

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

2021-06-17

Contents

1	Introduction	5
1.1	About our supplemental material	5
1.2	Contributing authors	5
1.3	Research overview	5
2	Data Availability	7
2.1	Source code	7
2.2	Experimental results	7
3	Compile and run experiments	9
3.1	Docker	9
4	Tournament selection vs Lexicase selection	11
4.1	Overview	11
4.2	Analysis dependencies	11
4.3	Setup	12
4.4	Exploration diagnostic performance	12
4.5	Final Performance	14
4.6	Unique starting positions	16
4.7	Manuscript figures	20
5	Diagnostic cardinality	23
5.1	Overview	23
5.2	Analysis dependencies	23
5.3	Setup	24
5.4	Exploration diagnostic performance	25
5.5	Unique starting positions	27
5.6	Manuscript figures	32
6	Increasing population size versus increasing generations	35
6.1	Overview	35
6.2	Analysis dependencies	35
6.3	Setup	36
6.4	Exploration diagnostic performance	37

6.5	Unique starting positions	40
6.6	Manuscript figures	44
7	Epsilon lexicase	47
7.1	Overview	47
7.2	Analysis dependencies	47
7.3	Setup	48
7.4	Exploration diagnostic performance	49
7.5	Unique starting positions	51
7.6	Manuscript figures	55
8	Down-sampled lexicase	57
8.1	Overview	57
8.2	Analysis dependencies	57
8.3	Setup	58
8.4	Exploration diagnostic performance	59
8.5	Unique starting positions	62
8.6	Manuscript figures	65
9	Cohort lexicase	67
9.1	Overview	67
9.2	Analysis dependencies	67
9.3	Setup	68
9.4	Exploration diagnostic performance	69
9.5	Unique starting positions	71
9.6	Manuscript figures	75
10	Down-sampled lexicase versus cohort lexicase	77
10.1	Overview	77
10.2	Analysis dependencies	77
10.3	Setup	78
10.4	Exploration diagnostic performance	79
10.5	Unique starting positions	82
10.6	Manuscript figures	85
11	Novelty lexicase	89
11.1	Overview	89
11.2	Analysis dependencies	89
11.3	Setup	90
11.4	Exploration diagnostic performance	91
11.5	Unique starting positions	93
11.6	Manuscript figures	96

Chapter 1

Introduction

This is the supplemental material associated with our 2021 GPTP contribution entitled, *An Exploration of Exploration: Measuring the ability of lexibase 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(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
```

4.3 Setup

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

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

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

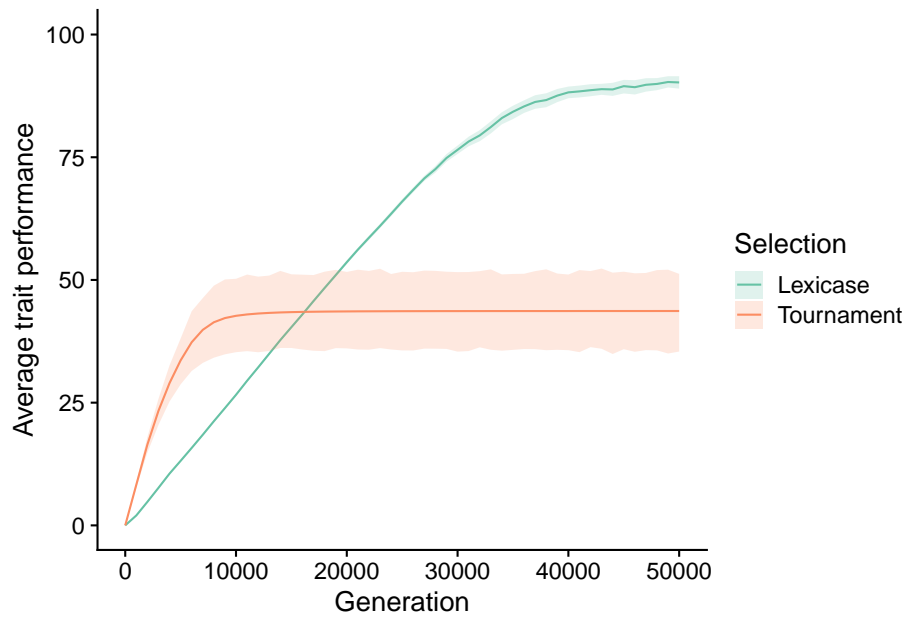
final_data <- filter(data, evaluations==max(data$evaluations))

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

4.4 Exploration diagnostic performance

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

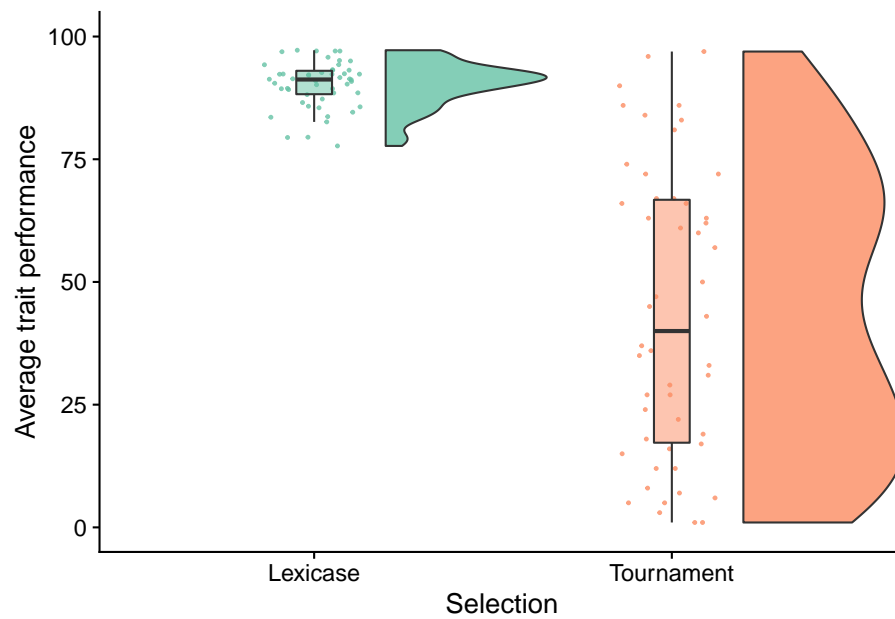
```
    x=gen,
    y=elite_trait_avg,
    color=selection_name,
    fill=selection_name
  )
) +
stat_summary(geom="line", fun=mean) +
stat_summary(
  geom="ribbon",
  fun.data="mean_cl_boot",
  fun.args=list(conf.int=0.95),
  alpha=0.2,
  linetype=0
) +
scale_y_continuous(
  name="Average trait performance",
  limits=c(0, 100)
) +
scale_x_continuous(
  name="Generation"
) +
scale_fill_brewer(
  name="Selection",
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
  palette=cb_palette
) +
scale_color_brewer(
  name="Selection",
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
  palette=cb_palette
)
elite_ave_performance_fig
```



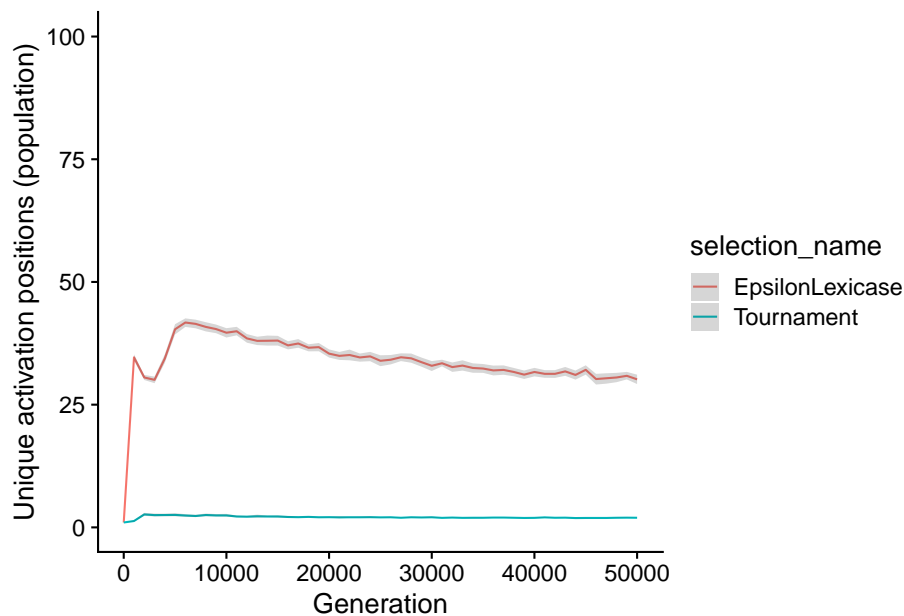
4.5 Final Performance

```
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("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
scale_fill_brewer(
  name="Selection",
  palette=cb_palette,
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
scale_color_brewer(
  name="Selection",
  palette=cb_palette,
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
theme(
  legend.position="none"
)
elite_final_performance_fig
```



```
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"
  )
```

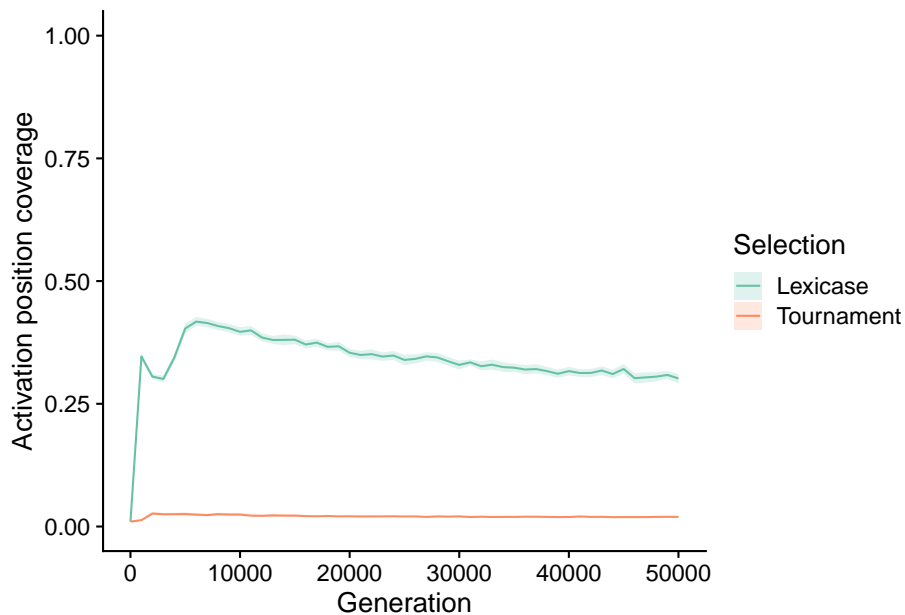
Different cardinalities have numbers of possible starting positions, so next, we look at the proportion of starting positions (out of all possible) maintained by populations.

