Supplemental Material for An Exploration of Exploration: Measuring the ability of lexicase selection to find obscure pathways

Jose Guadalupe Hernandez, Alexander Lalejini, and Charles Ofria

2021-06-21

Contents

Chapter 1

Introduction

This is the supplemental material associated with our 2021 GPTP contribution entitled, *An Exploration of Exploration: Measuring the ability of lexicase selection to find obscure pathways*. Preprint forthcoming.

1.1 About our supplemental material

This supplemental material is hosted on GitHub using GitHub pages. The source code and configuration files used to generate this supplemental material can be found in this GitHub repository. We compiled our data analyses and supplemental documentation into this nifty web-accessible book using bookdown.

Our supplemental material includes the following:

- TODO

1.2 Contributing authors

- Jose Guadalupe Hernandez
- Alexander Lalejini
- Charles Ofria

1.3 Research overview

Abstract:

TODO

Chapter 2

Data Availability

2.1 Source code

The source code for this work is available on GitHub at https://github.com/jgh9094/GPTP-2021-Exploration-Of-Exploration.

2.2 Experimental results

The data from our experiments are available online in an OSF repository (Lalejini and Hernandez, 2021) at https://osf.io/xpjft/.

Chapter 3

Compile and run experiments

Here, we provide a guide to compiling and running our experiments using our Docker image.

Please file an issue on GitHub if something is unclear or does not work.

3.1 Docker

TODO

3.1.1 Getting the right image

3.1.1.1 DockerHub

3.1.1.2 Local build

3.1.2 Spinning up a container

3.1.3 Running inside the container

3.1.4 Copying content from the container

Chapter 4

Tournament selection vs Lexicase selection

4.1 Overview

```
# Relative location of data.
working_directory <- "experiments/2021-05-27-tournament/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"

# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

4.2 Analysis dependencies

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

These analyses were conducted in 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          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

4.3 Setup

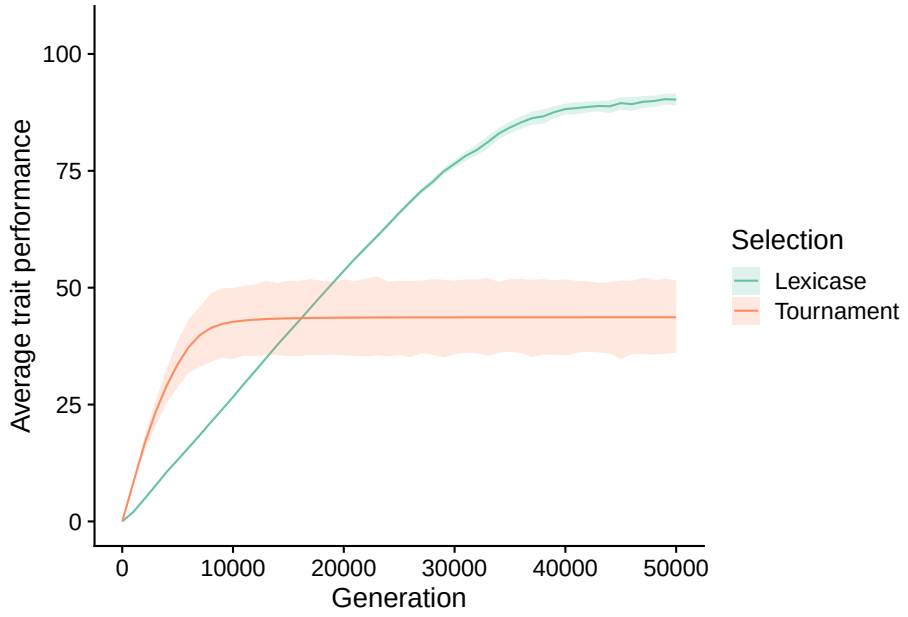
```
data_loc <- paste0(
  working_directory,
  "data/timeseries-res-1000g.csv"
)

data <- read.csv(data_loc, na.strings="NONE")

data$selection_name <- factor(
  data$selection_name,
  levels=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament")
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

# Labeler for stats annotations
p_label <- function(p_value) {
  threshold = 0.0001
```

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

# Significance threshold
alpha <- 0.05

# Common graph variables
performance_ylim <- 105
coverage_ylim <- 1.0

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

4.4 Exploration diagnostic performance

```
elite_ave_performance_fig <- ggplot(
    data,
    aes(
      x=gen,
      y=elite_trait_avg,
      color=selection_name,
      fill=selection_name
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, performance_ylim)
  ) +
  scale_x_continuous(
    name="Generation"
  ) +
```

```
  scale_fill_brewer(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
    palette=cb_palette
  )
elite_ave_performance_fig
```



4.5 Final Performance

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(elite_trait_avg ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
```

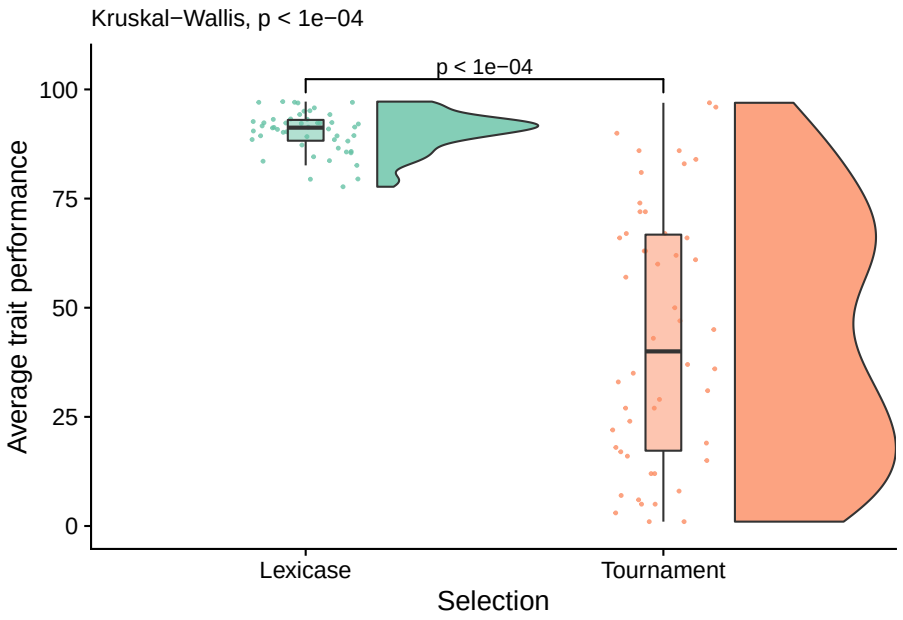
```
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(
      x=selection_name,
      y=elite_trait_avg,
      fill=selection_name
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=selection_name),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, performance_ylim)
  ) +
  scale_x_discrete(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
  ) +
  scale_color_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("Lexicase", "Tournament"),
```

```
    labels=c("Lexicase", "Tournament"),
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(
        signif(
          kruskal.test(
            formula=elite_trait_avg~selection_name,
            data=final_data)$p.value,digits=4
        )
      )
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(
      xmin=group1,
      xmax=group2,
      annotations=label,
      y_position=manual_position
    ),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  )
```

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

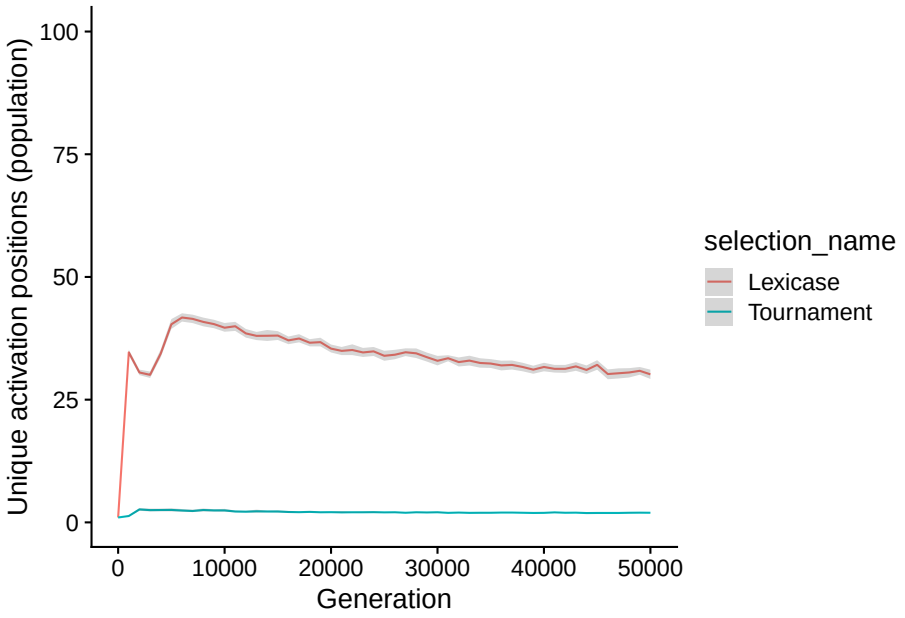
```
elite_final_performance_fig
```



.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.signif	y.position	groups
elite_trait_avg	Lexicase	Tournament	50	50	2357	0	0	****	97.477	Lexicase , T

4.6 Unique starting positions

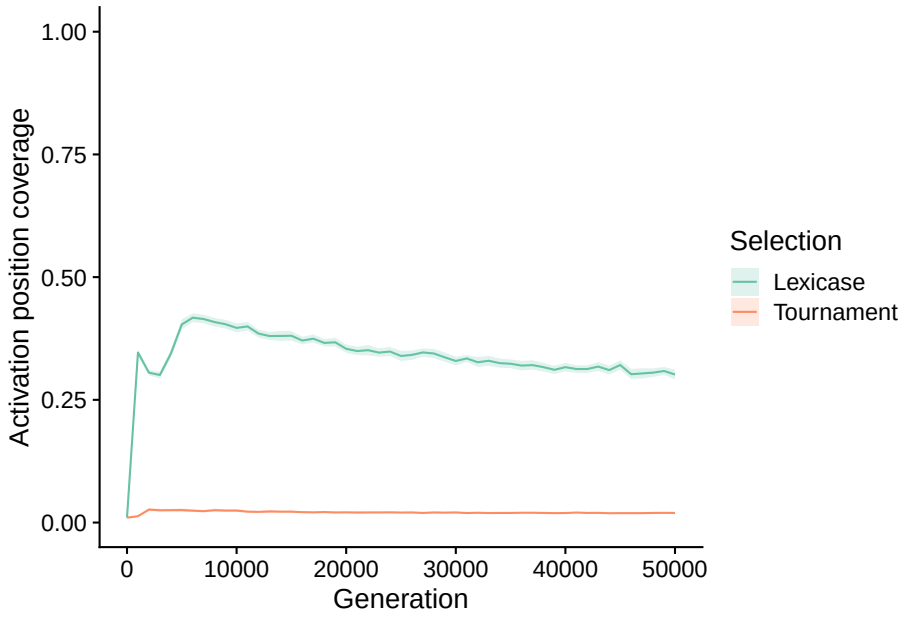
```
ggplot(data, aes(x=gen, y=uni_str_pos, color=selection_name)) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Unique activation positions (population)",
    limits=c(0, 100)
  ) +
  scale_x_continuous(
    name="Generation"
  )
```



Different cardinalities have numbers of possible starting positions, so next, we look at the proportion of starting positions (out of all possible) maintained by populations.

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
      x=gen,
      y=unique_start_positions_coverage,
      color=selection_name,
      fill=selection_name
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limit=c(0, coverage_ylim)
  ) +
```

```
  scale_x_continuous(
    name="Generation"
  ) +
  scale_fill_brewer(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



4.6.1 Final starting position Coverage

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(unique_start_positions_coverage ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
```

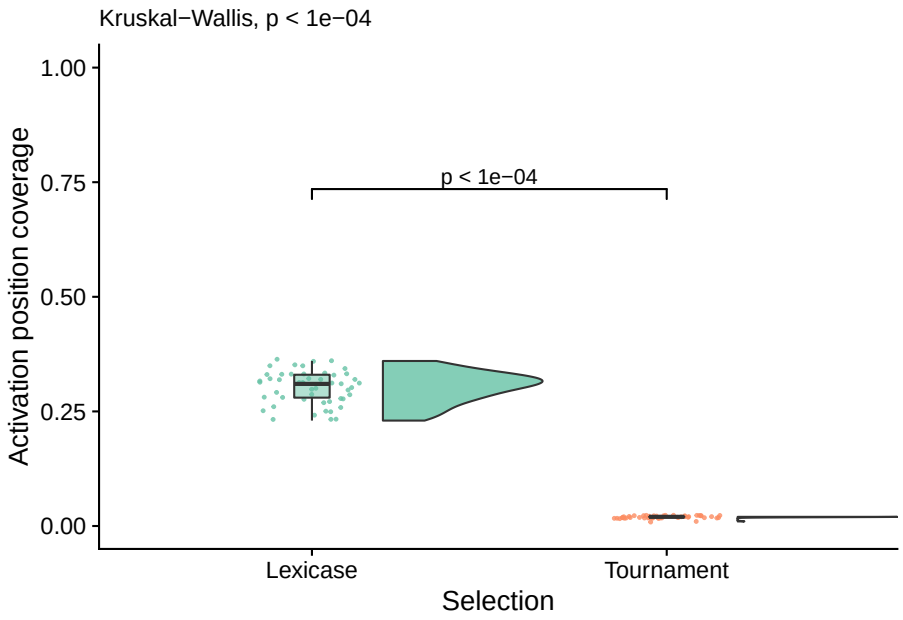
```
  add_xy_position(x="selection_name",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=selection_name,
      y=unique_start_positions_coverage,
      fill=selection_name
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=selection_name),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, coverage_ylim)
  ) +
  scale_x_discrete(
    name="Selection",
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
  ) +
  scale_color_brewer(
    name="Selection",
```

