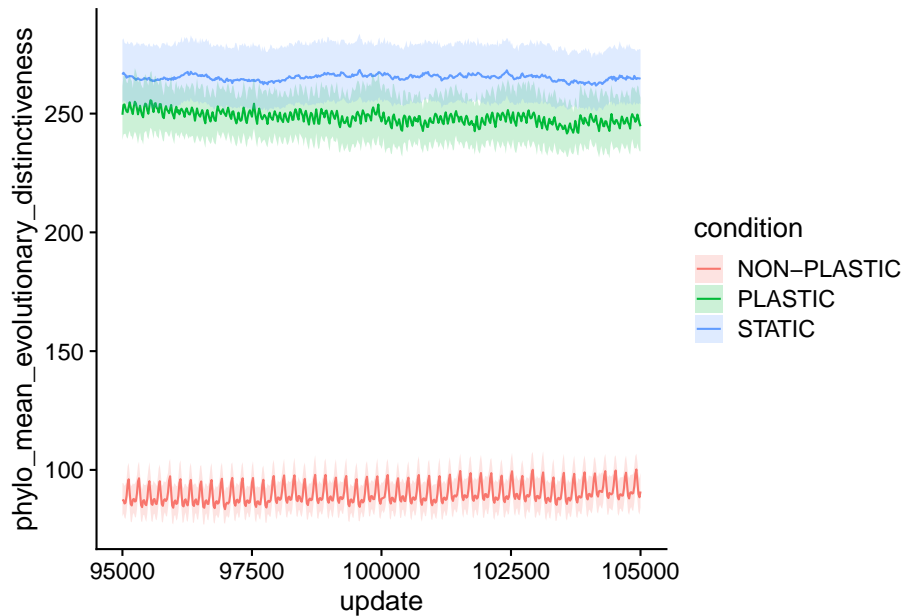

3.14   Camera-ready figures

Figures styled for the paper.

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
```

```r
  ) +
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"
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
theme(
  legend.position="none"
) +
coord_flip() +
ggsave(
  paste0(working_directory, "plots/", "mutation-accumulation.pdf"),
  width=5,
  height=4
)
```

Phenotypic volatility

```r
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"
  ) +
  theme(
    legend.position="none",
    axis.ticks.y=element_blank(),
    axis.text.y=element_blank(),
    axis.title.y=element_blank()
  ) +
  coord_flip() +
  ggsave(
    paste0(working_directory, "plots/", "phenotypic-volatility.pdf"),
    width=4,
    height=4
  )
```

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
  ) +
```

```r
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"
) +
scale_fill_brewer(
  palette="Paired"
) +
scale_color_brewer(
  palette="Paired"
) +
theme(
  legend.position="none",
  axis.ticks.y=element_blank(),
  axis.text.y=element_blank(),
  axis.title.y=element_blank()
) +
coord_flip() +
ggsave(
  paste0(working_directory, "plots/", "selective-sweeps.pdf"),
  width=4,
  height=4
)
```

All together

```r
grid <- plot_grid(
  mutation_count_fig,
  phenotypic_volatility_fig,
  selective_sweeps_fig,
  nrow=1,
```

```r
  align="v",
  labels="auto"
)
save_plot(
  paste0(working_directory, "plots/", "evolutionary-dynamics.pdf"),
  grid,
  base_height=6,
  base_asp=2.5
)
```

Chapter 4

Effect of phenotypic plasticity on the evolution and maintenance of complex features

4.1 Overview

```r
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)
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          0.3
## year           2020
## month          10
## day            10
## svn rev        79318
```

```
## language        R
## version.string R version 4.0.3 (2020-10-10)
## nickname        Bunny-Wunnies Freak Out
```

4.3   Setup

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

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

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(
```

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

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

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

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

4.4 Evolution of phenotypic plasticity

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

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