```
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, 1.0)
) +
```

```

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

```



4.6.1 Final starting position Coverage

```

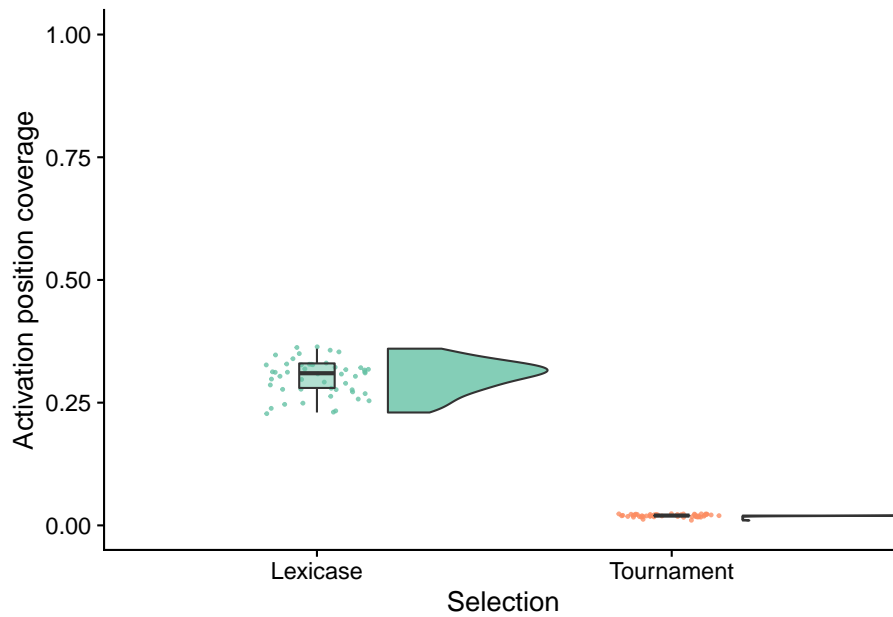
unique_start_positions_coverage_final_fig <- ggplot(
  final_data,
  aes(
    x=selection_name,
    y=unique_start_positions_coverage,

```

```

    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("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
scale_fill_brewer(
  name="Selection",
  palette=cb_palette,
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
scale_color_brewer(
  name="Selection",
  palette=cb_palette,
  limits=c("EpsilonLexicase", "Tournament"),
  labels=c("Lexicase", "Tournament"),
) +
theme(
  legend.position="none"
)
unique_start_positions_coverage_final_fig

```



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") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +

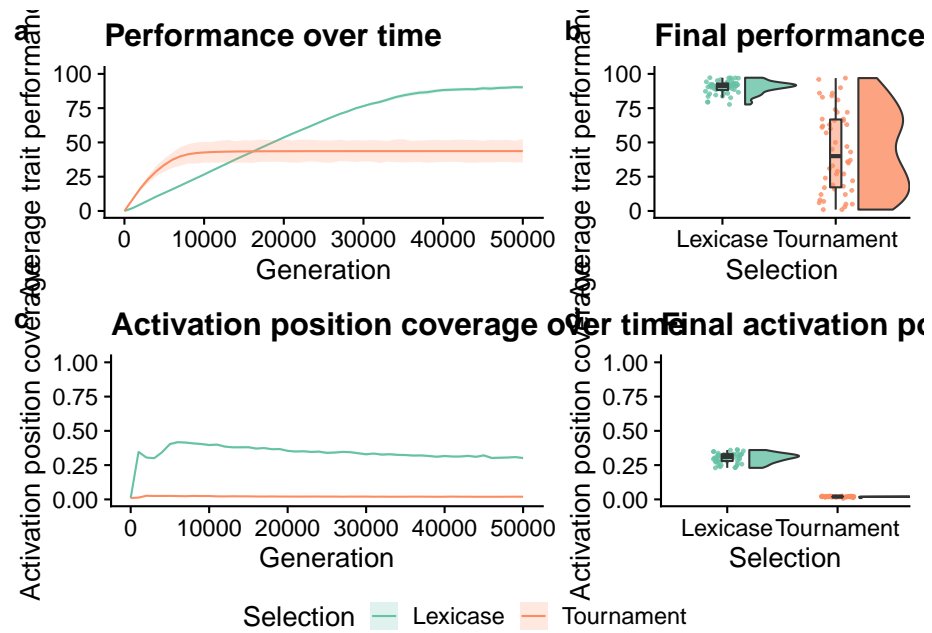
```

```
    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/tournament-vs-lexicase-panel.pdf",
    sep=""
  ),
  grid,
  base_width=12,
  base_height=8
)

grid
```



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(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
```

```
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

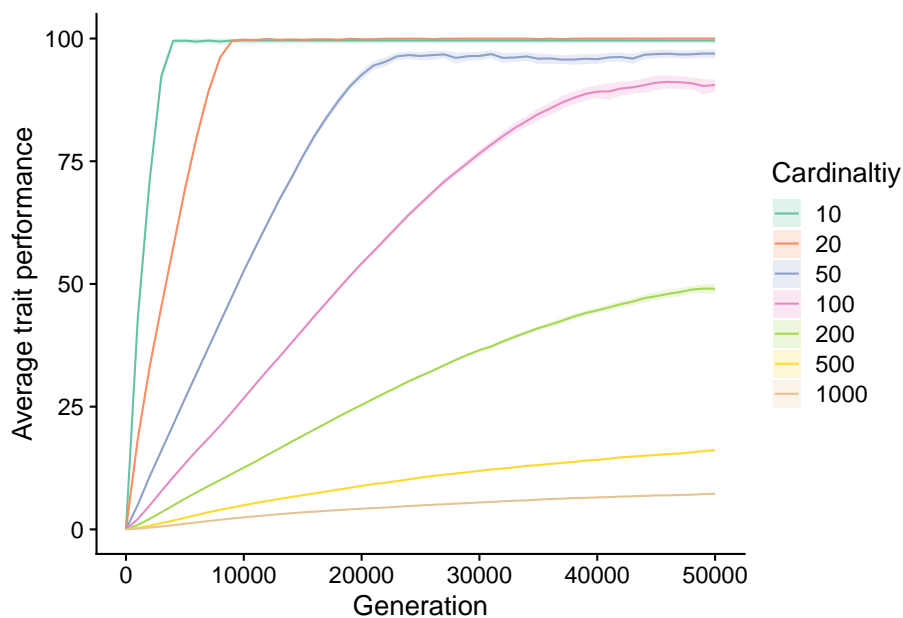
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

##### 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 aggregate 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="Cardinality",  
  palette=cb_palette  
) +  
scale_color_brewer(  
  name="Cardinality",  
  palette=cb_palette  
)  
elite_trait_ave_fit
```



5.4.1 Final performance

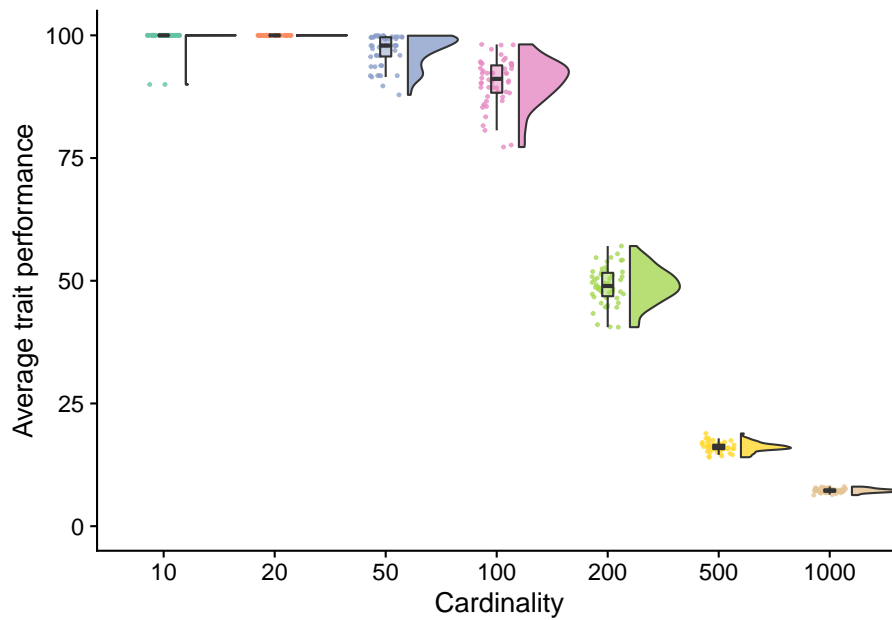
Next, we look only at the final performances of each treatment

```
final_data <- filter(data, gen==max(data$gen))
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="Cardinality",
  palette=cb_palette
) +
scale_color_brewer(
  name="Cardinality",
  palette=cb_palette
) +
theme(
  legend.position="none"
)
elite_trait_ave_fit_final

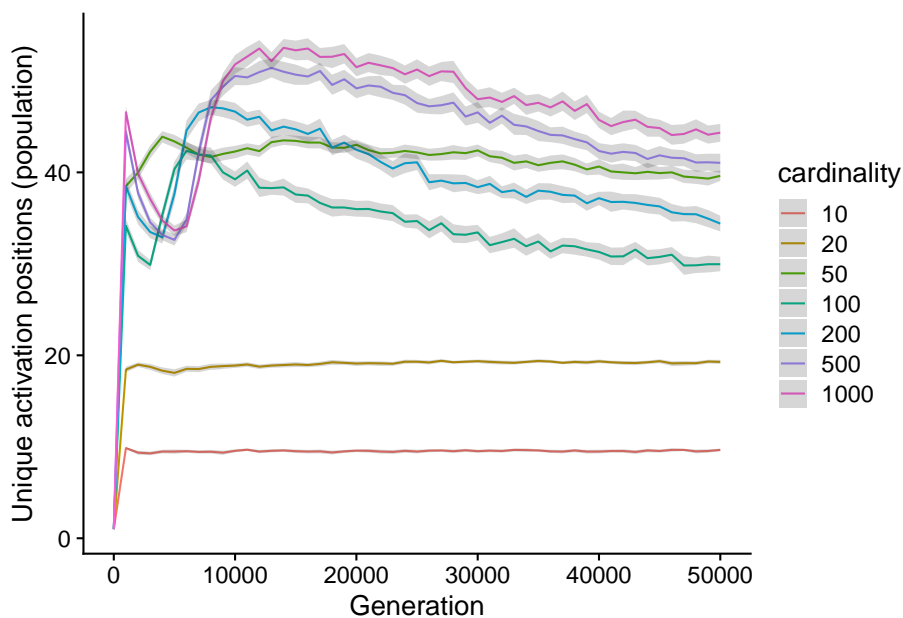
```



5.5 Unique starting positions

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

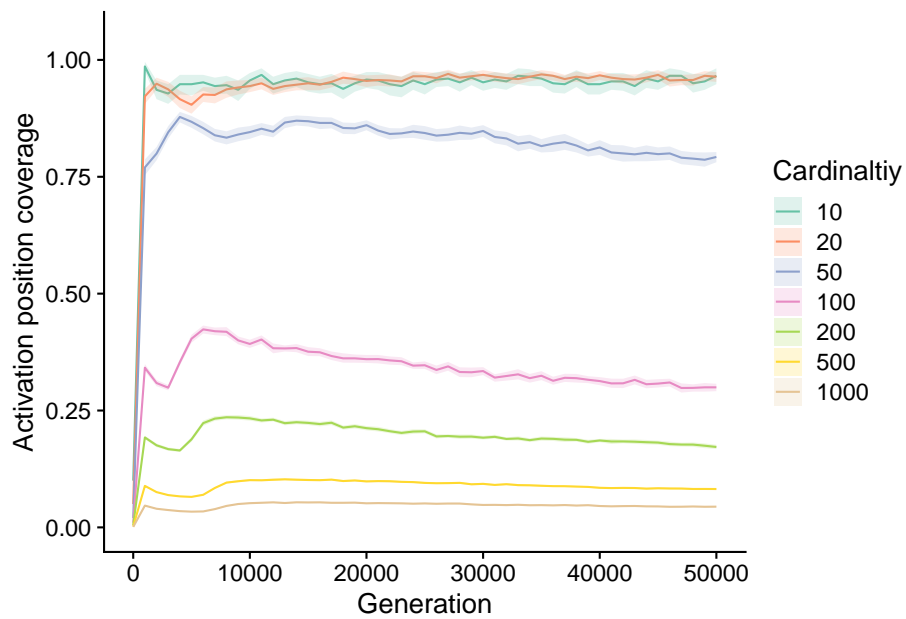
```
ggplot(data, aes(x=gen, y=uni_str_pos, color=cardinality)) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Unique activation positions (population)",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