```
    palette=cb_palette,
    limits=c("Lexicase", "Tournament"),
    labels=c("Lexicase", "Tournament"),
  ) +
  labs(
    subtitle=paste0(
      "Kruskal-Wallis, ",
      p_label(
        signif(
          kruskal.test(
            formula=unique_start_positions_coverage~selection_name,
            data=final_data)$p.value,digits=4
        )
      )
    )
  ) +
  ggsignif::geom_signif(
    data=filter(stat.test, p.adj <= alpha),
    aes(
      xmin=group1,
      xmax=group2,
      annotations=label,
      y_position=manual_position
    ),
    manual=TRUE,
    inherit.aes=FALSE
  ) +
  theme(
    legend.position="none"
  )
```

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

```
unique_start_positions_coverage_final_fig
```

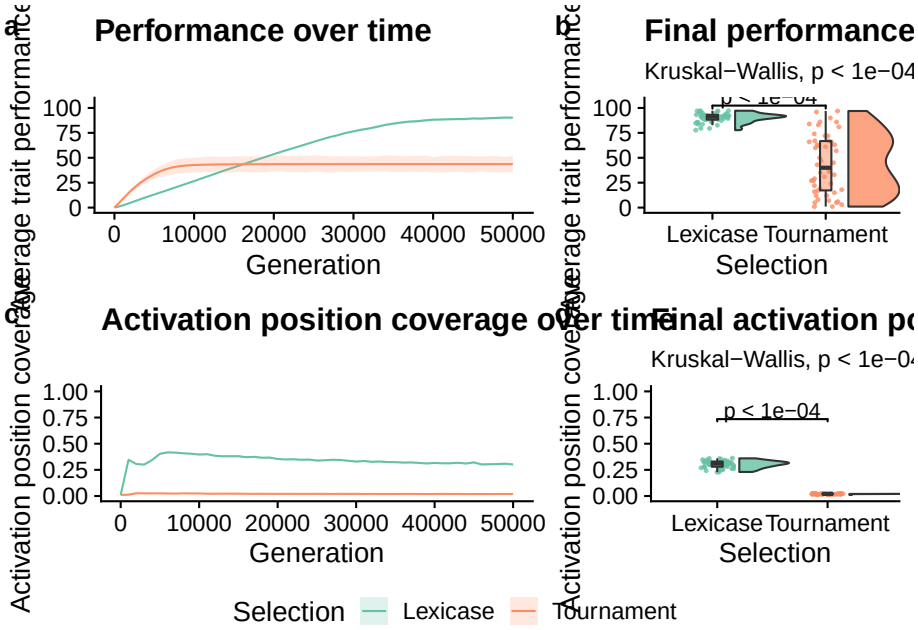


.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.si
unique_start_positions_coverage	Lexicase	Tournament	50	50	2500	0	0	****

4.7 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    labs(subtitle="") +
    theme(legend.position="none"),
  elite_final_performance_fig +
```

```
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    labs(subtitle="") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(
    working_directory,
    "imgs/tournament-vs-lexicase-panel.pdf",
    sep=""
  ),
  grid,
  base_width=12,
  base_height=8
)

grid
```


a
Performance over time
Average trait performance
100
75
50
25
0
0
10000
20000
30000
40000
50000
Generation
b
Final performance
Kruskal−Wallis, p < 1e−04
p < 1e−04
Lexicase
Tournament
Selection
c
Activation position coverage over time
Activation position coverage
1.00
0.75
0.50
0.25
0.00
Generation
d
Final activation pc
Kruskal−Wallis, p < 1e−0
p < 1e−04
Selection
Lexicase
Tournament

Chapter 5

Diagnostic cardinality

5.1 Overview

```
# Relative location of data.
working_directory <-
  "experiments/2021-05-27-cardinality/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

5.2 Analysis dependencies

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

These analyses were conducted in 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          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

5.3 Setup

```
data_loc <- paste0(
  working_directory,
  "data/timeseries-res-1000g.csv"
)
data <- read.csv(
  data_loc,
  na.strings="NONE"
)

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT

data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, gen==max(data$gen))
```

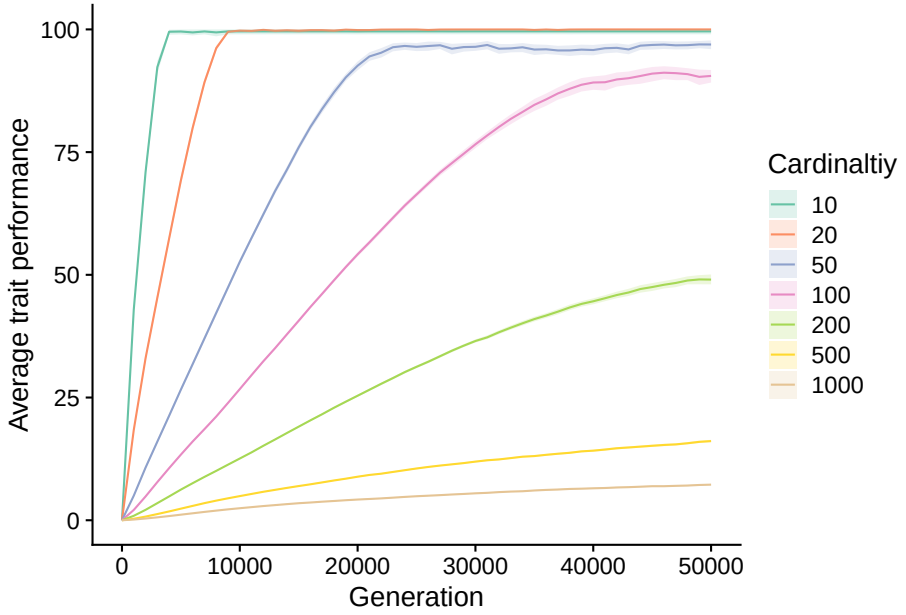
```
# Labeler for stats annotations
p_label <- function(p_value) {
  threshold = 0.0001
  if (p_value < threshold) {
    return(paste0("p < ", threshold))
  } else {
    return(paste0("p = ", p_value))
  }
}

# Significance threshold
alpha <- 0.05

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

5.4 Exploration diagnostic performance

First, we look at performance over time. Specifically, we look at the normalized aggregage score of the most performant individuals over time. To control for different cardinalities having different maximum scores, we normalized performances (by dividing by cardinality) to values between 0 and 100.

```
elite_trait_ave_fit <- ggplot(
    data,
    aes(
      x=gen,
      y=elite_trait_avg,
      color=cardinality,
      fill=cardinality
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
```

```
  scale_x_continuous(
    name="Generation"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  )
elite_trait_ave_fit
```

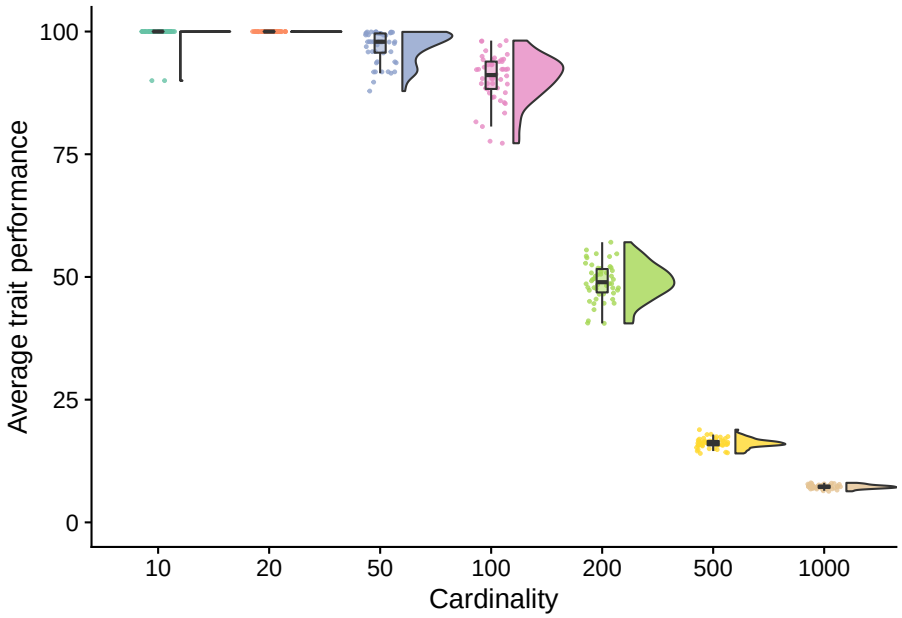


5.4.1 Final performance

Next, we look only at the final performances of each treatment

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(elite_trait_avg ~ cardinality) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="cardinality",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
elite_trait_ave_fit_final <- ggplot(
    final_data,
    aes(x=cardinality, y=elite_trait_avg, fill=cardinality)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=cardinality),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Cardinality"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_trait_ave_fit_final
```

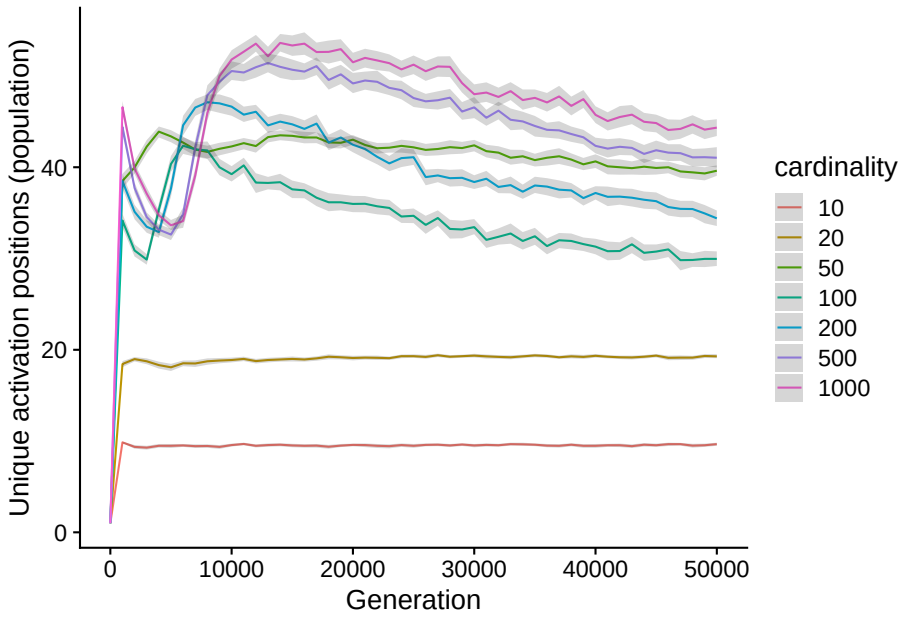


.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.signif	y.position	group
elite_trait_avg	10	20	50	50	2399	0	0	****	191.9440	10, 20
elite_trait_avg	10	50	50	50	2404	0	0	****	288.4852	10, 50
elite_trait_avg	10	100	50	50	2438	0	0	****	385.0264	10 , 1
elite_trait_avg	10	200	50	50	2500	0	0	****	481.5676	10 , 2
elite_trait_avg	10	500	50	50	2500	0	0	****	578.1088	10 , 5
elite_trait_avg	10	1000	50	50	2500	0	0	****	674.6500	10 , 1
elite_trait_avg	20	50	50	50	2500	0	0	****	771.1912	20, 50
elite_trait_avg	20	100	50	50	2500	0	0	****	867.7324	20 , 1
elite_trait_avg	20	200	50	50	2500	0	0	****	964.2736	20 , 2
elite_trait_avg	20	500	50	50	2500	0	0	****	1060.8148	20 , 5
elite_trait_avg	20	1000	50	50	2500	0	0	****	1157.3560	20 , 1
elite_trait_avg	50	100	50	50	2166	0	0	****	1253.8972	50 , 1
elite_trait_avg	50	200	50	50	2500	0	0	****	1350.4384	50 , 2
elite_trait_avg	50	500	50	50	2500	0	0	****	1446.9796	50 , 5
elite_trait_avg	50	1000	50	50	2500	0	0	****	1543.5208	50 , 1
elite_trait_avg	100	200	50	50	2500	0	0	****	1640.0620	100, 2
elite_trait_avg	100	500	50	50	2500	0	0	****	1736.6032	100, 5
elite_trait_avg	100	1000	50	50	2500	0	0	****	1833.1444	100 ,
elite_trait_avg	200	500	50	50	2500	0	0	****	1929.6856	200, 5
elite_trait_avg	200	1000	50	50	2500	0	0	****	2026.2268	200 ,
elite_trait_avg	500	1000	50	50	2500	0	0	****	2122.7680	500 ,

5.5 Unique starting positions

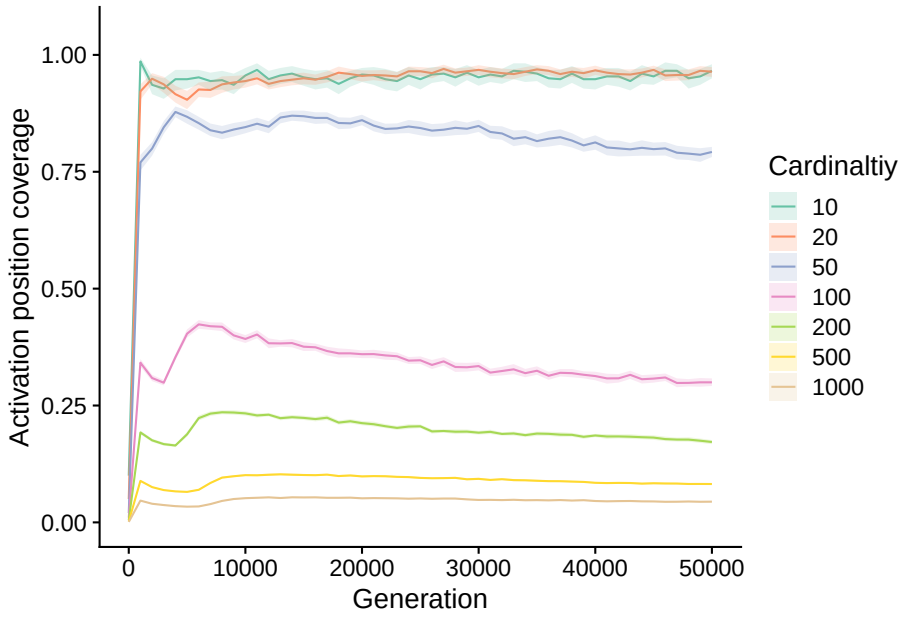
Next, we analyze the number of unique starting position maintained by populations.