```
## `summarise()` has grouped output by 'sensors', 'env_label', 'condition'. You can override usin
```

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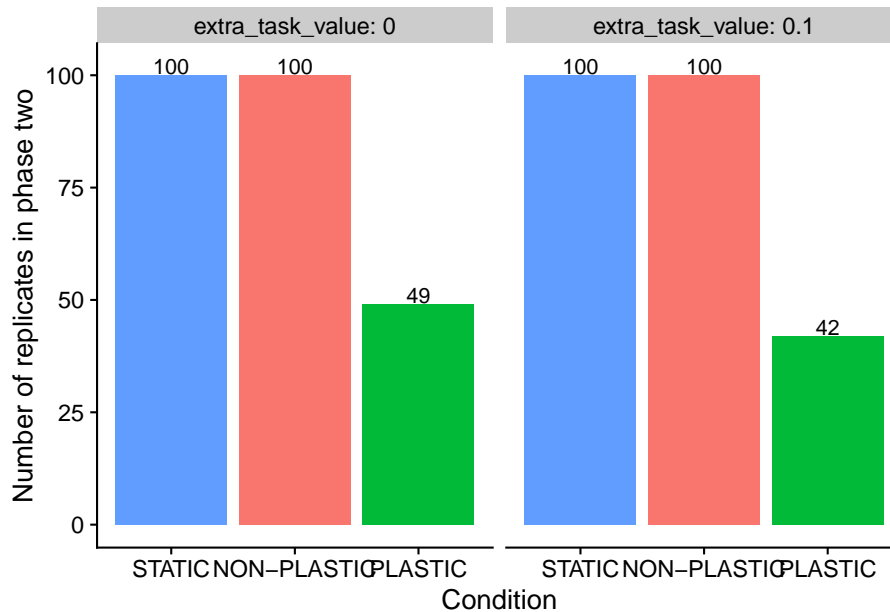
We can confirm our expectation that the dominant genotypes in non-plastic conditions are not phenotypically plastic.

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

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

```
ggplot(filter(summary_data_group_counts, is_plastic), aes(x=condition, y=n, fill=condit
  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") +
  facet_wrap(~extra_task_value, labeller=label_both) +
  theme(
    legend.position="none"
  )
```
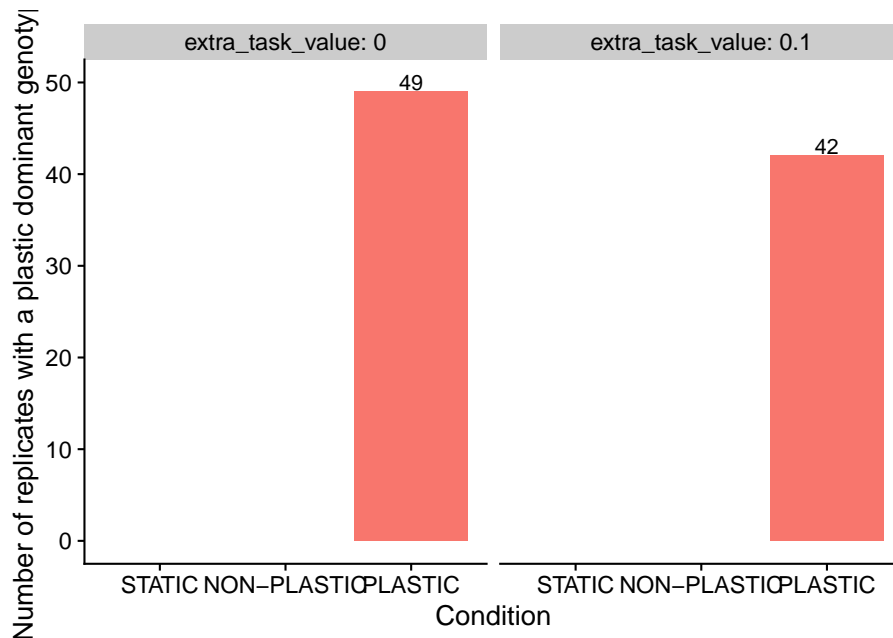

4.5 Final dominant task performance

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

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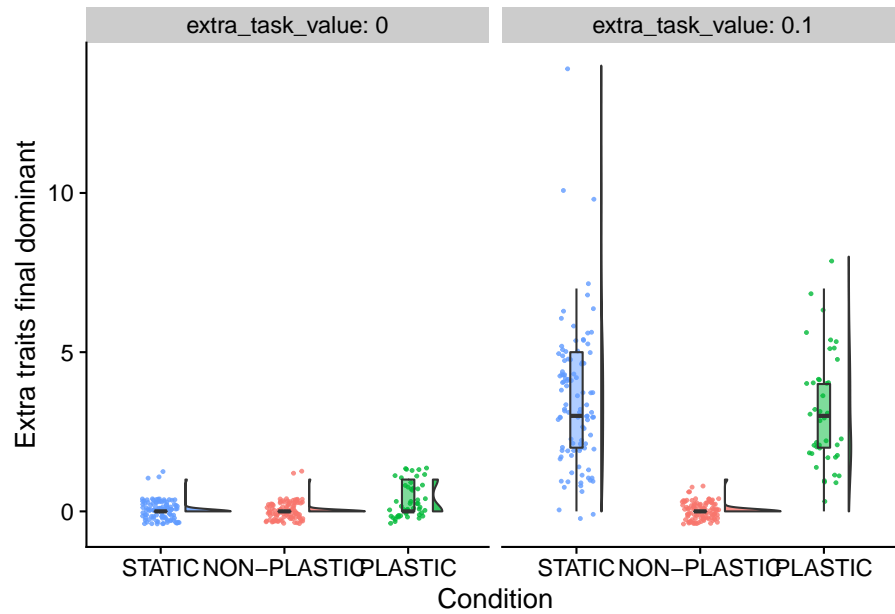
```
reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
kruskal.test(
  formula=dominant_extra_tasks~condition,
  data=dom_task_data
)
```