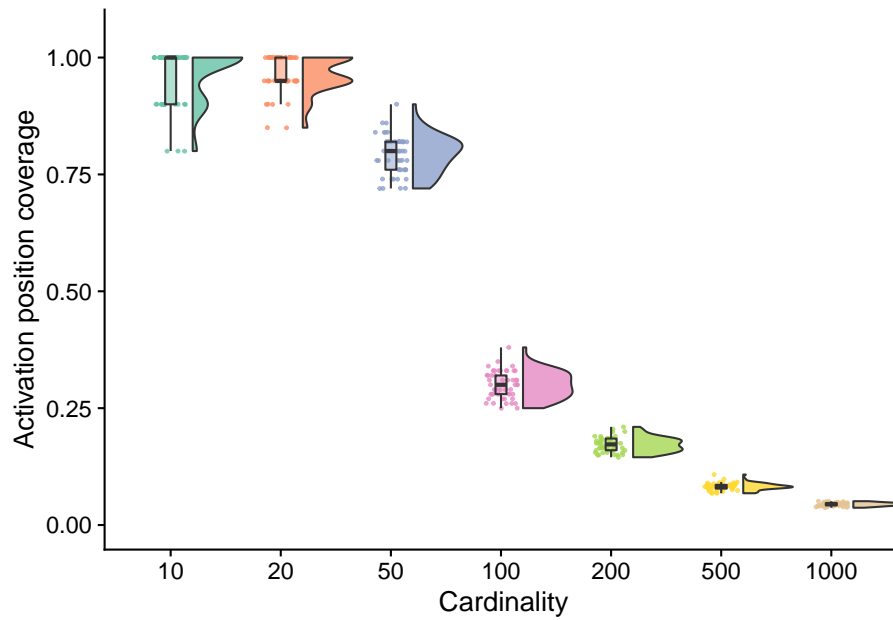
5.5.1 Final starting position coverage

```
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="Cardinality",
  palette=cb_palette
) +
scale_color_brewer(
  name="Cardinality",
  palette=cb_palette
) +
theme(
  legend.position="none"
)
final_unique_start_positions_coverage_fig

```



5.6 Manuscript figures

Combine figures for the manuscript.

```

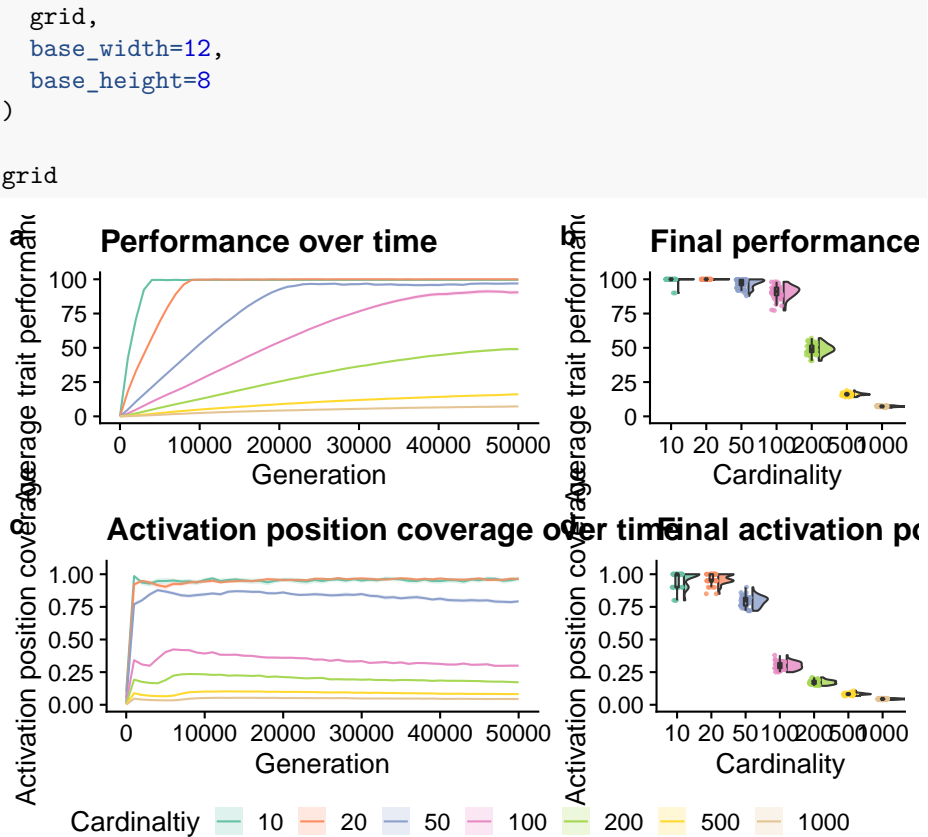
legend <- cowplot::get_legend(
  elite_trait_ave_fit +
    guides(
      color=guide_legend(nrow=1),
      fill=guide_legend(nrow=1)
    ) +
    theme(
      legend.position = "bottom",
      legend.box="horizontal",
      legend.justification="center"
    )
)

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=""),

```

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
)

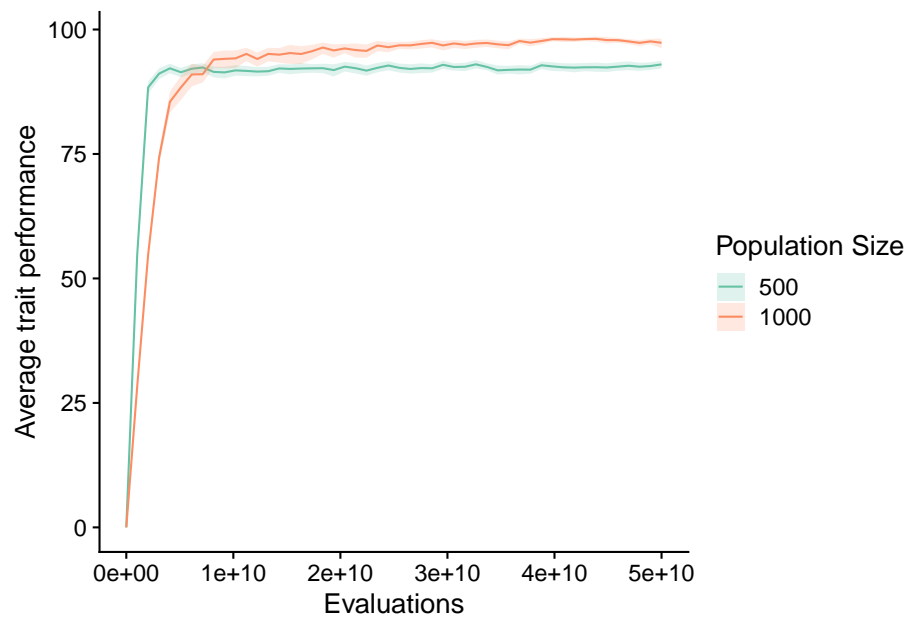
data <- filter(data, cardinality=="100") # These runs finished.
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT
```

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

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

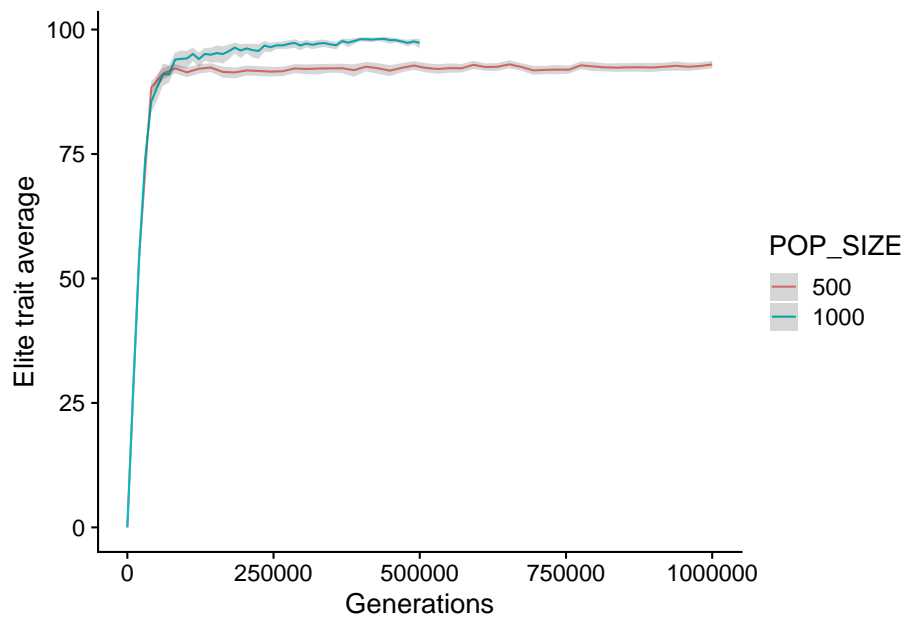
6.4 Exploration diagnostic performance

```
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"
  )
```



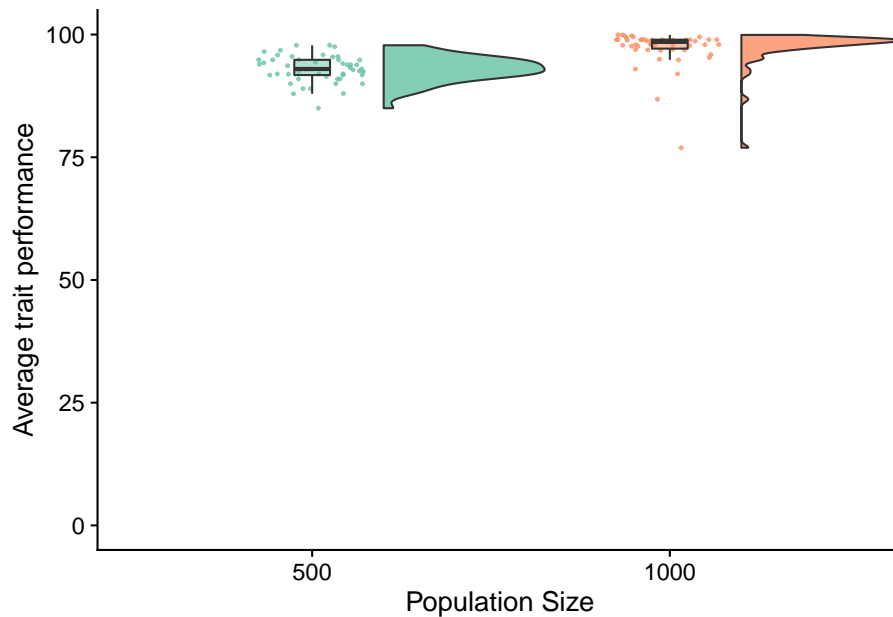
6.4.1 Final performance