```
ggplot(data, aes(x=gen, y=uni_str_pos, color=cardinality)) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Unique activation positions (population)",
  ) +
  scale_x_continuous(
    name="Generation"
  )
```



Different cardinalities have numbers of possible starting positions, so next, we look at the proportion of starting positions (out of all possible) maintained by populations.

```
unique_start_positions_coverage_fig <- ggplot(
    data,
    aes(
      x=gen,
      y=unique_start_positions_coverage,
      color=cardinality,
      fill=cardinality
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.05)
  ) +
  scale_x_continuous(
    name="Generation"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  )
unique_start_positions_coverage_fig
```

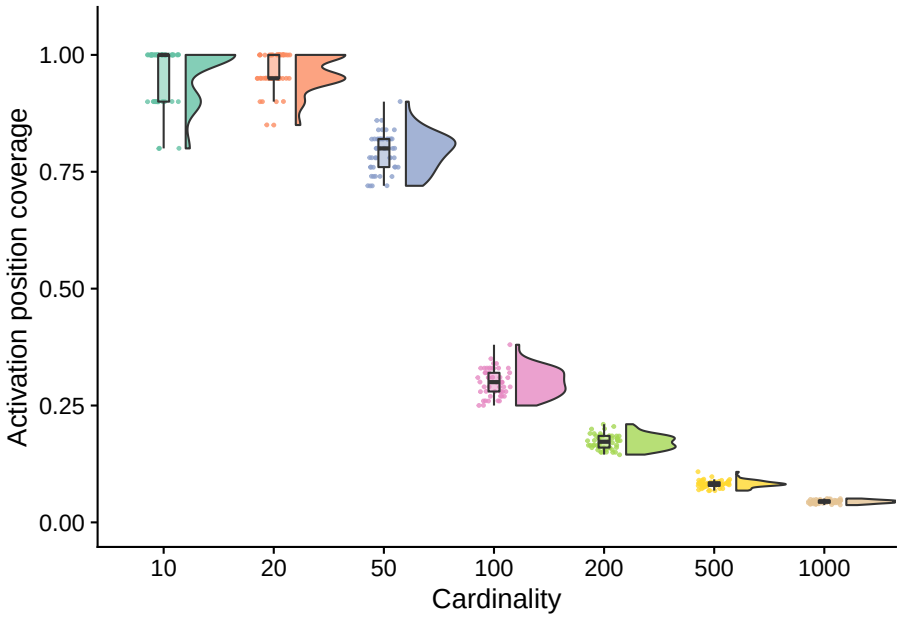


5.5.1 Final starting position coverage

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(unique_start_positions_coverage ~ cardinality) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="cardinality",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
final_unique_start_positions_coverage_fig <- ggplot(
    final_data,
    aes(
      x=cardinality,
      y=unique_start_positions_coverage,
      fill=cardinality
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
```

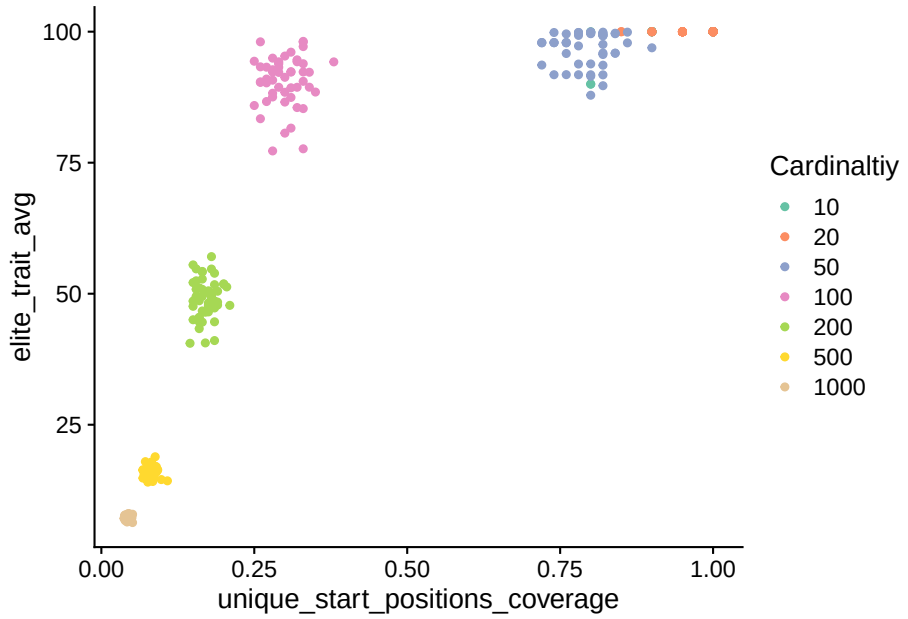
```
  geom_point(
    mapping=aes(color=cardinality),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.05)
  ) +
  scale_x_discrete(
    name="Cardinality"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
final_unique_start_positions_coverage_fig
```



.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.signif	y.positio
unique_start_positions_coverage	10	20	50	50	1448	0.126	1	ns	1.9490
unique_start_positions_coverage	10	50	50	50	2424	0.000	0	****	2.9454
unique_start_positions_coverage	10	100	50	50	2500	0.000	0	****	3.9419
unique_start_positions_coverage	10	200	50	50	2500	0.000	0	****	4.9383
unique_start_positions_coverage	10	500	50	50	2500	0.000	0	****	5.9348
unique_start_positions_coverage	10	1000	50	50	2500	0.000	0	****	6.9312
unique_start_positions_coverage	20	50	50	50	2492	0.000	0	****	7.9277
unique_start_positions_coverage	20	100	50	50	2500	0.000	0	****	8.9241
unique_start_positions_coverage	20	200	50	50	2500	0.000	0	****	9.9206
unique_start_positions_coverage	20	500	50	50	2500	0.000	0	****	10.9170
unique_start_positions_coverage	20	1000	50	50	2500	0.000	0	****	11.9135
unique_start_positions_coverage	50	100	50	50	2500	0.000	0	****	12.9099
unique_start_positions_coverage	50	200	50	50	2500	0.000	0	****	13.9064
unique_start_positions_coverage	50	500	50	50	2500	0.000	0	****	14.9028
unique_start_positions_coverage	50	1000	50	50	2500	0.000	0	****	15.8993
unique_start_positions_coverage	100	200	50	50	2500	0.000	0	****	16.8957
unique_start_positions_coverage	100	500	50	50	2500	0.000	0	****	17.8922
unique_start_positions_coverage	100	1000	50	50	2500	0.000	0	****	18.8886
unique_start_positions_coverage	200	500	50	50	2500	0.000	0	****	19.8851
unique_start_positions_coverage	200	1000	50	50	2500	0.000	0	****	20.8815
unique_start_positions_coverage	500	1000	50	50	2500	0.000	0	****	21.8780

5.6 Does activation position coverage predict performance?

```
ggplot(
    final_data,
    aes(
      x=unique_start_positions_coverage,
      y=elite_trait_avg,
      color=cardinality
    )
  ) +
  geom_point() +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  )
```




```
cor.test(
  x=final_data$unique_start_positions_coverage,
  y=final_data$elite_trait_avg,
  method="spearman",
  exact=FALSE
)
```

```
##
```

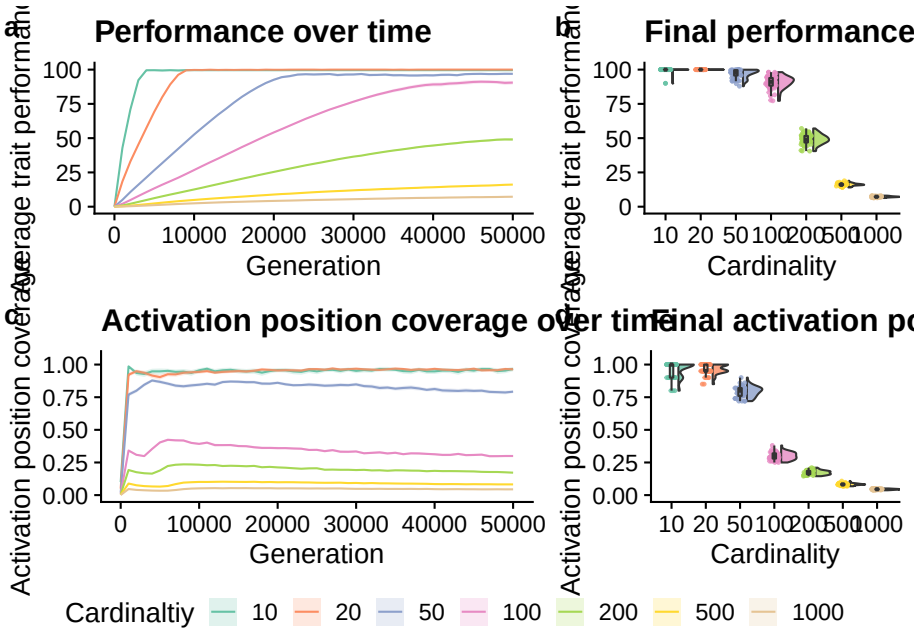
```
##  Spearman's rank correlation rho
##
## data:  final_data$unique_start_positions_coverage and final_data$elite_trait_avg
## S = 262488, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9632668
```

5.7 Manuscript figures

Combine figures for the manuscript.

```
legend <- cowplot::get_legend(
    elite_trait_ave_fit +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_trait_ave_fit +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_trait_ave_fit_final +
    ggtitle("Final performance") +
    theme(),
  unique_start_positions_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  final_unique_start_positions_coverage_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)
```

```
grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(working_directory, "imgs/cardinality-panel.pdf", sep=""),
  grid,
  base_width=12,
  base_height=8
)

grid
```



Chapter 6

Increasing population size versus increasing generations

6.1 Overview

```
# Relative location of data.
working_directory <-
  "experiments/2021-06-03-cardinality-pop-size/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"

# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

6.2 Analysis dependencies

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

These analyses were conducted in 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          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

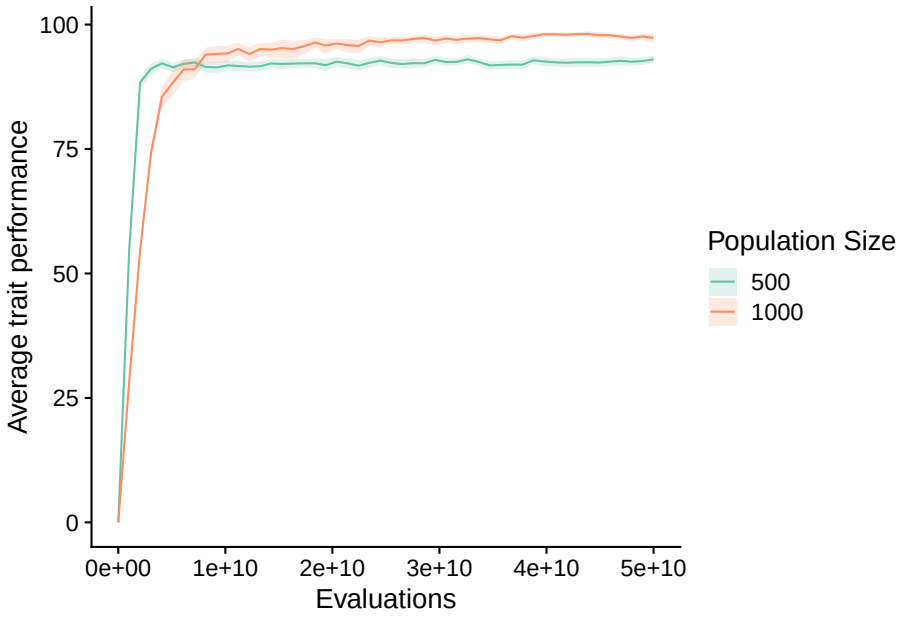
6.3 Setup

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

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

data$POP_SIZE <- as.factor(
  data$POP_SIZE
)

data <- filter(data, cardinality=="100") # These runs finished.
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT
```

```
final_data <- filter(data, evaluations==max(data$evaluations))

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

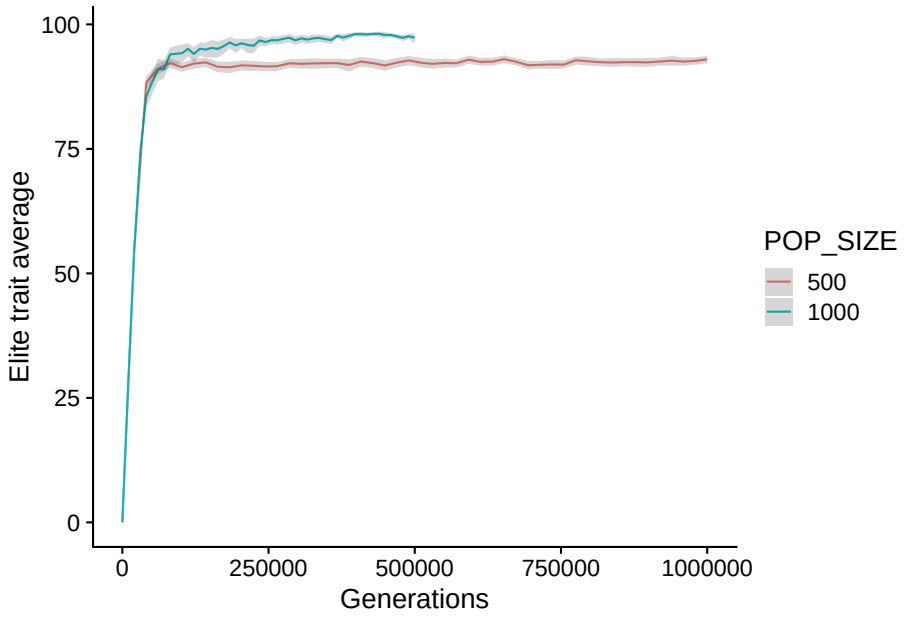
6.4 Exploration diagnostic performance