```
##
##  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
```

```
ggplot(summary_data, aes(x=condition, y=final_pop_extra_tasks_0.05, 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"
  )
```
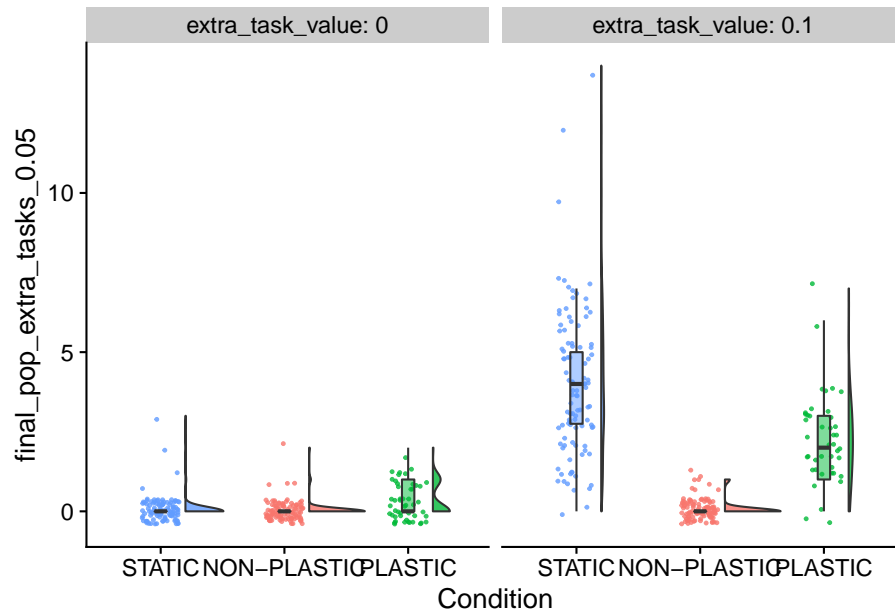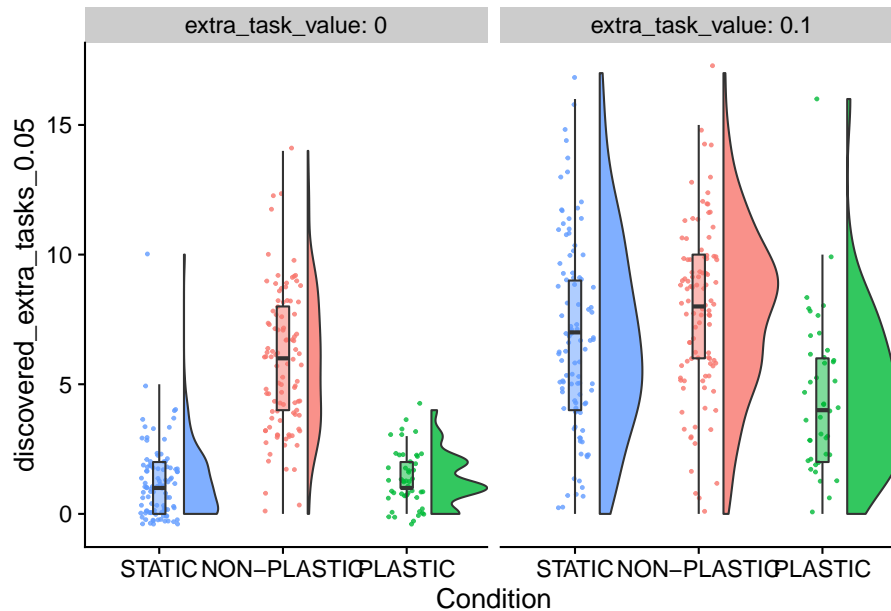
4.7 Population-level tasks discovered