```
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

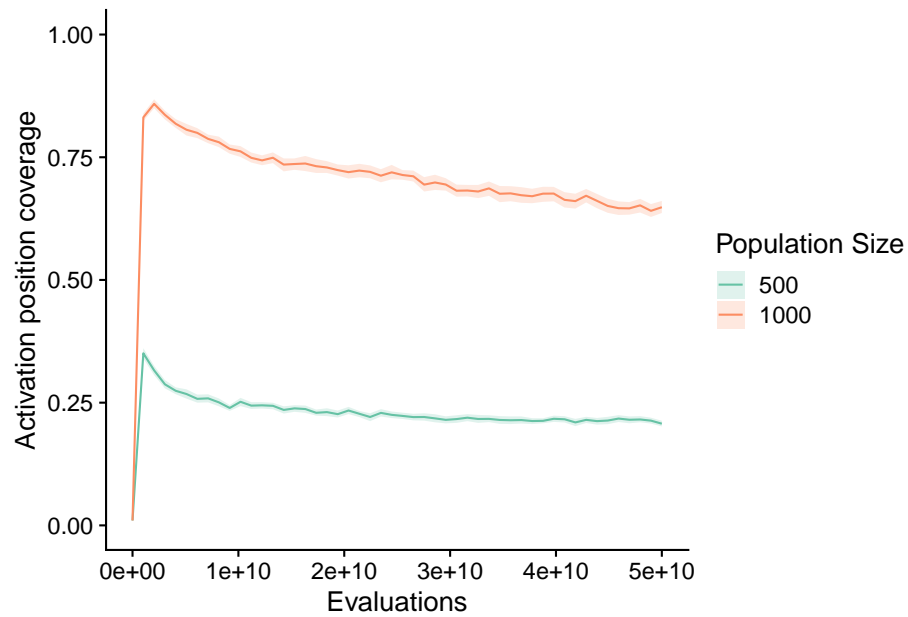
```

unique_start_position_coverage_fig <- ggplot(
  data,
  aes(

```



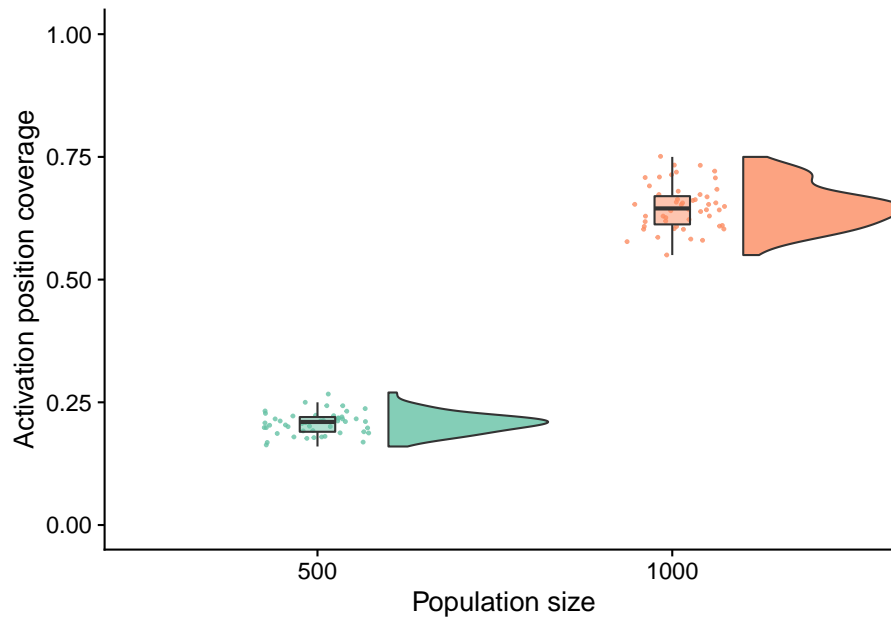
```
    x=evaluations,
    y=unique_start_positions_coverage,
    color=POP_SIZE,
    fill=POP_SIZE
  )
) +
stat_summary(geom="line", fun=mean) +
stat_summary(
  geom="ribbon",
  fun.data="mean_cl_boot",
  fun.args=list(conf.int=0.95),
  alpha=0.2,
  linetype=0
) +
scale_y_continuous(
  name="Activation 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

```

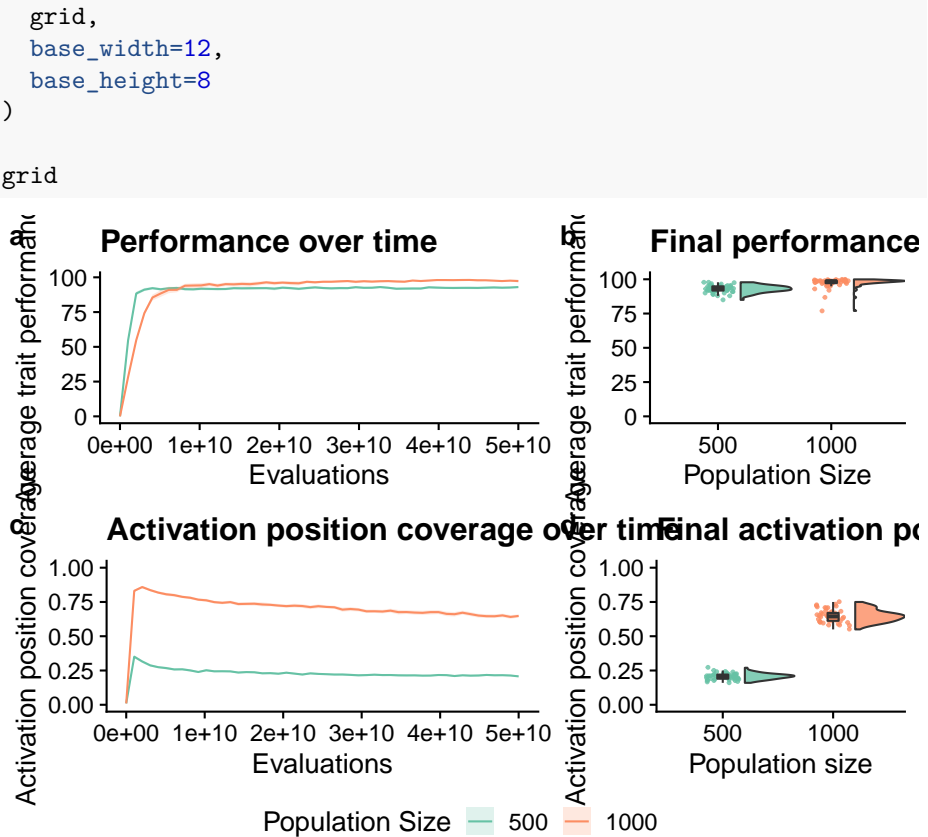
legend <- cowplot::get_legend(
  elite_ave_performance_fig +
    guides(
      color=guide_legend(nrow=1),
      fill=guide_legend(nrow=1)
    ) +
    theme(
      legend.position = "bottom",
      legend.box="horizontal",
      legend.justification="center"
    )
)

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=""),

```



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
)

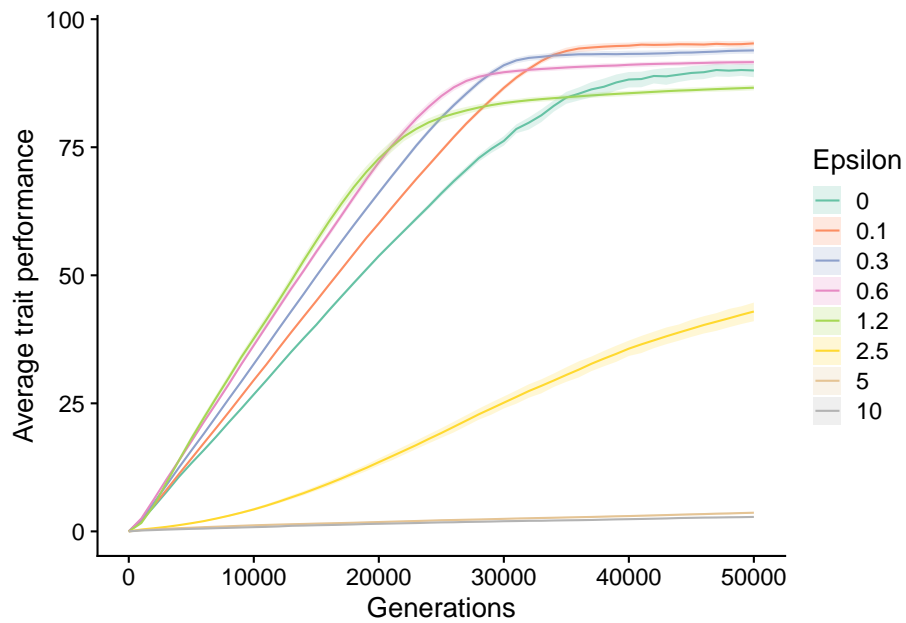
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

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
```



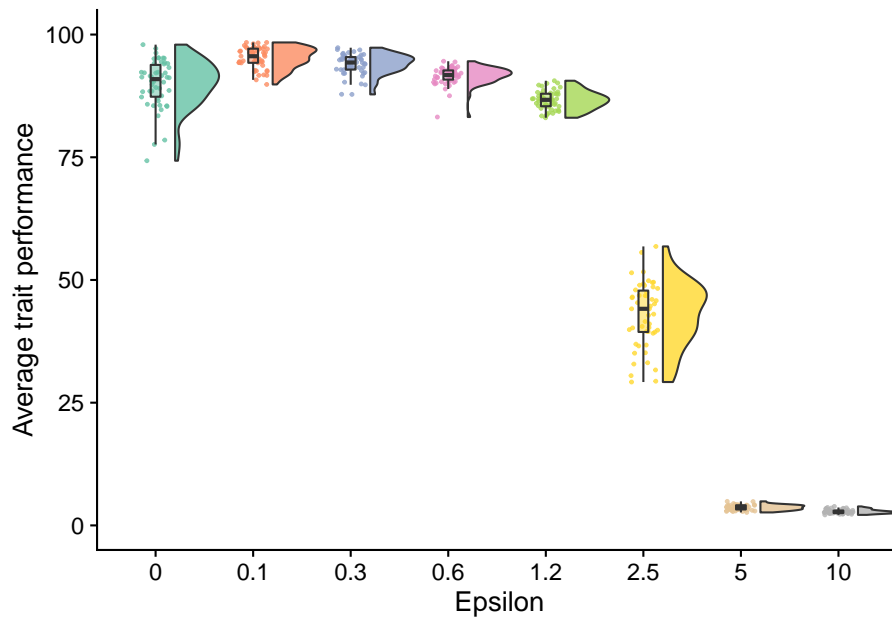
7.4.1 Final performance

```
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