```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(
      x=evaluations,
      y=elite_trait_avg,
      color=POP_SIZE,
      fill=POP_SIZE
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Population Size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Population Size",
    palette=cb_palette
  )
elite_ave_performance_fig
```



Same as above, but by generations instead of evaluations.

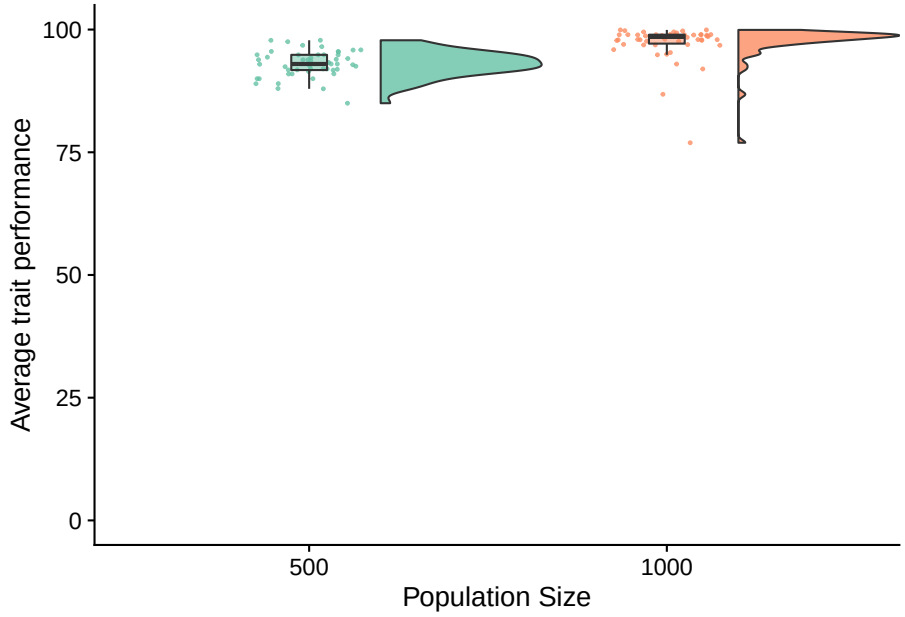
```
ggplot(data, aes(x=gen, y=elite_trait_avg, color=POP_SIZE)) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Elite trait average"
  ) +
  scale_x_continuous(
    name="Generations"
  )
```



6.4.1 Final performance

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(x=POP_SIZE, y=elite_trait_avg, fill=POP_SIZE)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=POP_SIZE),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
```

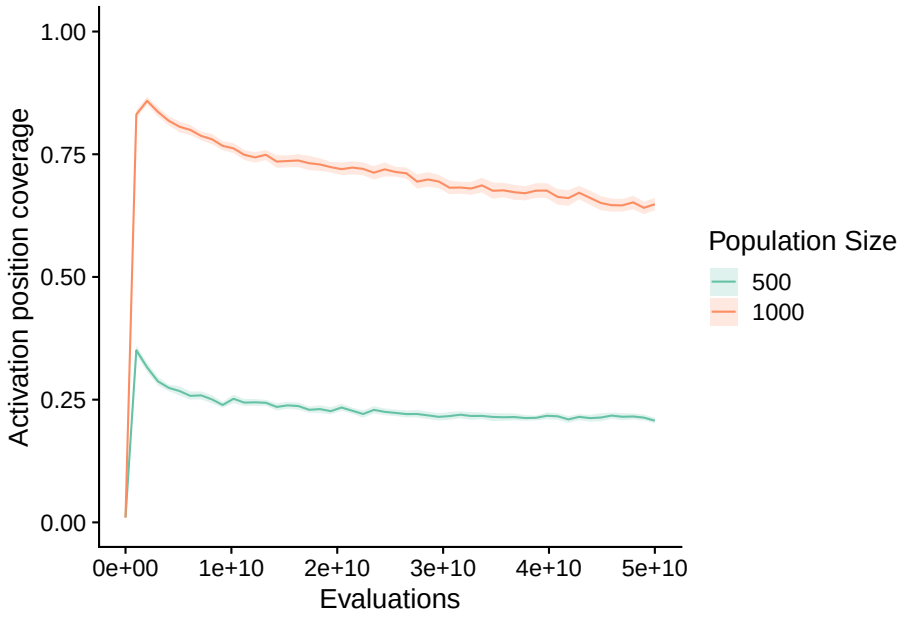
```
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Population Size"
  ) +
  scale_fill_brewer(
    name="Population Size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Population Size",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```



6.5 Unique starting positions

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
```

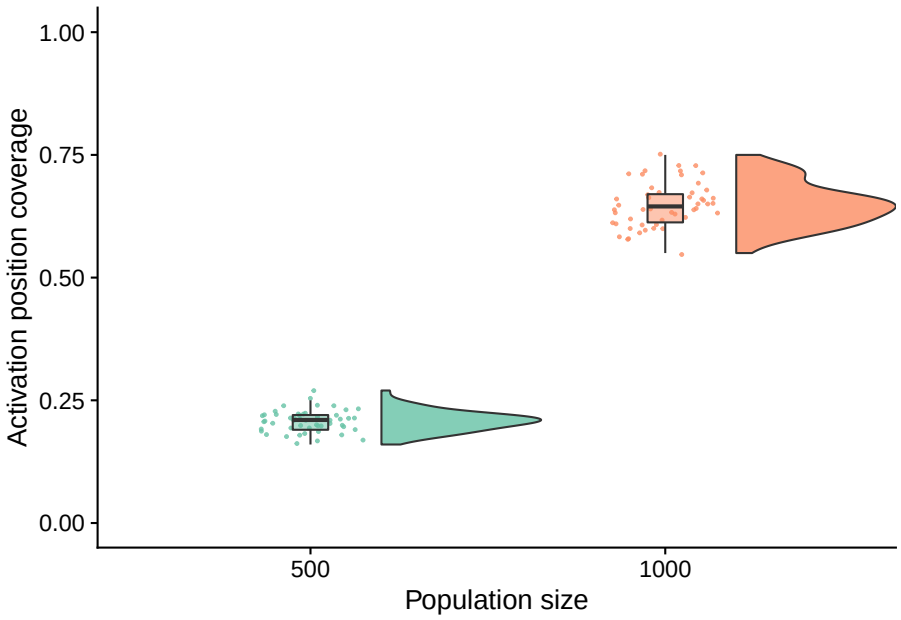
```
      x=evaluations,
      y=unique_start_positions_coverage,
      color=POP_SIZE,
      fill=POP_SIZE
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Population Size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Population Size",
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



6.5.1 Final starting position coverage

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=POP_SIZE,
      y=unique_start_positions_coverage,
      fill=POP_SIZE
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=POP_SIZE),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
```

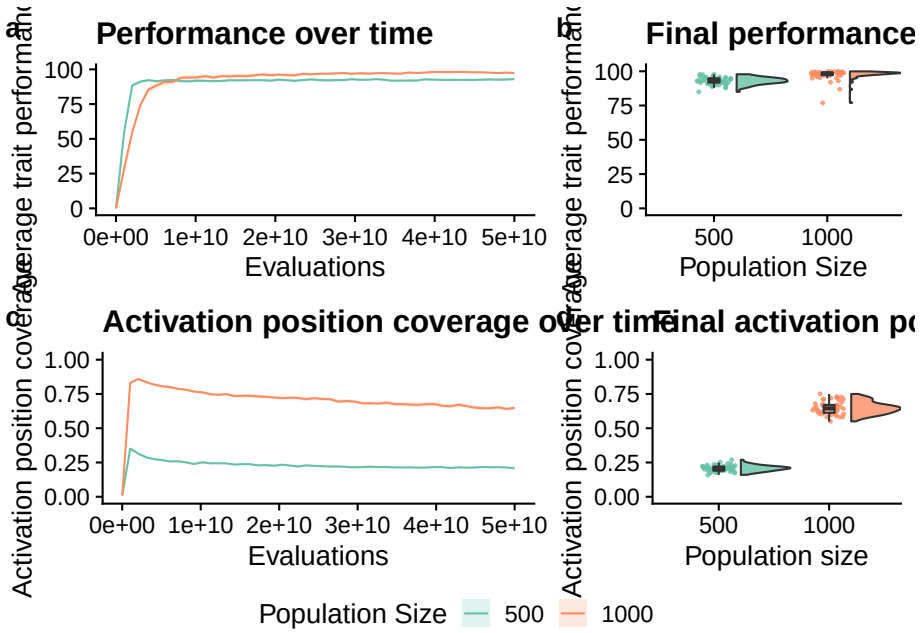
```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="Population size"
  ) +
  scale_fill_brewer(
    name="Population size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Population size",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```



6.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)


save_plot(
  paste(working_directory, "imgs/pop-size-panel.pdf", sep=""),
```

```
  grid,
  base_width=12,
  base_height=8
)

grid
```



Chapter 7

Epsilon lexicase

7.1 Overview

```
# Relative location of data.
working_directory <- "experiments/2021-05-28-epsilon/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

7.2 Analysis dependencies

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

These analyses were conducted in the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
```

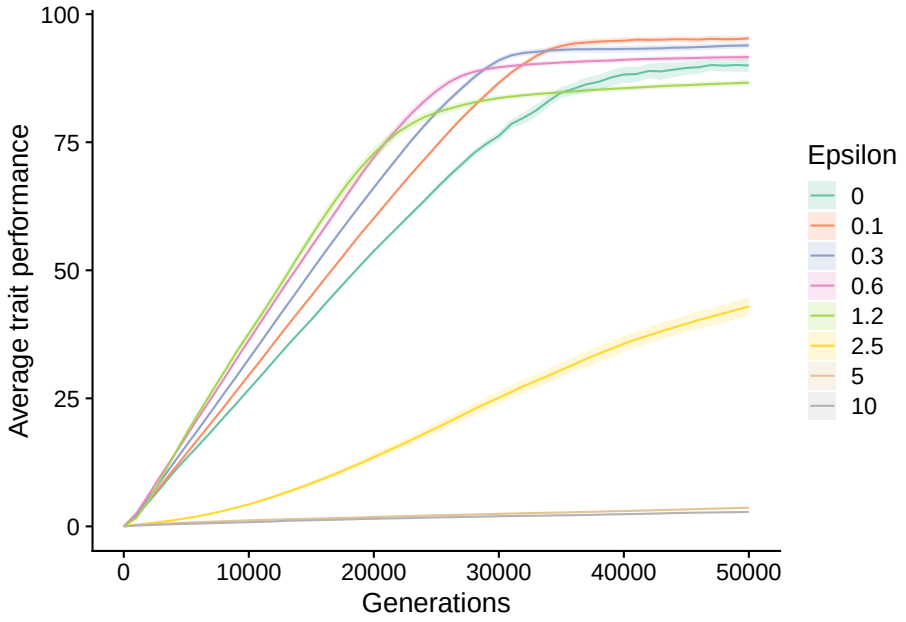
```
## system         x86_64, linux-gnu
## status
## major          4
## minor          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

7.2 Setup

```
data_loc <- paste0(
  working_directory,
  "data/timeseries-res-1000g.csv"
)
data <- read.csv(data_loc, na.strings="NONE")

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

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

7.4 Exploration diagnostic performance

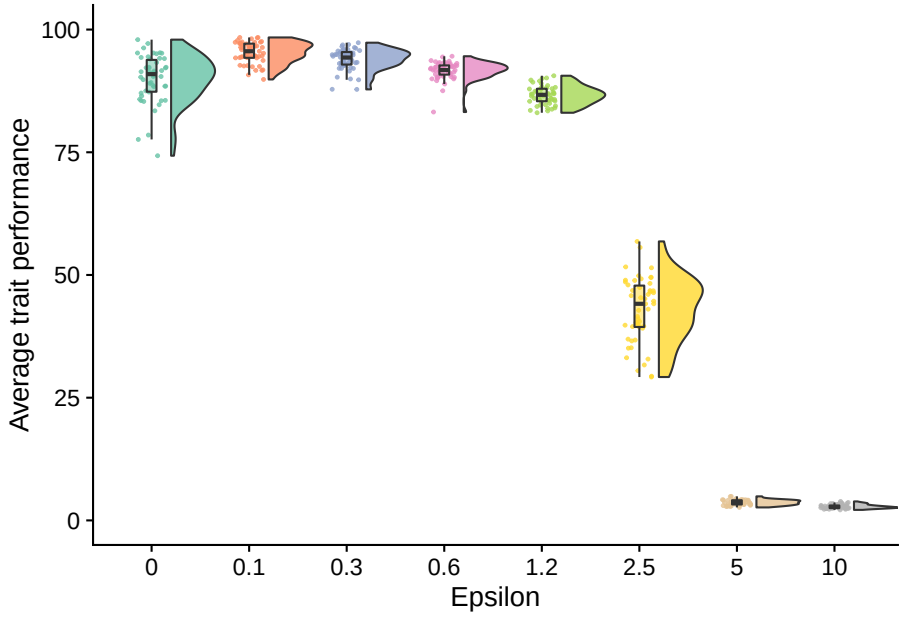
```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(x=gen, y=elite_trait_avg, color=epsilon, fill=epsilon)
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Epsilon",
    palette=cb_palette
  )
elite_ave_performance_fig
```