```
ggplot(summary_data, aes(x=condition, y=discovered_extra_tasks_0.05, 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"
)
```


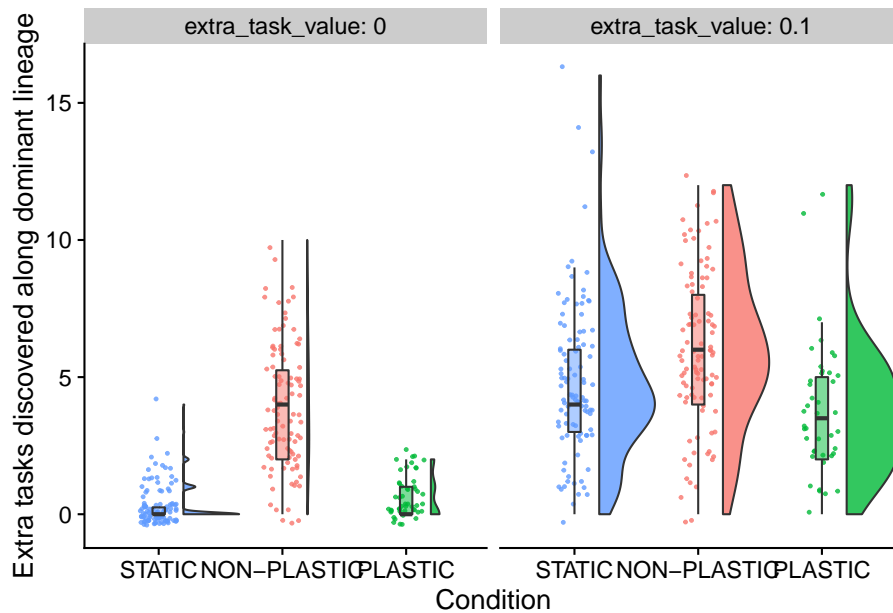## 4.8 Extra tasks along lineage of final dominant genotype

4.8.1 Tasks discovered

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

```
geom_boxplot(
  width = .1,
  outlier.shape = NA,
  alpha = 0.5
) +
scale_x_discrete(
  name="Condition",
  limits=condition_order
) +
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
)
```



```
reward_level <- 0.1
dom_task_data <- filter(summary_data, extra_task_value==reward_level)
```

```r
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
```

```r
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
```
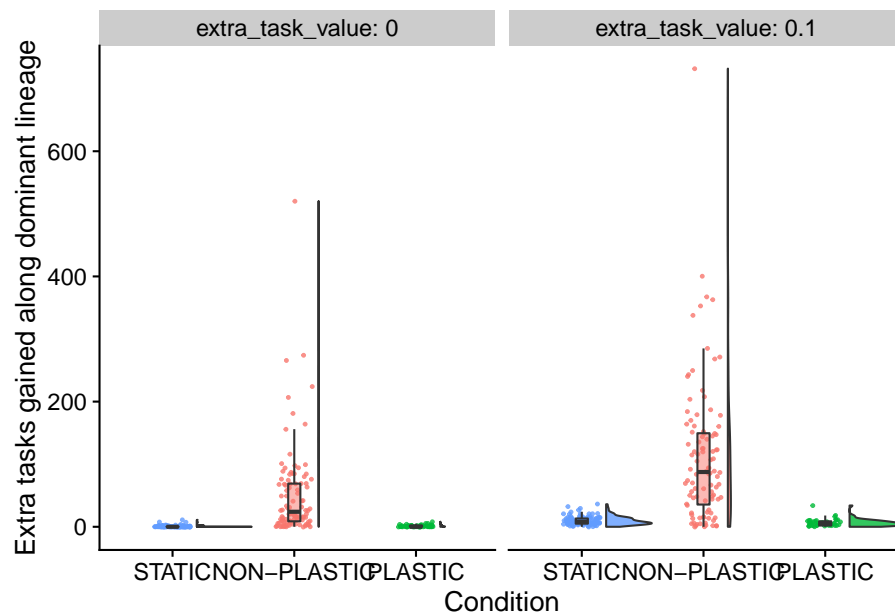
4.8.2 Tasks gained