```



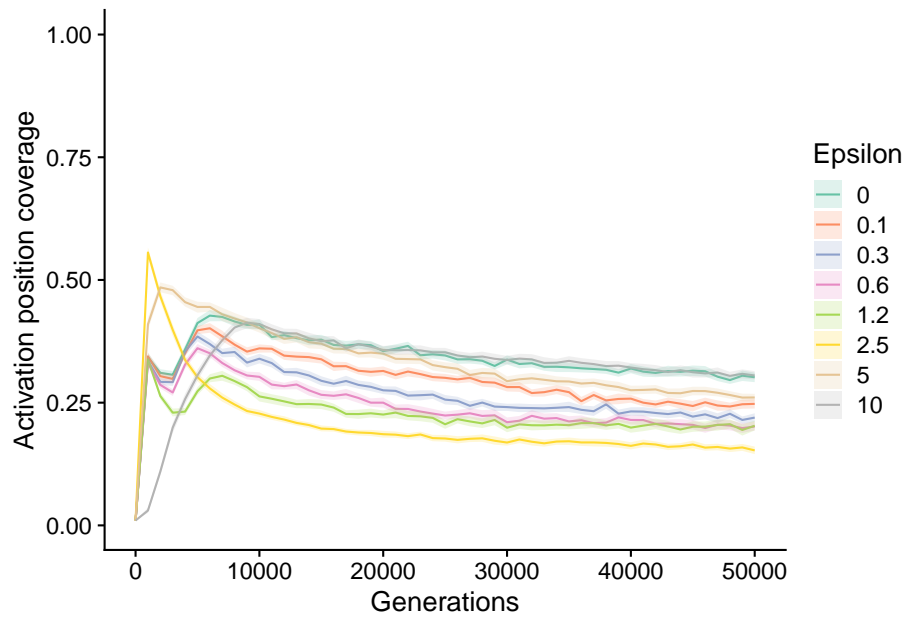
7.5 Unique starting positions

```

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
```



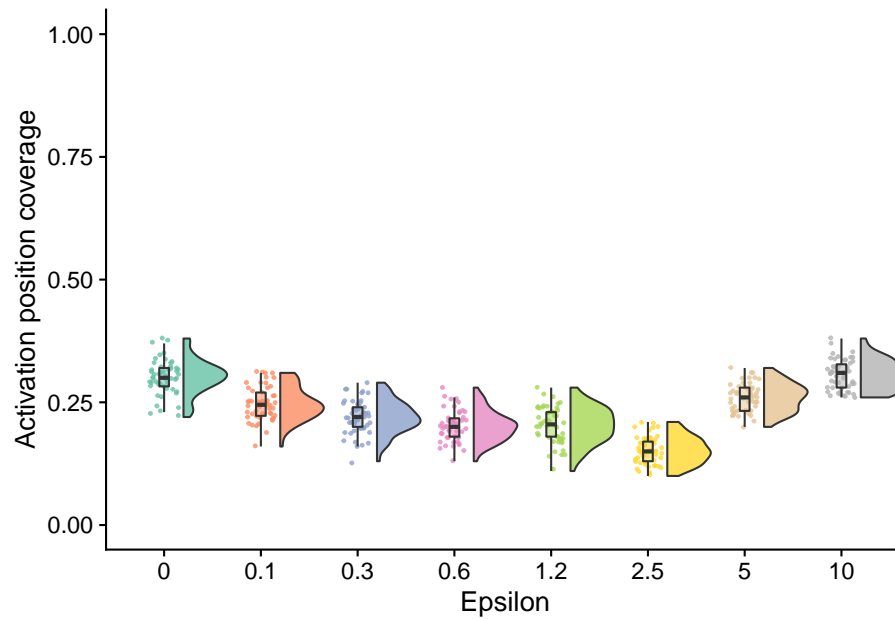
7.5.1 Final coverage

```
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)
)

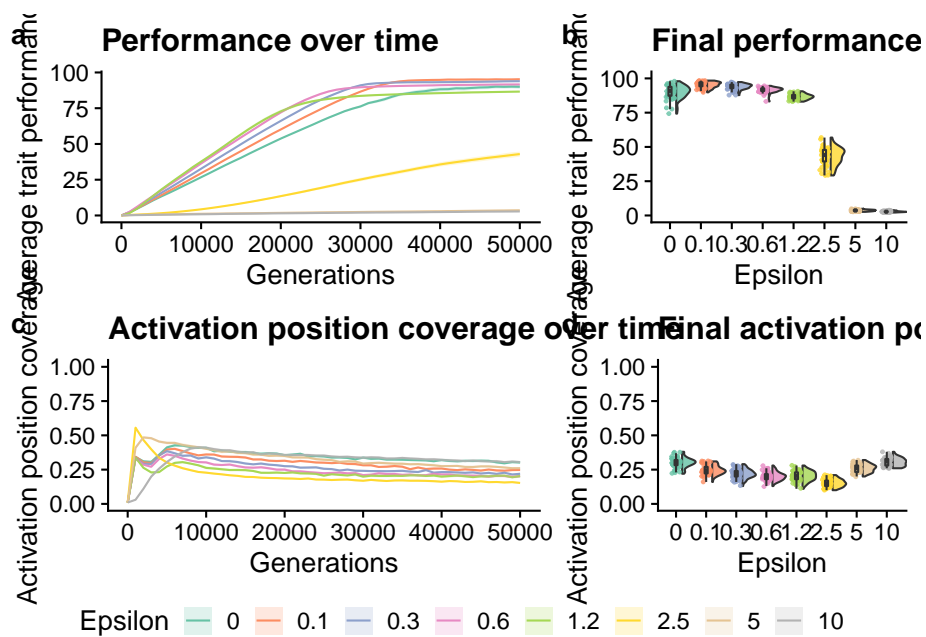
save_plot(
  paste(working_directory, "imgs/epsilon-panel.pdf", sep=""),
  grid,

```

```

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
```

```
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)
)

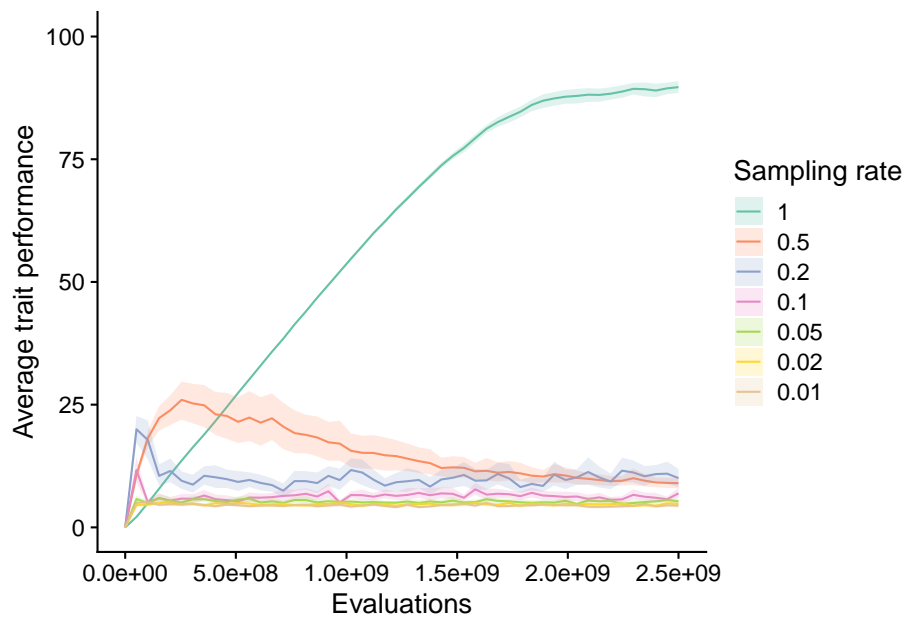
data$elite_trait_avg <- data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <- data$uni_str_pos / data$OBJECTIVE_CNT

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
```



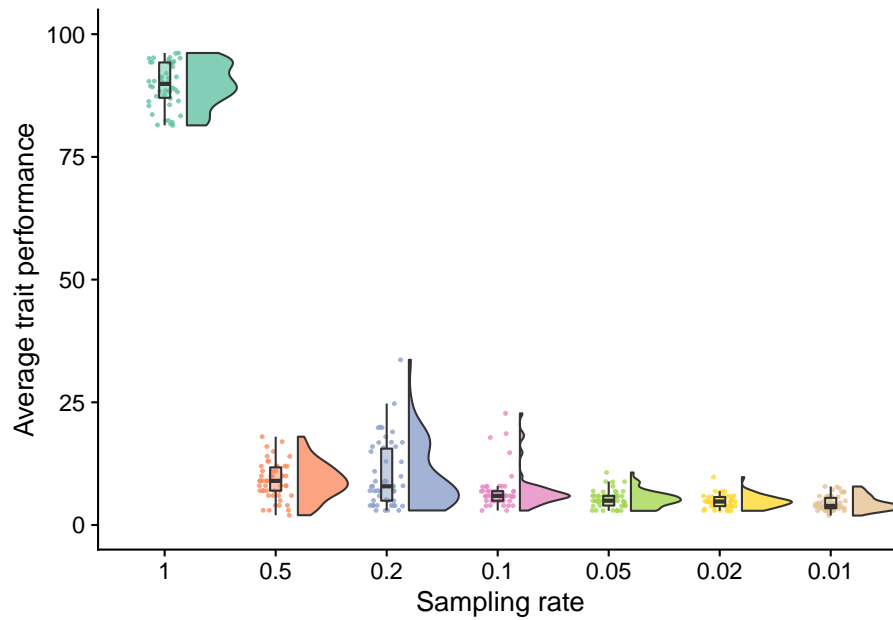
8.4.1 Final performance

```
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,
```

```

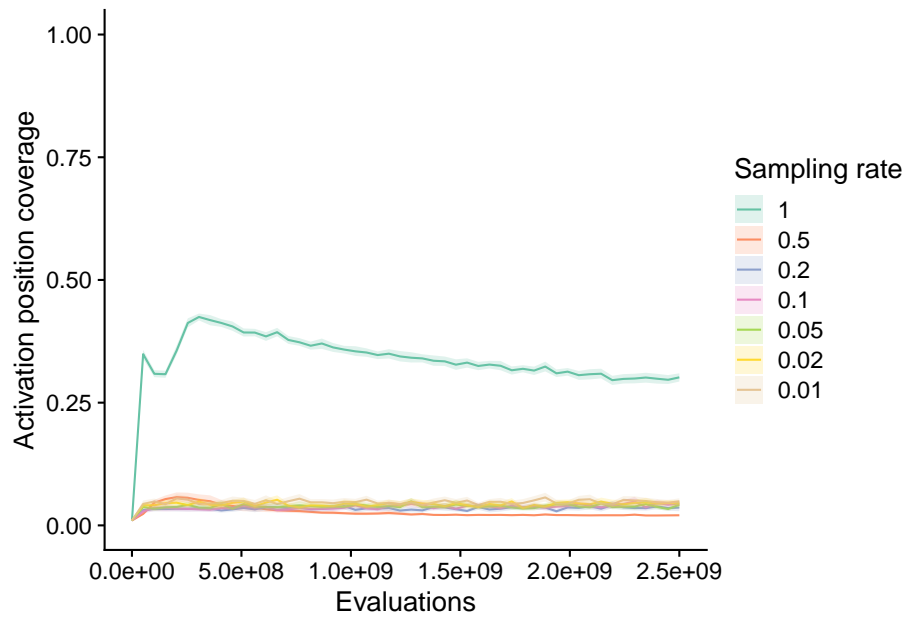
alpha = 0.5
) +
scale_y_continuous(
  name="Average trait performance",
  limits=c(0, 100)
) +
scale_x_discrete(
  name="Sampling rate"
) +
scale_fill_brewer(
  name="Sampling rate",
  palette=cb_palette
) +
scale_color_brewer(
  name="Sampling rate",
  palette=cb_palette
) +
theme(
  legend.position="none"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



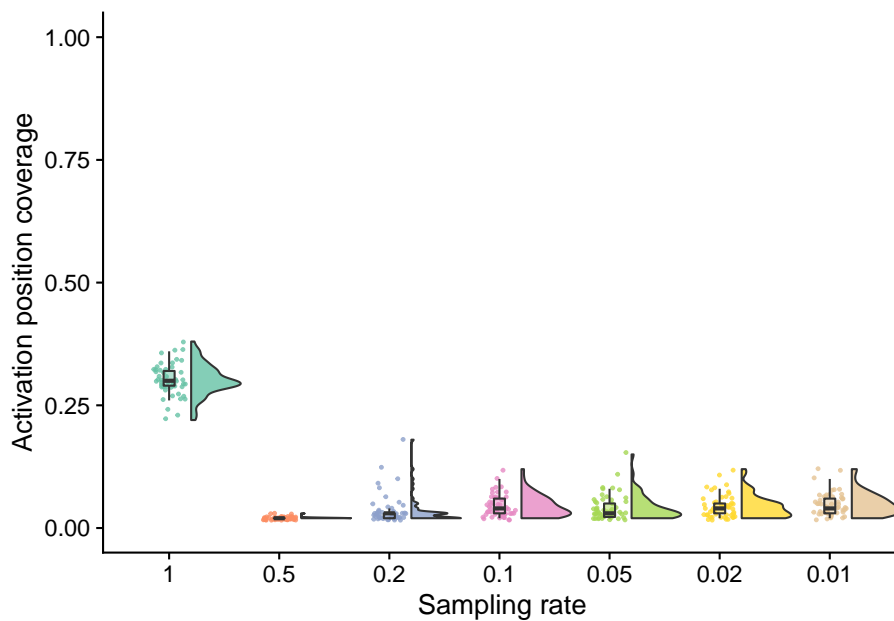
8.5.1 Final starting position coverage