7.4.1 Final performance

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(x=epsilon, y=elite_trait_avg, fill=epsilon)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=epsilon),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
```

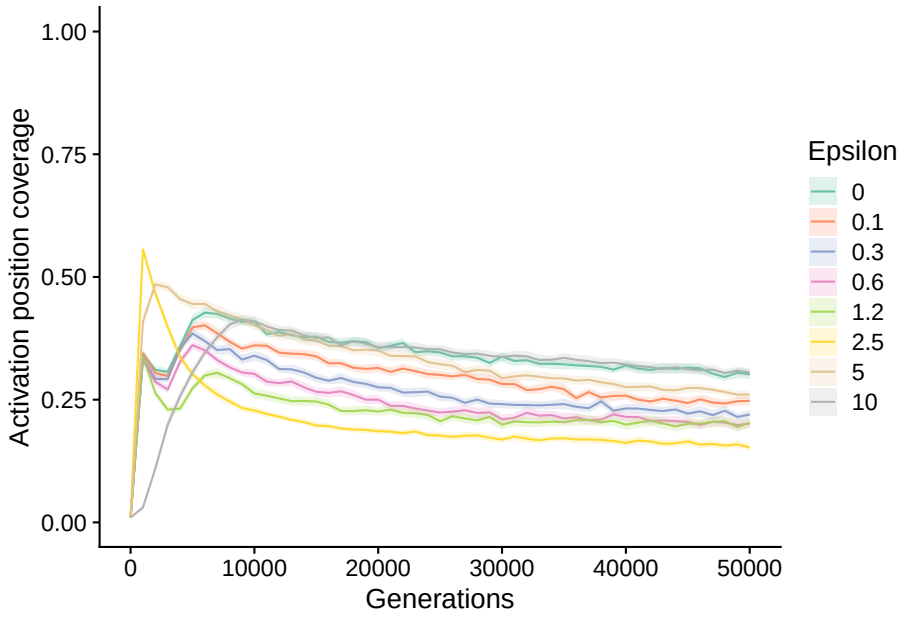
```
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Epsilon"
  ) +
  scale_fill_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```



7.5 Unique starting positions

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
```

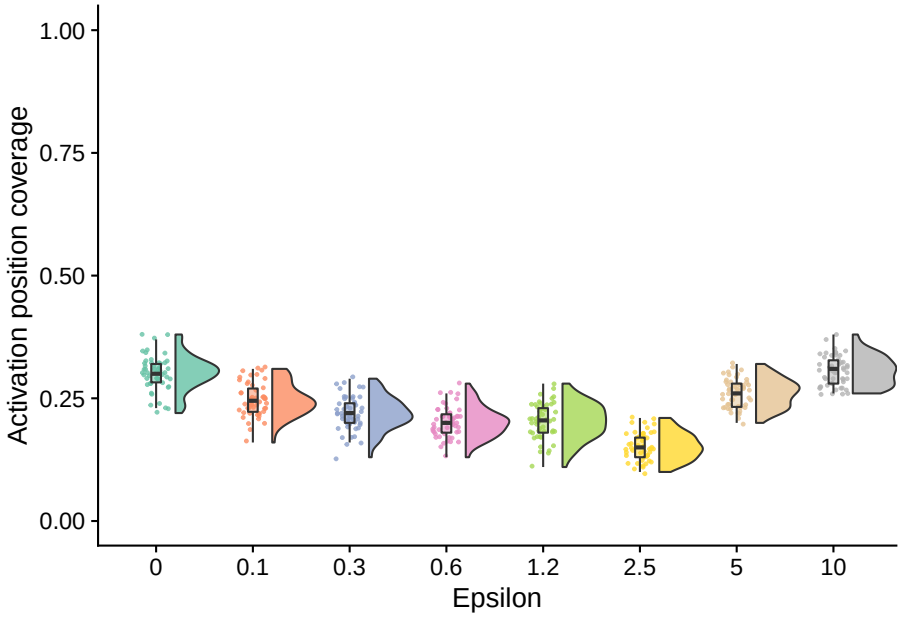
```
      x=gen,
      y=unique_start_positions_coverage,
      color=epsilon,
      fill=epsilon
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Epsilon",
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



7.5.1 Final coverage

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=epsilon,
      y=unique_start_positions_coverage,
      fill=epsilon
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=epsilon),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
```

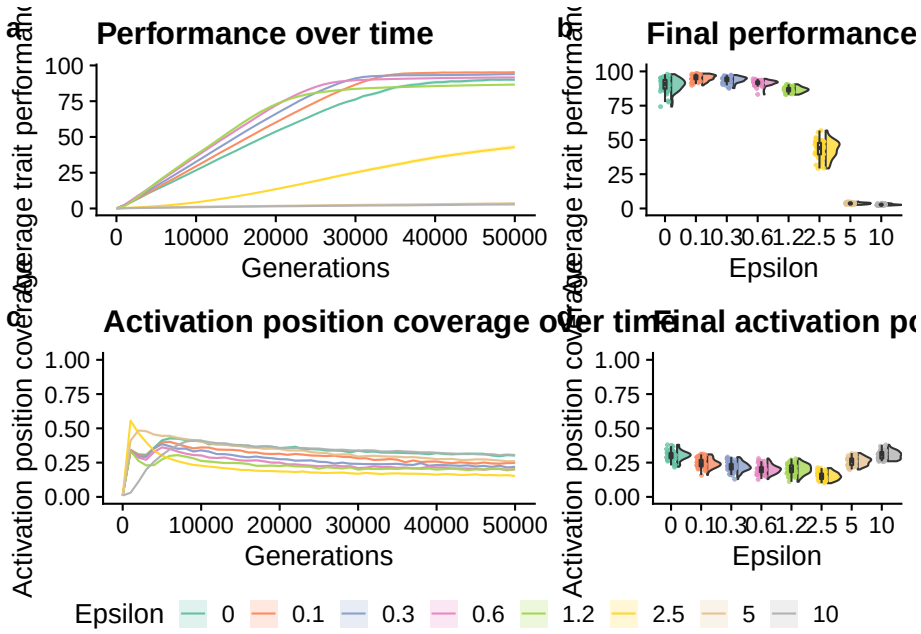
```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="Epsilon"
  ) +
  scale_fill_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Epsilon",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```



7.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(working_directory, "imgs/epsilon-panel.pdf", sep=""),
  grid,
```

```
  base_width=12,
  base_height=8
)

grid
```



Chapter 8

Down-sampled lexicase

8.1 Overview

```
# Relative location of data.
working_directory <- "experiments/2021-05-28-downsampled/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

8.2 Analysis dependencies

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

These analyses were conducted in the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
```

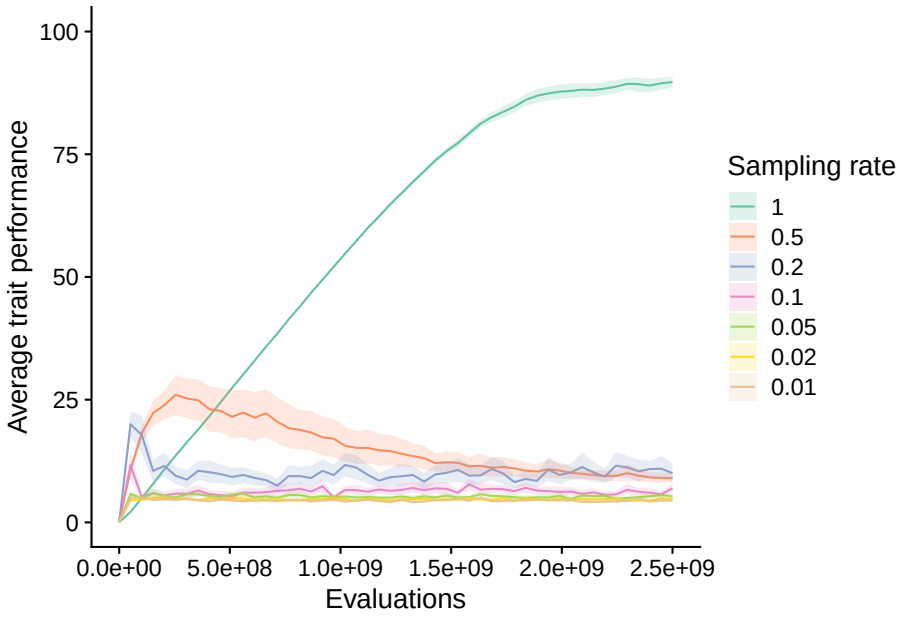
```
## system         x86_64, linux-gnu
## status
## major          4
## minor          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

8.2 Setup

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

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

data$proportion <- factor(
  data$DSLEX_PROP,
  levels=c(1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01)
)

data$elite_trait_avg <- data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <- data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

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

8.4 Exploration diagnostic performance

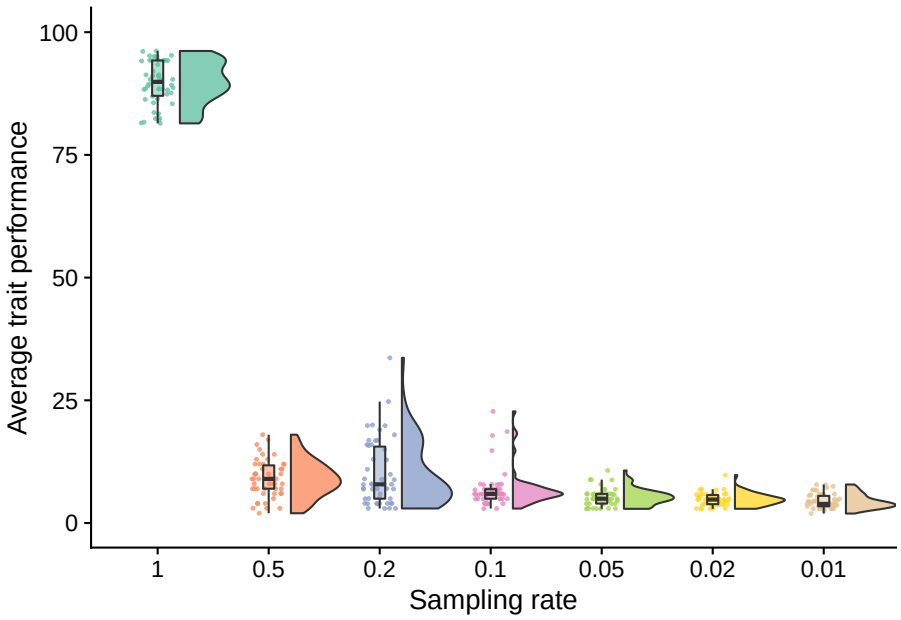
```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(
      x=evaluations,
      y=elite_trait_avg,
      color=proportion,
      fill=proportion
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Sampling rate",
    palette=cb_palette
  )
elite_ave_performance_fig
```



8.4.1 Final performance

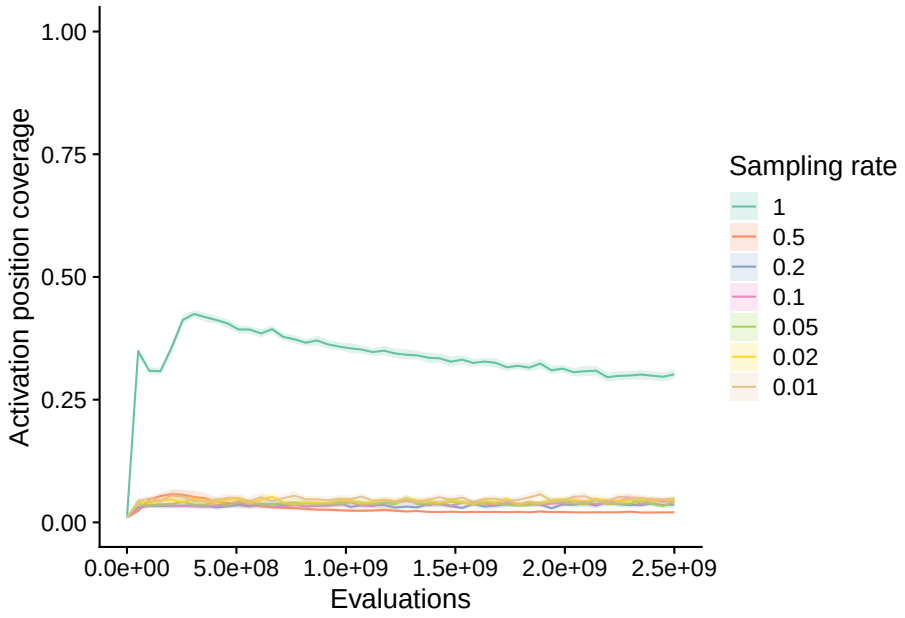
```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(
      x=proportion,
      y=elite_trait_avg,
      fill=proportion
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=proportion),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
```

```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Sampling rate"
  ) +
  scale_fill_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```



8.5 Unique starting positions

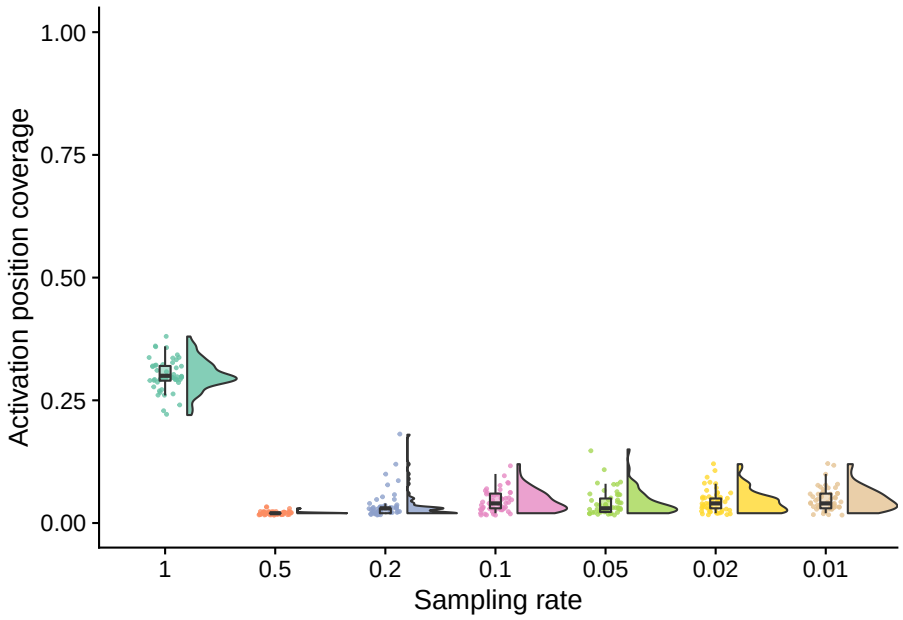
```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
      x=evaluations,
      y=unique_start_positions_coverage,
      color=proportion,
      fill=proportion
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Sampling rate",
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