```r
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,
```

```r
    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
)
```


4.8.3 Tasks lost

```r
ggplot(summary_data, aes(x=condition, y=dominant_lineage_extra_traits_lost, fill=condit
  geom_flat_violin(
```

```r
    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(
  ~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
)
```
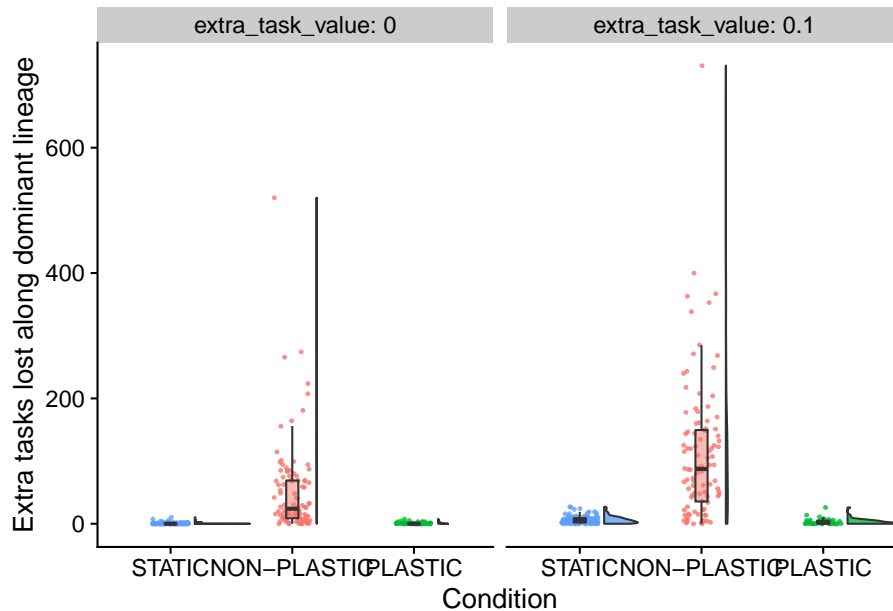
4.9 Extra task performance over time

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_
  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
```

```
)
```
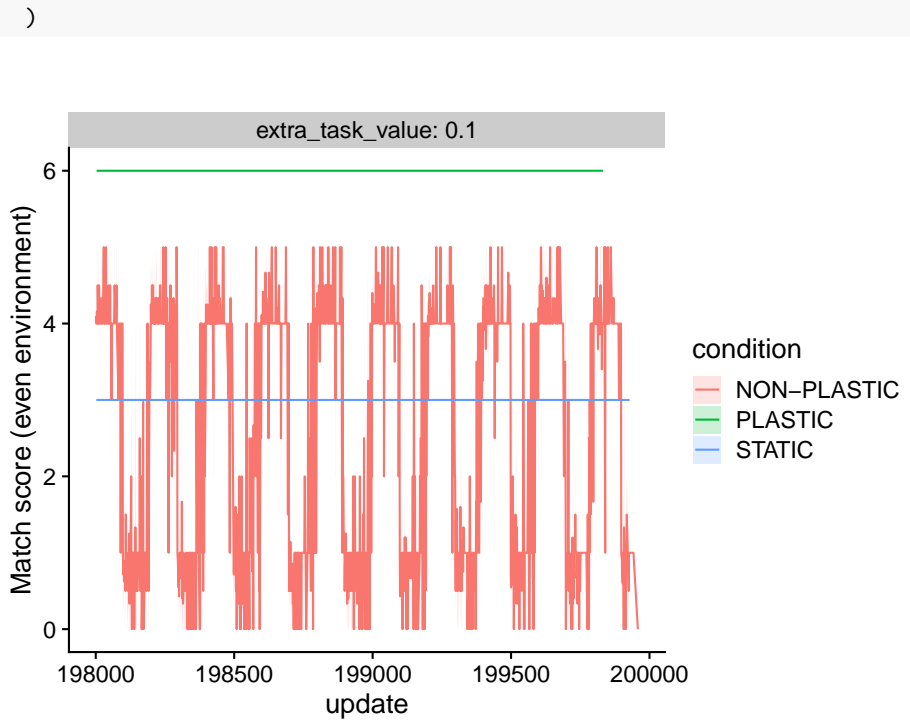