```
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)
)

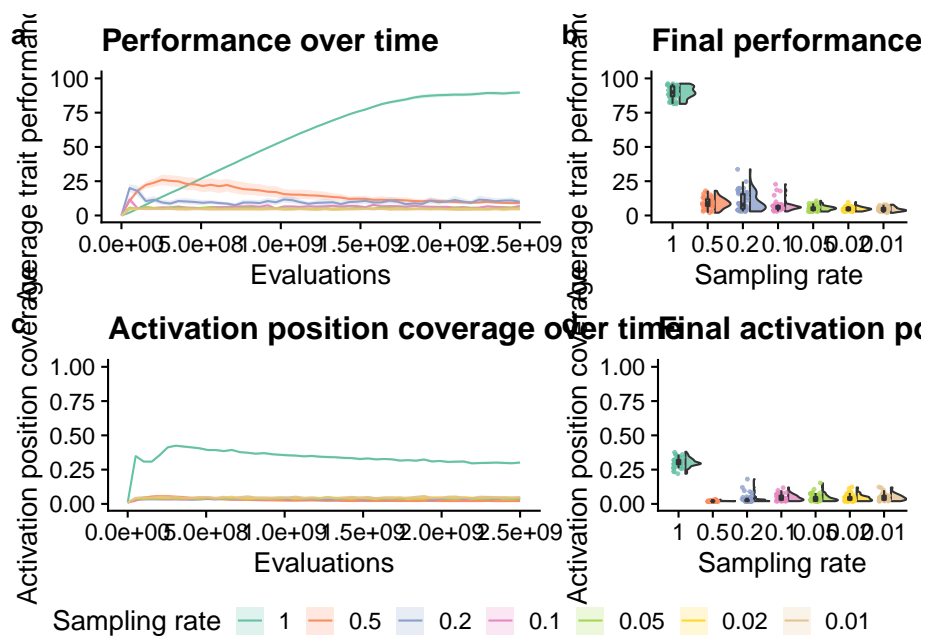
save_plot(
  paste(working_directory, "imgs/down-sampled-panel.pdf", sep=""),
  grid,

```

```

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)
)

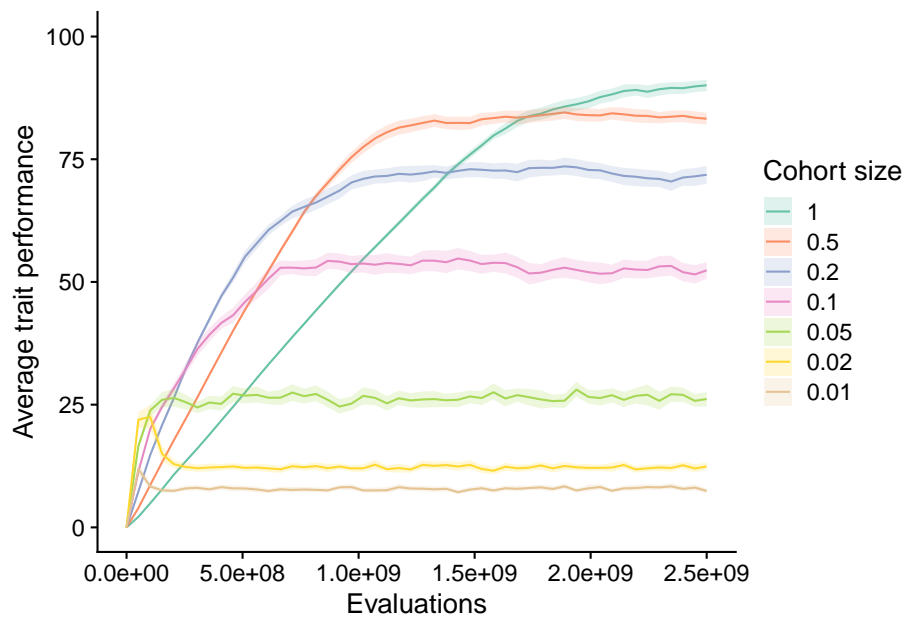
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

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
```



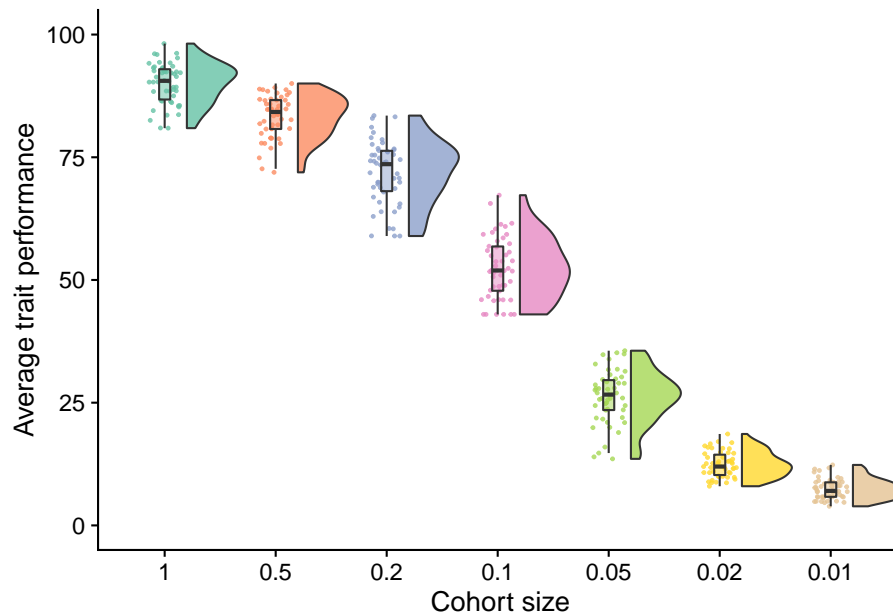
9.4.1 Final performance

```
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

```



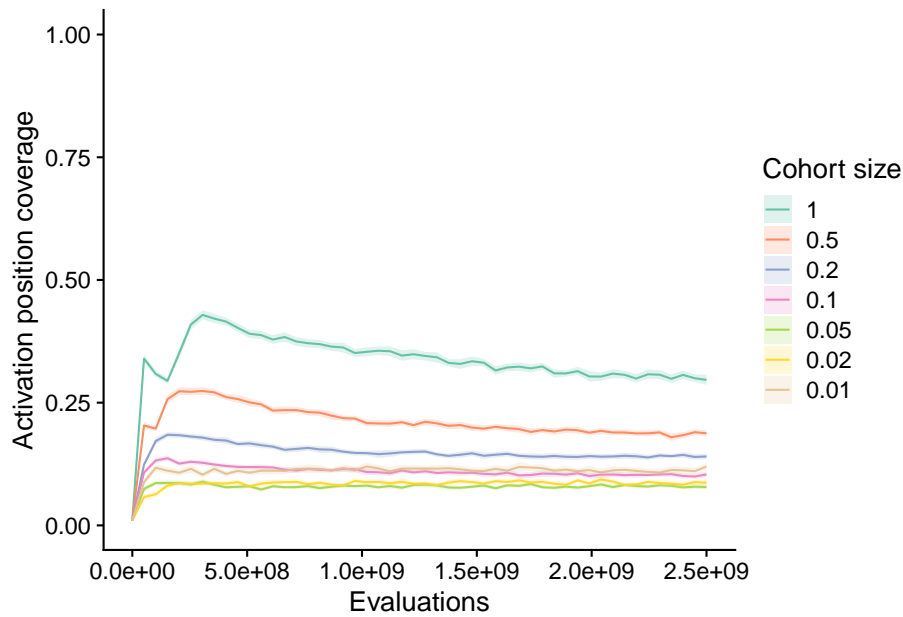
9.5 Unique starting positions