8.5.1 Final starting position coverage

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=proportion,
      y=unique_start_positions_coverage,
      fill=proportion
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=proportion),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
```

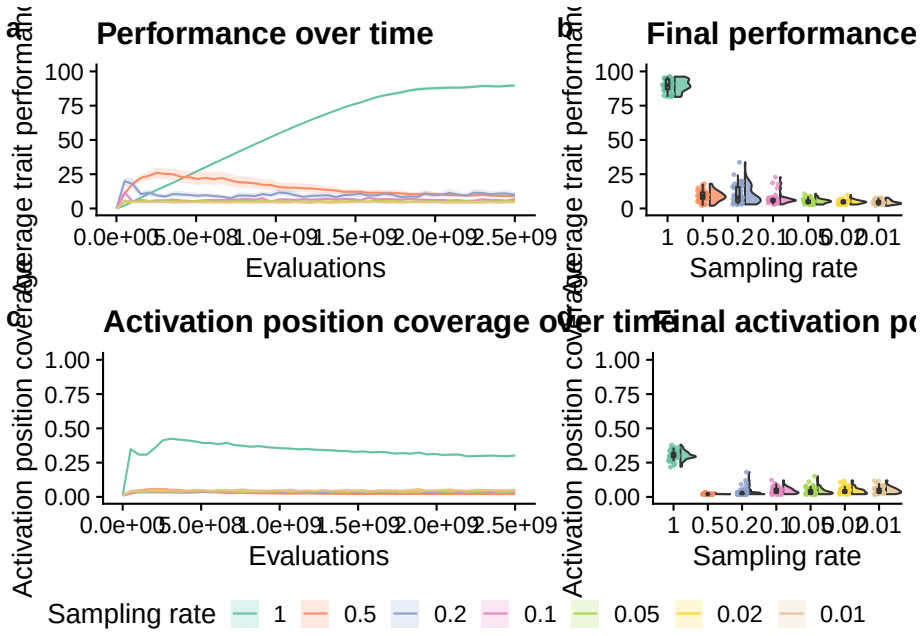
```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="Sampling rate"
  ) +
  scale_fill_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Sampling rate",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```



8.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(working_directory, "imgs/down-sampled-panel.pdf", sep=""),
  grid,
```

```
  base_width=12,
  base_height=8
)

grid
```



Chapter 9

Cohort lexicase

9.1 Overview

```
# Relative location of data.
working_directory <- "experiments/2021-06-01-cohort/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

9.2 Analysis dependencies

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

These analyses were conducted in the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
```

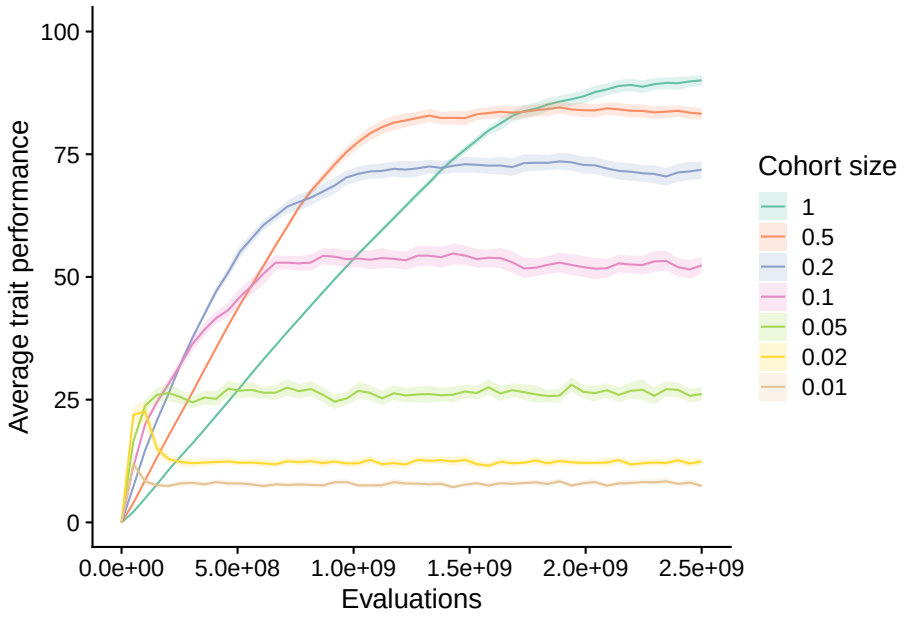
```
## system         x86_64, linux-gnu
## status
## major          4
## minor          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

9.3 Setup

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

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

data$proportion <- factor(
  data$COH_LEX_PROP,
  levels=c(1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01)
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

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

9.4 Exploration diagnostic performance

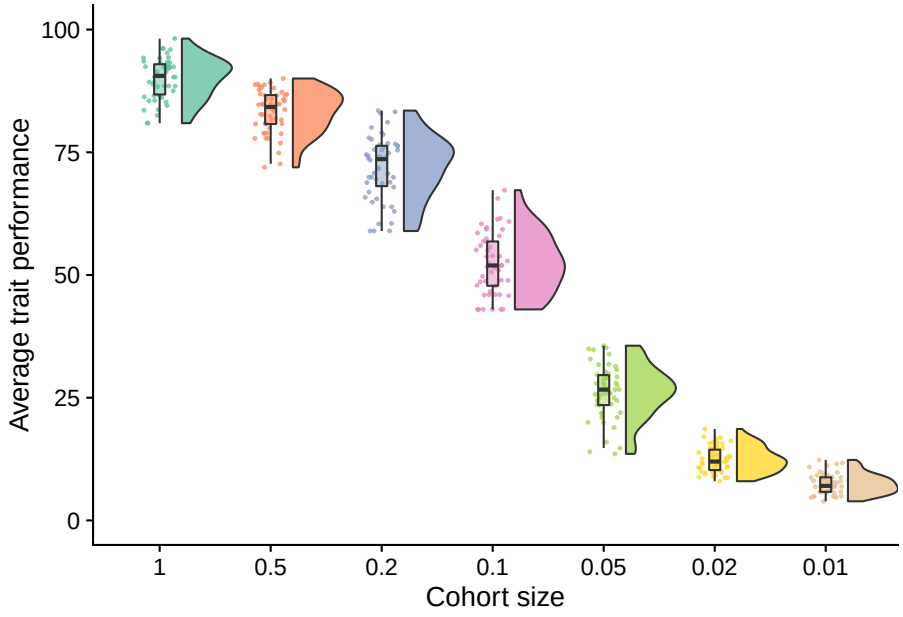
```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(
      x=evaluations,
      y=elite_trait_avg,
      color=proportion,
      fill=proportion
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cohort size",
    palette=cb_palette
  )
elite_ave_performance_fig
```



9.4.1 Final performance

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(x=proportion, y=elite_trait_avg, fill=proportion)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=proportion),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
```

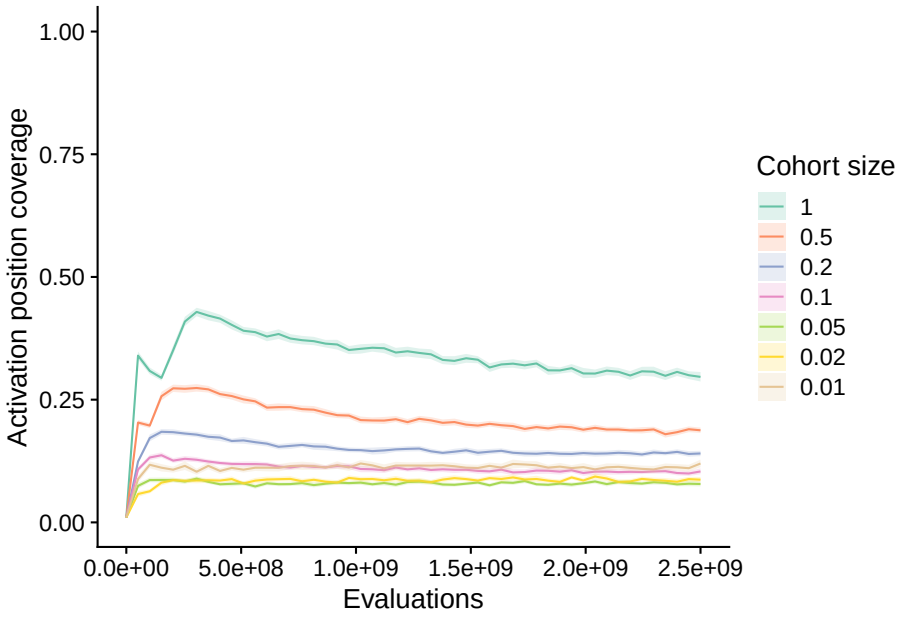
```
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Cohort size"
  ) +
  scale_fill_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```



9.5 Unique starting positions

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
```

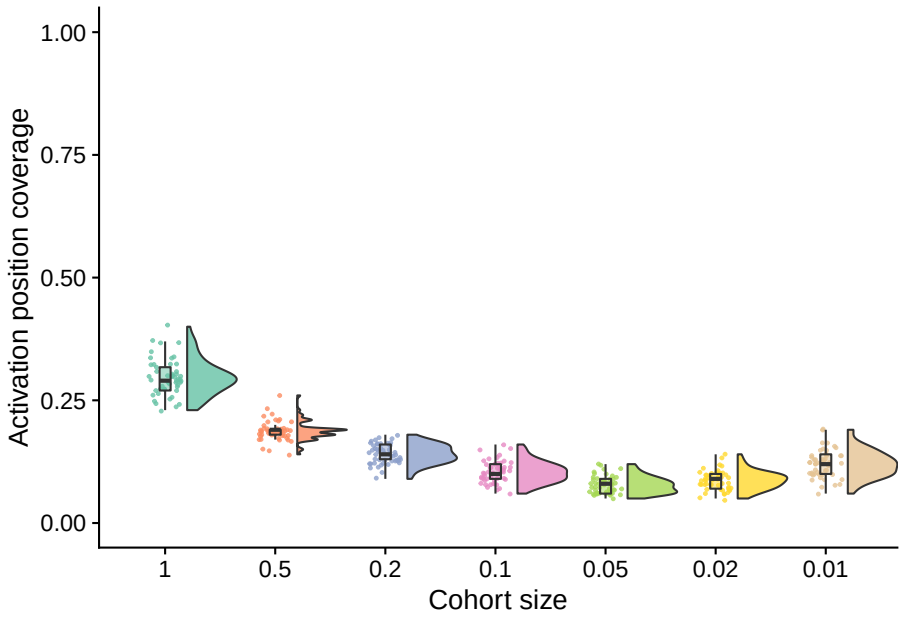
```
      x=evaluations,
      y=unique_start_positions_coverage,
      color=proportion,
      fill=proportion
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Evaluations"
  ) +
  scale_fill_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cohort size",
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