Extra tasks over time

```
ggplot(filter(lineage_reward10, update>198000 & update<=200000), aes(x=update, y=extra_traits, co
  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
  )
```
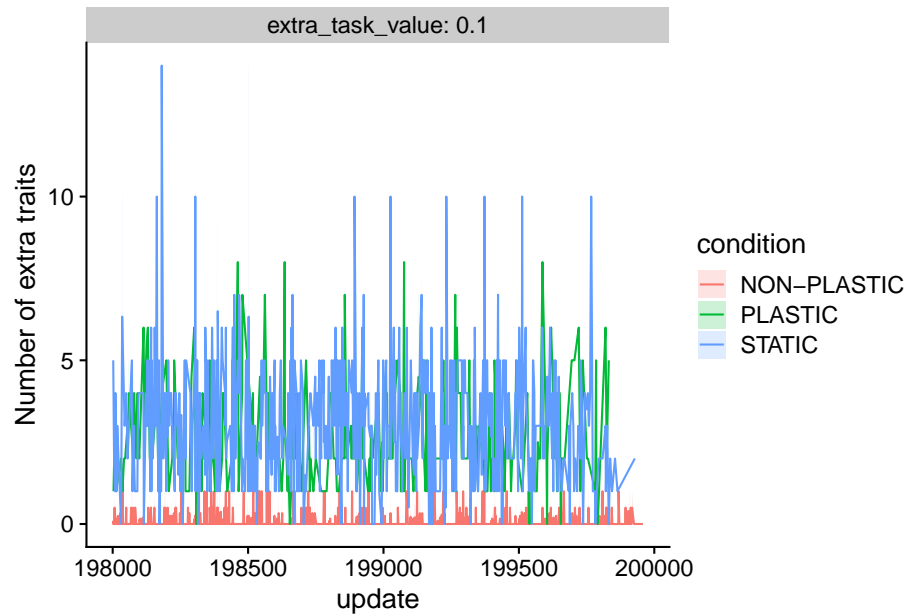
4.10 Manuscript figures

Figures for the manuscript.

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()
```

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
  ) +
```

```r
  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()
```

Final dominant lineage tasks lost.

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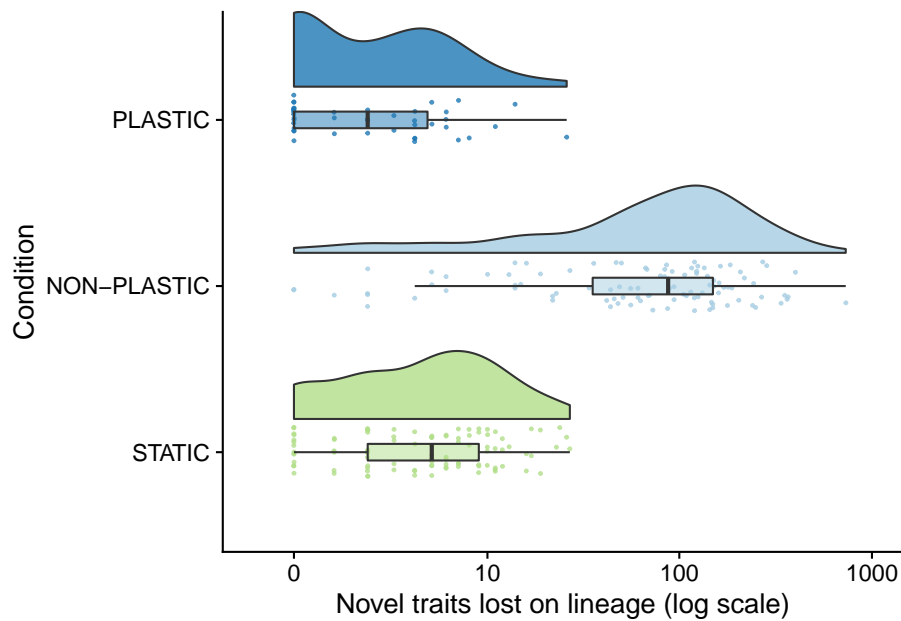
```
  scale_fill_brewer(
    palette="Paired"
  ) +
  scale_color_brewer(
    palette="Paired"
  ) +
  theme(
    legend.position="none"
  ) +
  coord_flip()

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(),a
  nrow=1,
  align="v",
  labels="auto"
)
save_plot(
  paste0(working_directory, "plots/", "complex-traits-panel.pdf"),
```

```
    grid,
    base_height=6,
    base_asp=2.5
)
```