```

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
```

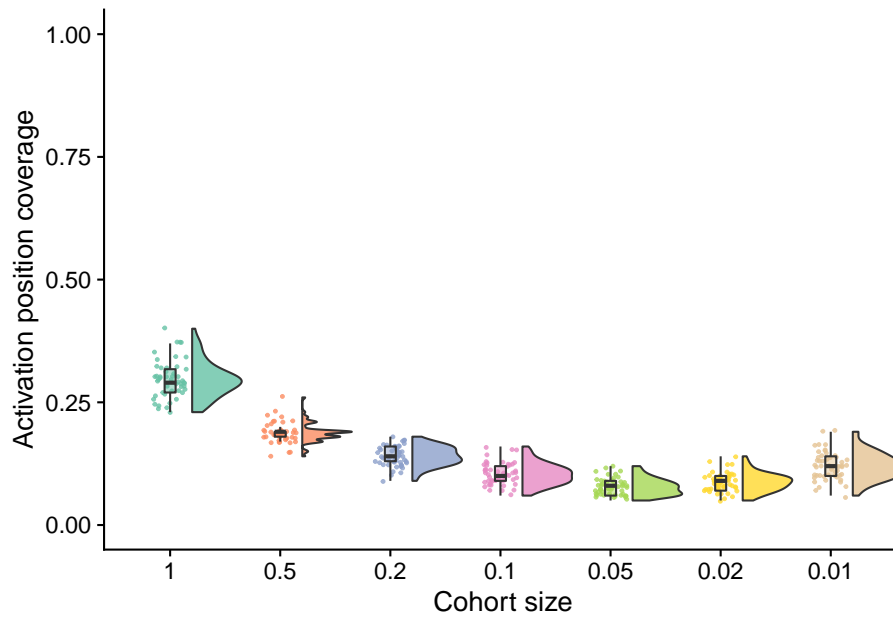
9.5.1 Final starting position coverage

```
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,

```

```

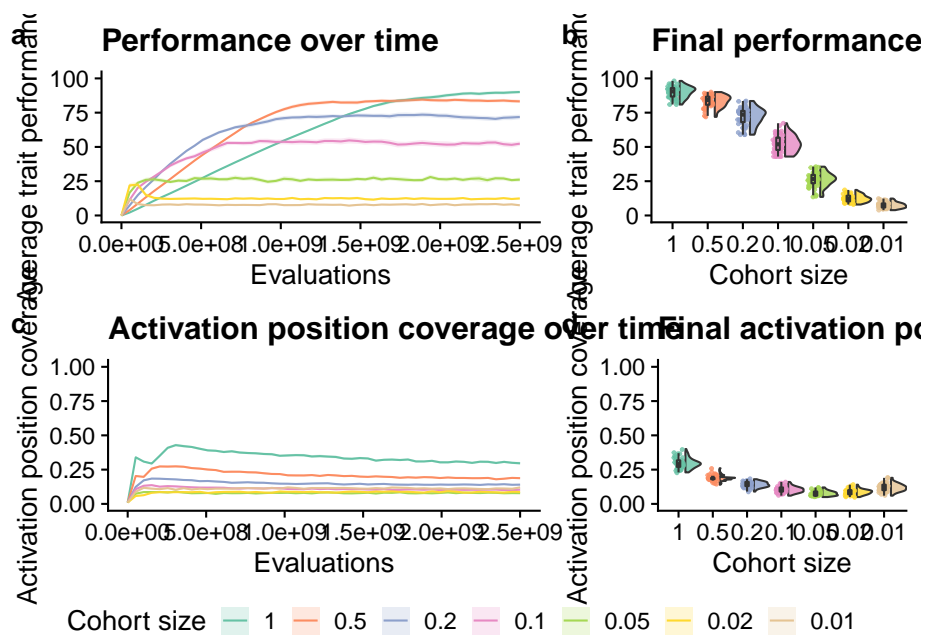
base_width=12,
base_height=8
)

```

```

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(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
```

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

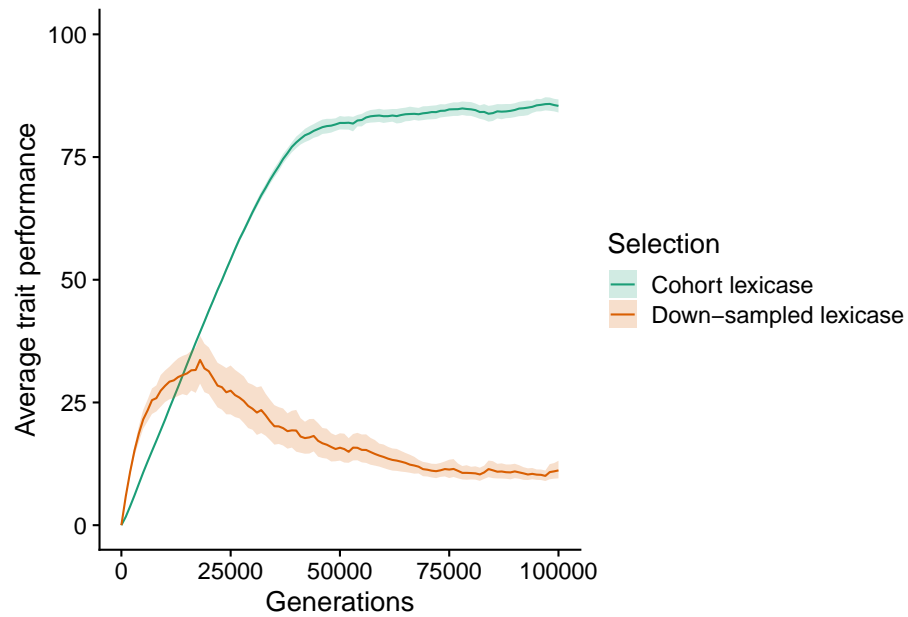
final_data <- filter(data, evaluations==max(data$evaluations))
```

```
##### 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
```

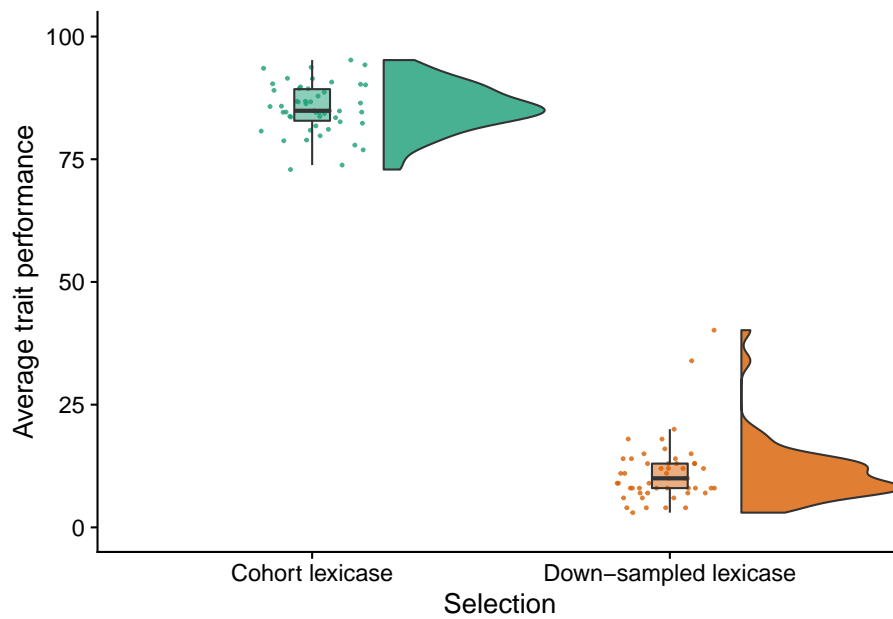


10.4.1 Final performance

```
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
```



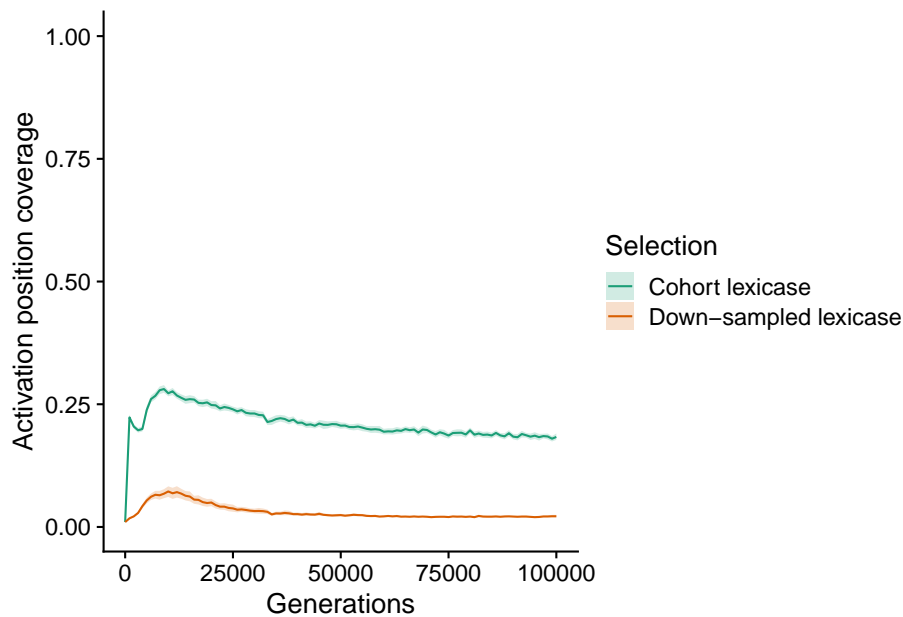
10.5 Unique starting positions

```
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

```

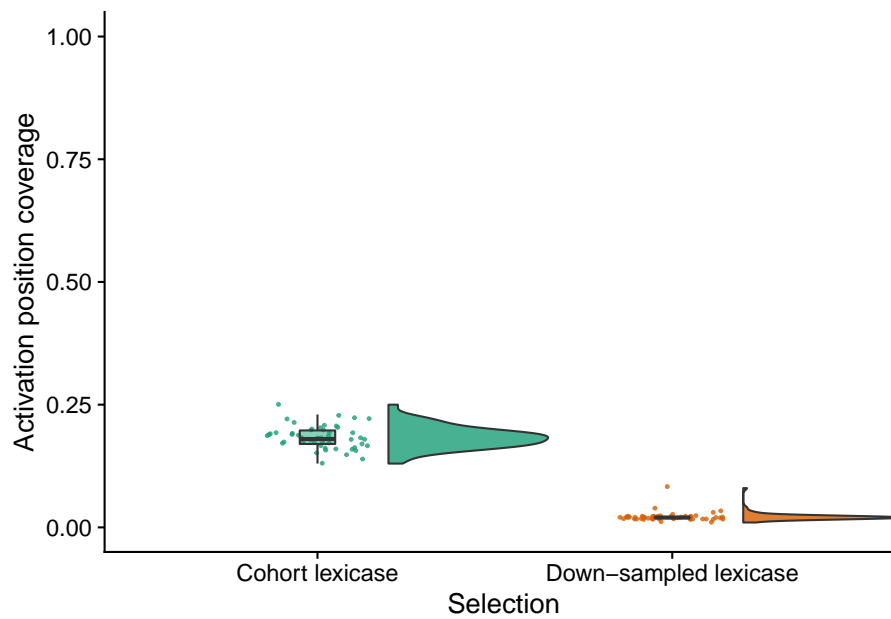
unique_start_positions_coverage_final_fig <- ggplot(
  final_data,
  aes(
    x=selection_name,

```

```

        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

```



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 +