9.5.1 Final starting position coverage

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=proportion,
      y=unique_start_positions_coverage,
      fill=proportion
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=proportion),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
```

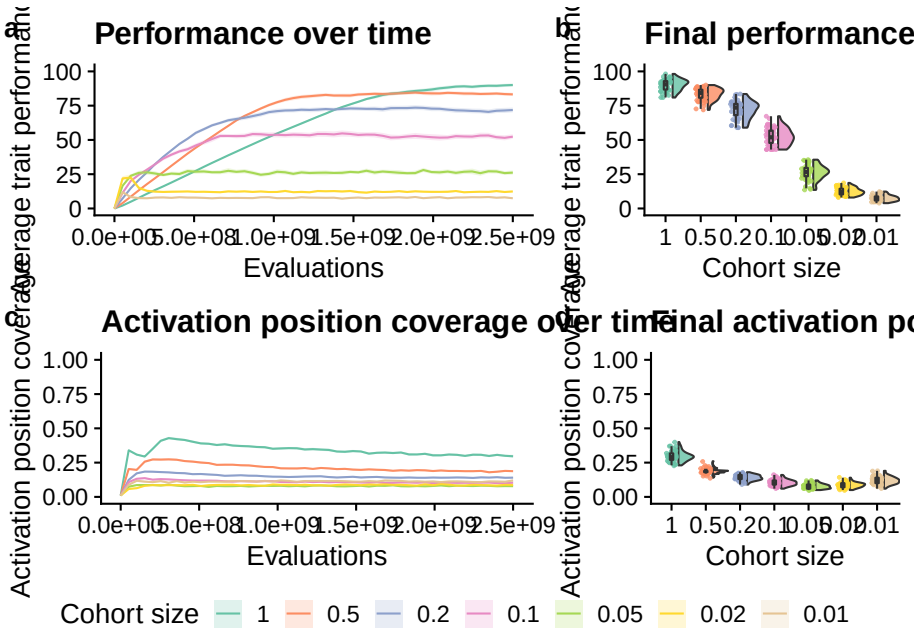
```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="Cohort size"
  ) +
  scale_fill_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cohort size",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```



9.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(working_directory, "imgs/cohort-panel.pdf", sep=""),
  grid,
```

```
  base_width=12,
  base_height=8
)

grid
```



Chapter 10

Down-sampled lexicase versus cohort lexicase

10.1 Overview

```
# Relative location of data.
working_directory <-
  "experiments/2021-06-05-downsample-vs-cohort/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Dark2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

10.2 Analysis dependencies

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

These analyses were conducted in 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          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

10.3 Setup

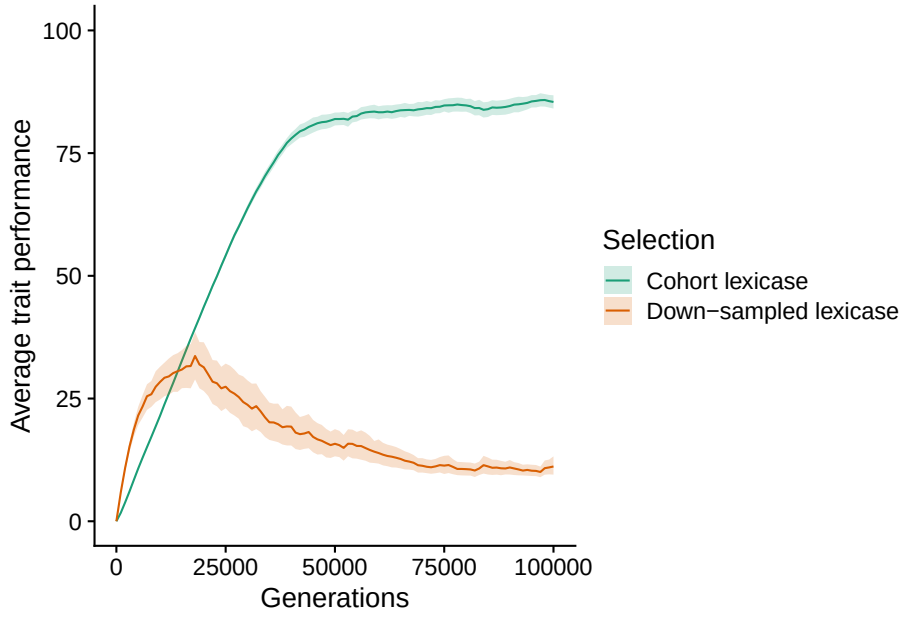
```
data_loc <- paste0(working_directory, "data/timeseries.csv")
data <- read.csv(data_loc, na.strings="NONE")

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

# I always set cohort and downsampled lexicase sampling rates to
# be the same on a given run (regardless of selection scheme)
data$proportion <- factor(
  data$COH_LEX_PROP,
  levels=c(1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01)
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
```

```
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

# Labeler for stats annotations
p_label <- function(p_value) {
  threshold = 0.0001
  if (p_value < threshold) {
    return(paste0("p < ", threshold))
  } else {
    return(paste0("p = ", p_value))
  }
}

# Significance threshold
alpha <- 0.05

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

10.4 Exploration diagnostic performance

```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(
      x=gen,
      y=elite_trait_avg,
      color=selection_name,
      fill=selection_name
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
```

```
    limits=c(0, 100)
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  scale_color_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  )
elite_ave_performance_fig
```



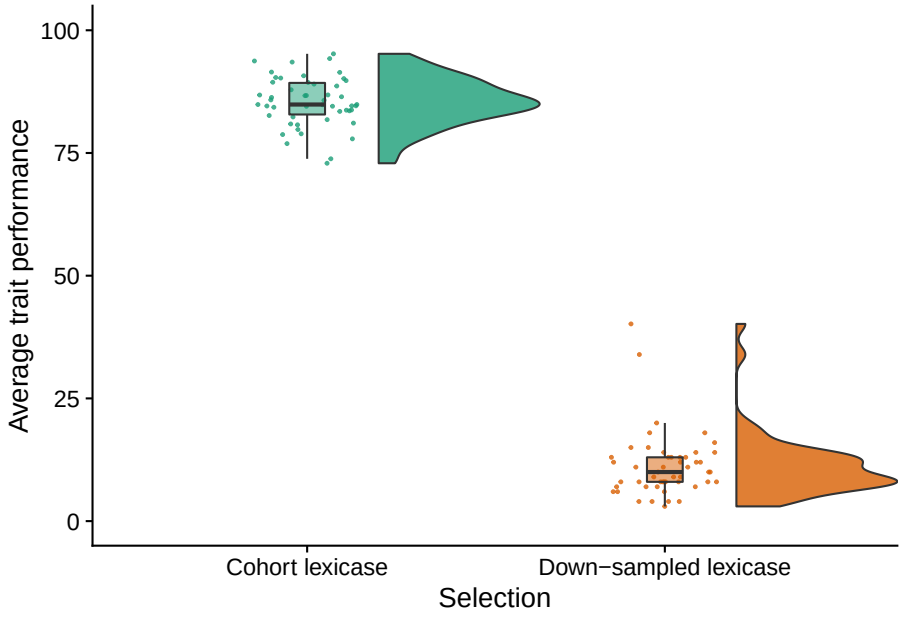
10.4.1 Final performance

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(elite_trait_avg ~ selection_name) %>%
```

```
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(x=selection_name, y=elite_trait_avg, fill=selection_name)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=selection_name),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Selection",
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  scale_color_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
```

```
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```

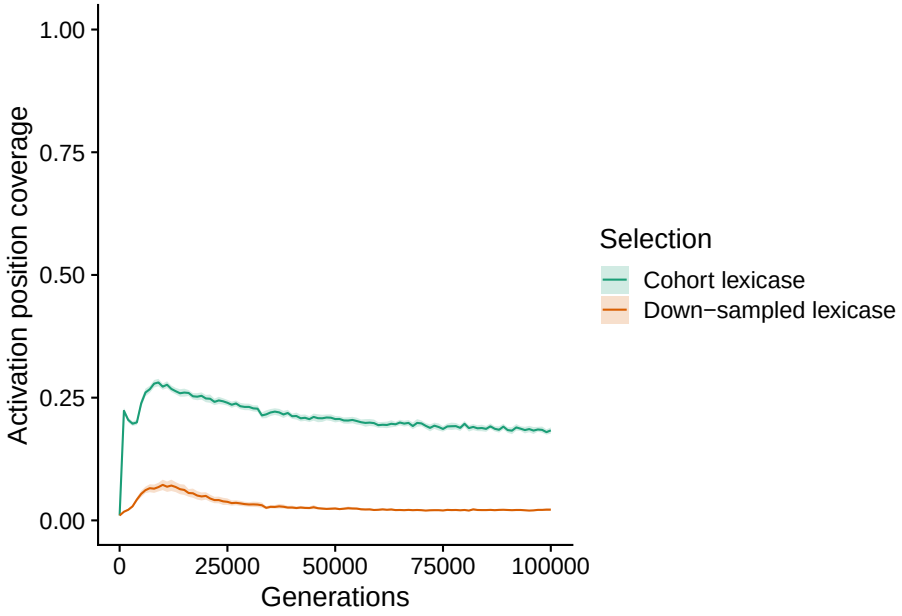


.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.sig
elite_trait_avg	CohortLexicase	DownSampledLexicase	50	50	2500	0	0	****

10.5 Unique starting positions

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(
      x=gen,
      y=unique_start_positions_coverage,
      color=selection_name,
      fill=selection_name
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
```

```
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")

  ) +
  scale_color_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  )
unique_start_position_coverage_fig
```

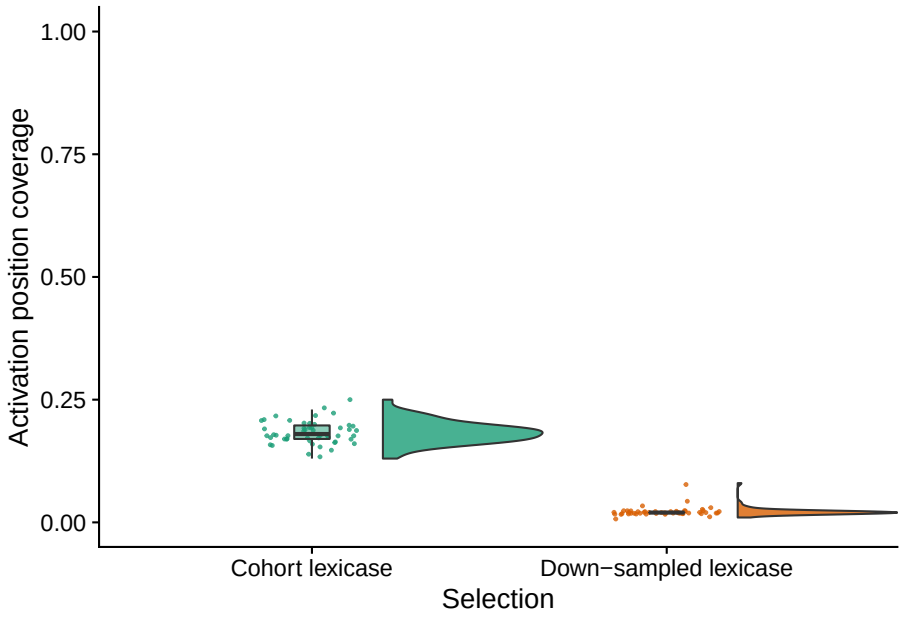


10.5.1 Final starting position coverage

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(unique_start_positions_coverage ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name",step.increase=1)
stat.test$manual_position <- stat.test$y.position * 1.05
stat.test$label <- mapply(p_label,stat.test$p.adj)
```

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(
      x=selection_name,
      y=unique_start_positions_coverage,
      fill=selection_name
    )
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
```

```
  geom_point(
    mapping=aes(color=selection_name),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="Selection",
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  scale_fill_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  scale_color_brewer(
    name="Selection",
    palette=cb_palette,
    limits=c("CohortLexicase", "DownSampledLexicase"),
    labels=c("Cohort lexicase", "Down-sampled lexicase")
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```

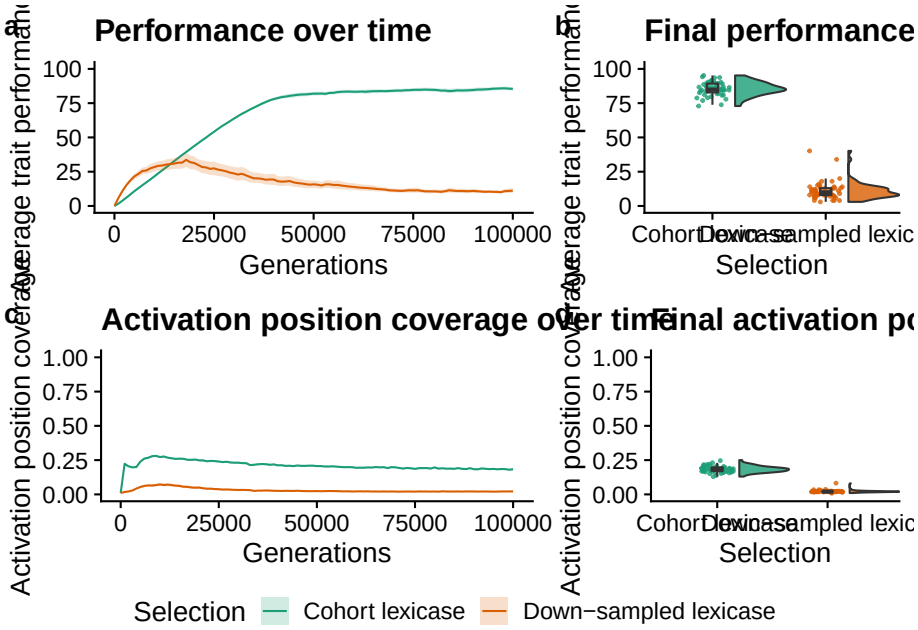


.y.	group1	group2	n1	n2	statistic	p
unique_start_positions_coverage	CohortLexicase	DownSampledLexicase	50	50	2500	0

10.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
```

```
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(
    working_directory,
    "imgs/down-sampled-vs-cohort-panel.pdf",
    sep=""
  ),
  grid,
  base_width=12,
  base_height=8
)

grid
```


a
Performance over time
Average trait performanc
100
75
50
25
0
0
25000
50000
75000
100000
Generations
b
Final performance
Average trait performanc
100
75
50
25
0
Cohort lexicase
Down−sampled lexic
Selection
c
Activation position coverage over time
Activation position coverage
1.00
0.75
0.50
0.25
0.00
0
25000
50000
75000
100000
Generations
d
Final activation pc
Activation position coverage
1.00
0.75
0.50
0.25
0.00
Cohort lexicase
Down−sampled lexic
Selection
Selection
Cohort lexicase
Down−sampled lexicase

Chapter 11

Novelty lexicase

11.1 Overview

```
# Relative location of data.
working_directory <- "experiments/2021-06-01-novelty/analysis/"
# working_directory <- "./"

# Settings for visualization
cb_palette <- "Set2"
# Create directory to dump plots
dir.create(paste0(working_directory, "imgs"), showWarnings=FALSE)
```

11.2 Analysis dependencies

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

These analyses were conducted in the following computing environment:

```
print(version)
```

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
```

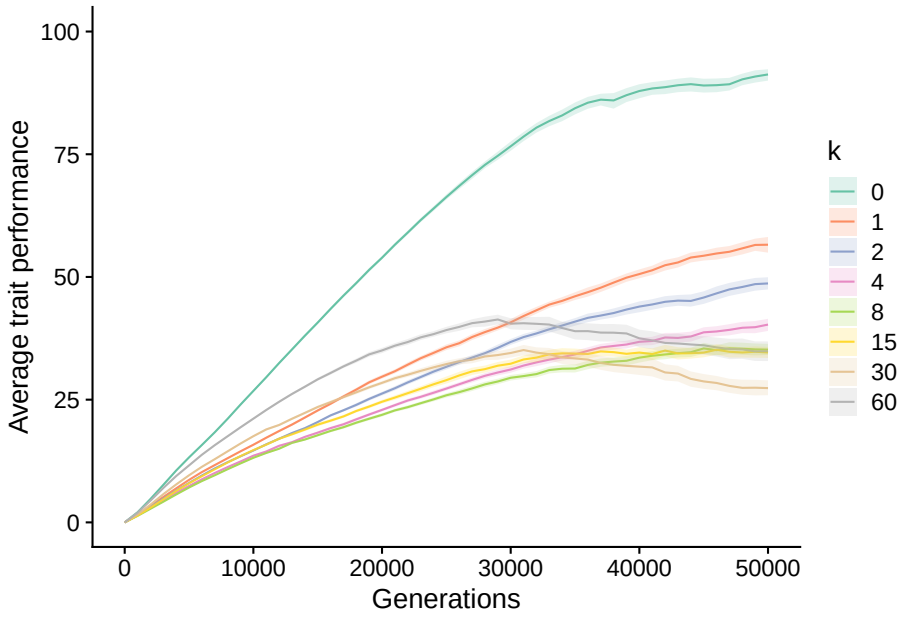
```
## system         x86_64, linux-gnu
## status
## major          4
## minor          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen
```

11.3 Setup

```
data_loc <- paste0(
  working_directory,
  "data/timeseries-res-1000g.csv"
)

data <- read.csv(data_loc, na.strings="NONE")

data$cardinality <- as.factor(
  data$OBJECTIVE_CNT
)
data$selection_name <- as.factor(
  data$selection_name
)

data$epsilon <- as.factor(
  data$LEX_EPS
)

data$k <- as.factor(
   data$NOVEL_K
)

data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

final_data <- filter(data, evaluations==max(data$evaluations))

####### misc #######
```

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

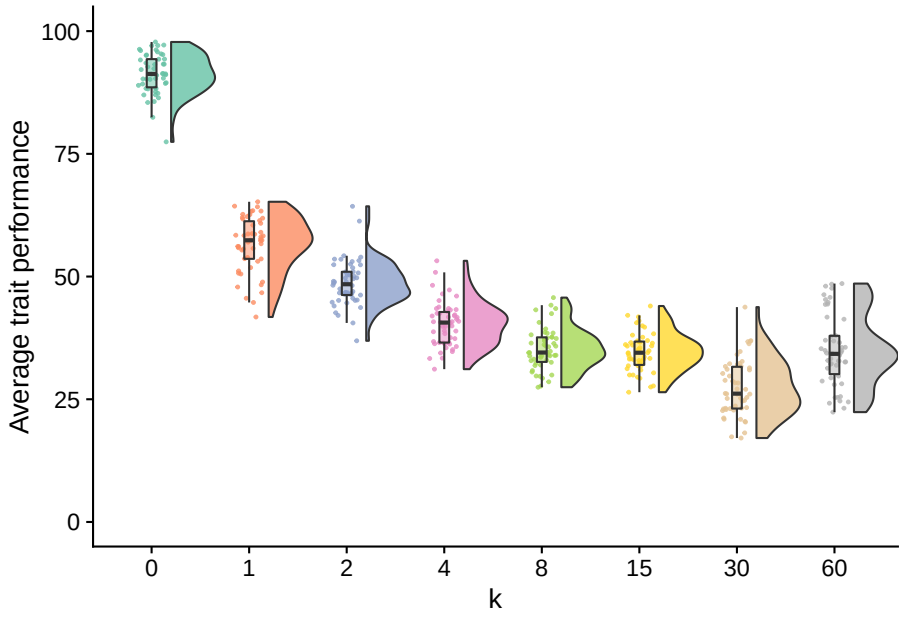
```
elite_ave_performance_fig <-
  ggplot(
    data,
    aes(x=gen, y=elite_trait_avg, color=k, fill=k)
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="k",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="k",
    palette=cb_palette
  )
elite_ave_performance_fig
```



11.4.1 Final performance

```
elite_final_performance_fig <- ggplot(
    final_data,
    aes(x=k, y=elite_trait_avg, fill=k)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=k),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Average trait performance",
```

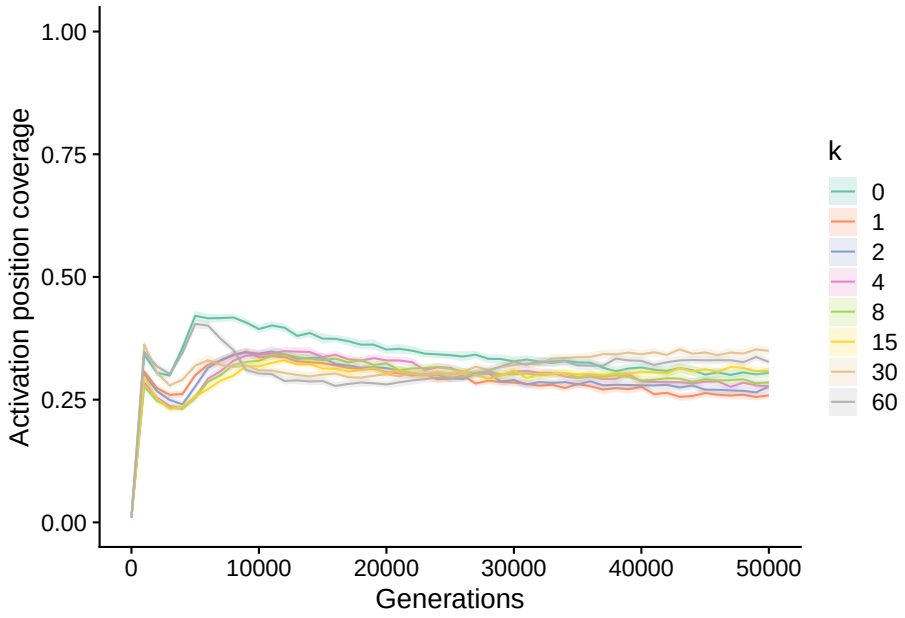
```
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="k"
  ) +
  scale_fill_brewer(
    name="k",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="k",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_final_performance_fig
```



11.5 Unique starting positions

```
unique_start_position_coverage_fig <- ggplot(
    data,
    aes(x=gen, y=unique_start_positions_coverage, color=k, fill=k)
```

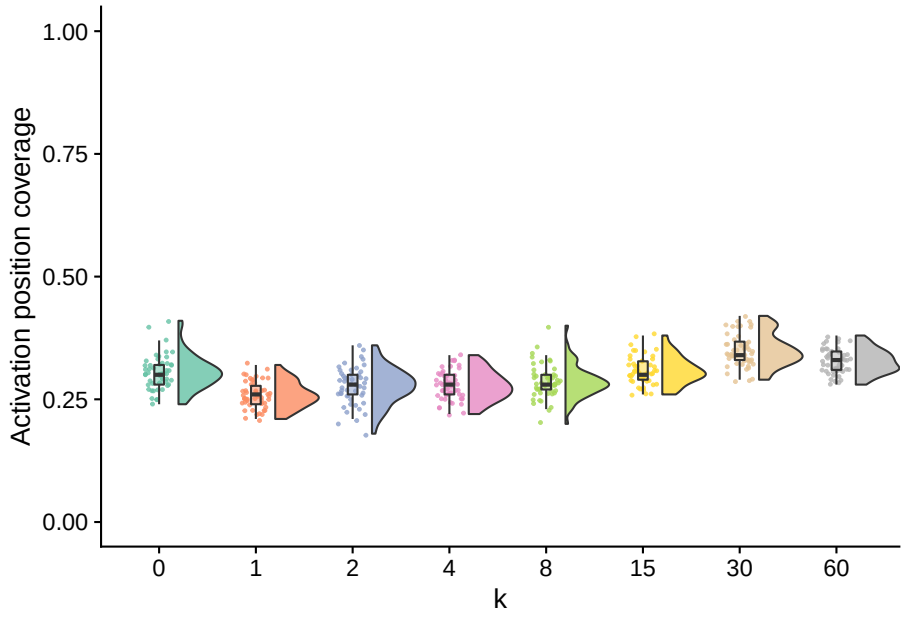
```
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Activation position coverage",
    limits=c(0.0, 1.0)
  ) +
  scale_x_continuous(
    name="Generations"
  ) +
  scale_fill_brewer(
    name="k",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="k",
    palette=cb_palette
  )
unique_start_position_coverage_fig
```



11.5.1 Final starting position coverage

```
unique_start_positions_coverage_final_fig <- ggplot(
    final_data,
    aes(x=k, y=unique_start_positions_coverage, fill=k)
  ) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
  ) +
  geom_point(
    mapping=aes(color=k),
    position = position_jitter(width = .15),
    size = .5,
    alpha = 0.8
  ) +
  geom_boxplot(
    width = .1,
    outlier.shape = NA,
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Activation position coverage",
```

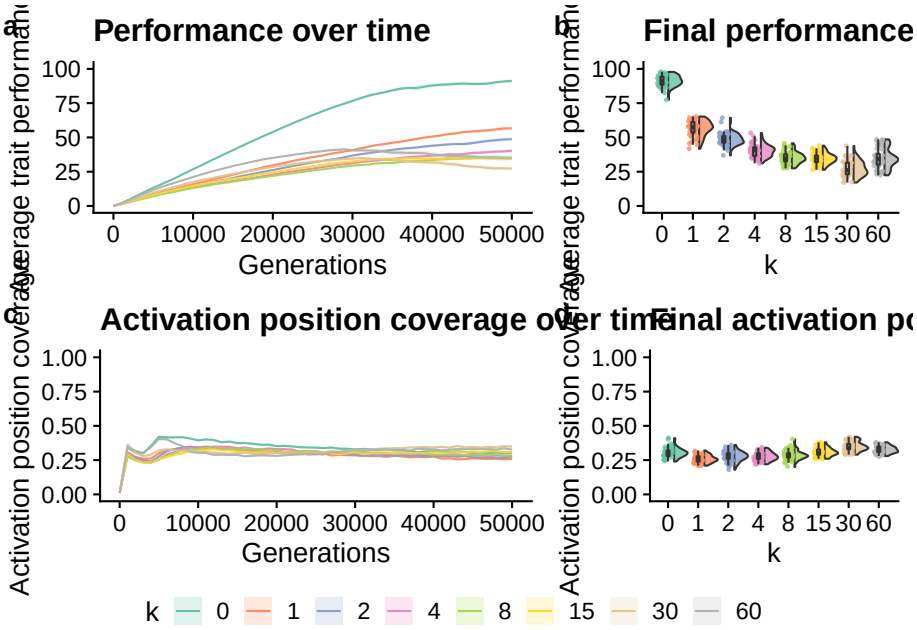
```
    limits=c(0, 1.0)
  ) +
  scale_x_discrete(
    name="k"
  ) +
  scale_fill_brewer(
    name="k",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="k",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
unique_start_positions_coverage_final_fig
```



11.6 Manuscript figures

```
legend <- cowplot::get_legend(
    elite_ave_performance_fig +
      guides(
```

```
        color=guide_legend(nrow=1),
        fill=guide_legend(nrow=1)
      ) +
      theme(
        legend.position = "bottom",
        legend.box="horizontal",
        legend.justification="center"
      )
  )

grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
    theme(legend.position="none"),
  unique_start_positions_coverage_final_fig +
    ggtitle("Final activation position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(3,2),
  labels="auto"
)

grid <- plot_grid(
  grid,
  legend,
  nrow=2,
  ncol=1,
  rel_heights=c(1, 0.1)
)

save_plot(
  paste(working_directory, "imgs/novelty-panel.pdf", sep=""),
  grid,
  base_width=12,
  base_height=8
)

grid
```


a
Performance over time
Average trait performanc
100
75
50
25
0
0
10000
20000
30000
40000
50000
Generations
b
Final performance
Average trait performanc
100
75
50
25
0
0
1
2
4
8
15
30
60
k
c
Activation position coverage over time
Activation position coverage
1.00
0.75
0.50
0.25
0.00
0
10000
20000
30000
40000
50000
Generations
Final activation pc
Activation position coverage
1.00
0.75
0.50
0.25
0.00
0
1
2
4
8
15
30
60
k
k
0
1
2
4
8
15
30
60

Bibliography

Lalejini, A. M. and Hernandez, J. G. (2021). Data for measuring the ability of lexicase selection to find obscure pathways to optimality.