```

```

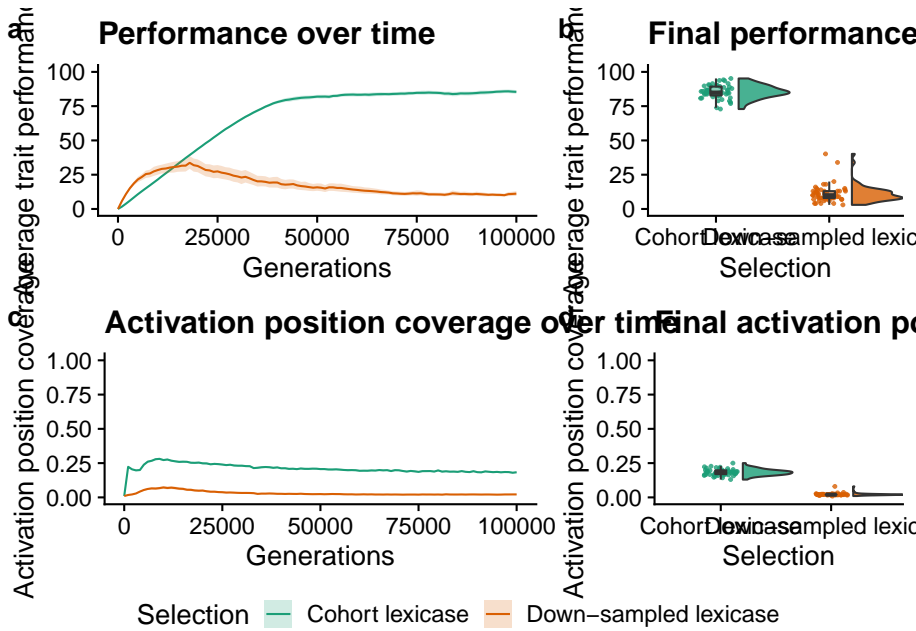
      ggtitle("Activation position coverage over time") +
      theme(legend.position="none"),
unique_start_positions_coverage_final_fig +
      ggtitle("Final activation position coverage") +
      theme(),
nrow=2,
ncol=2,
rel_widths=c(3,2),
labels="auto"
)

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

```



Chapter 11

Novelty lexibase

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
)

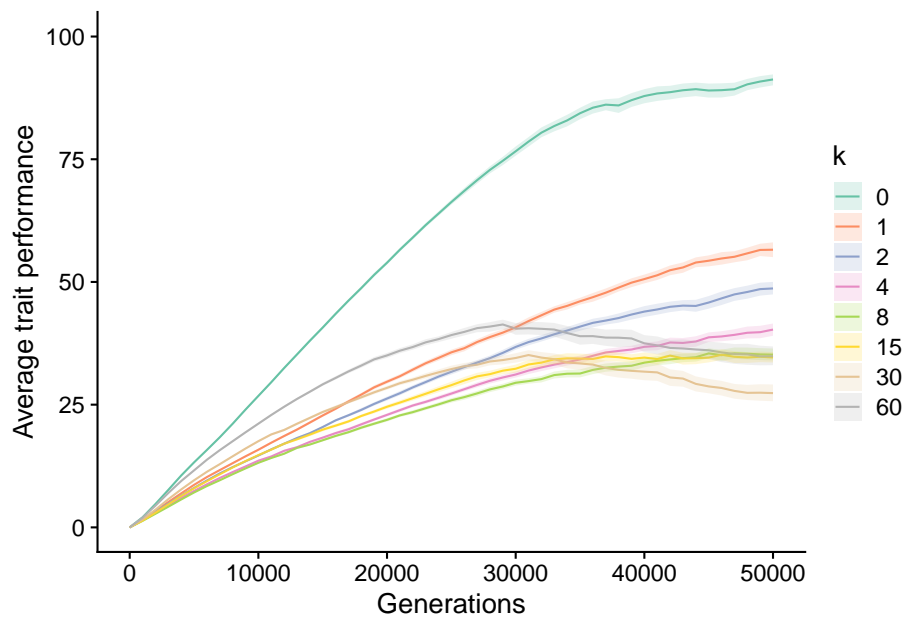
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

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
```



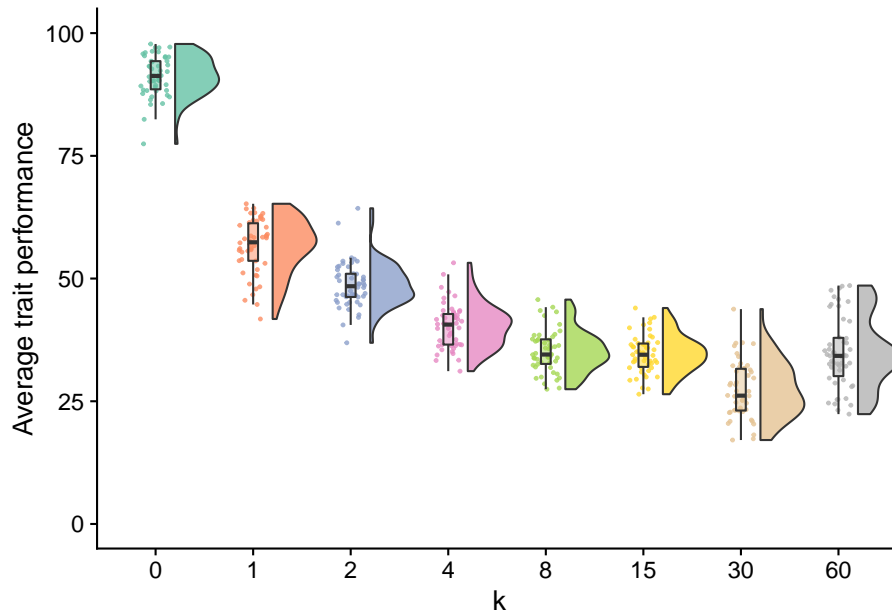
11.4.1 Final performance

```
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

```



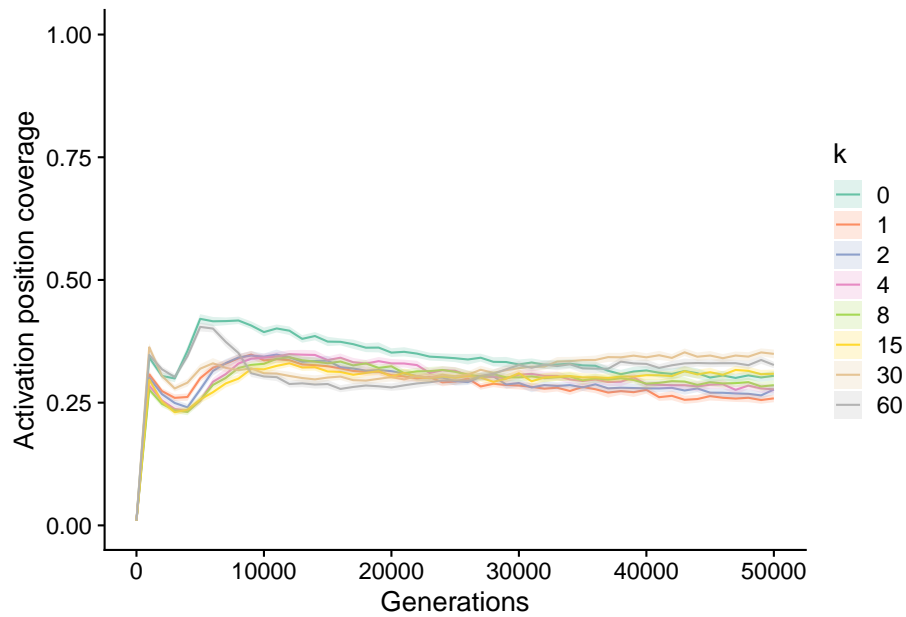
11.5 Unique starting positions

```

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
```



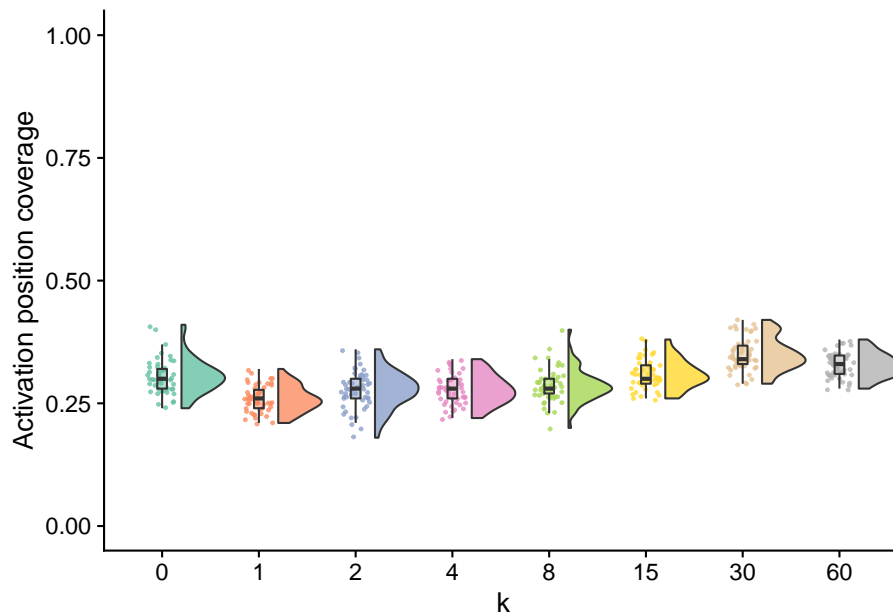
11.5.1 Final starting position coverage

```
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"
    )
)

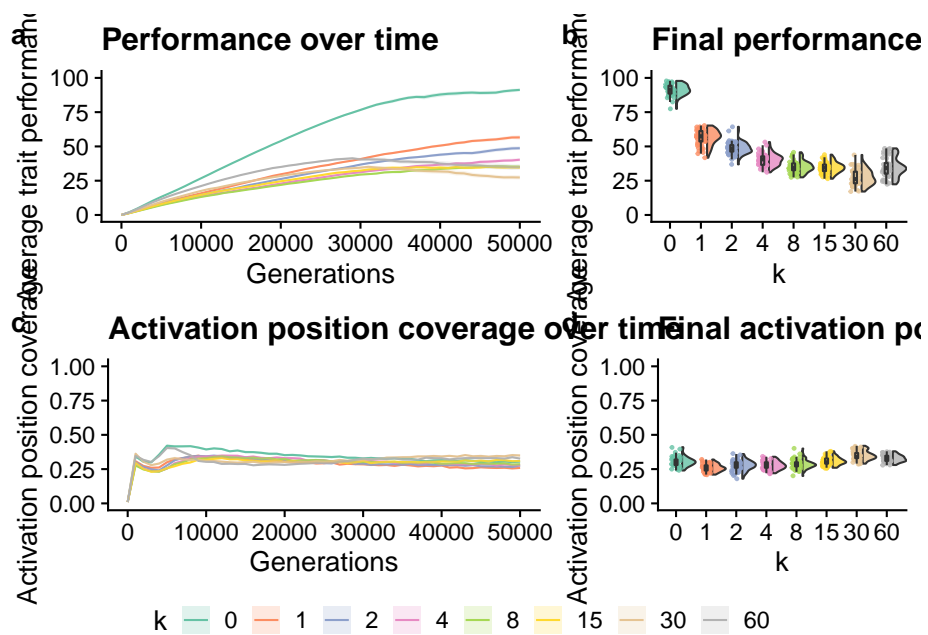
grid <- plot_grid(
  elite_ave_performance_fig +
    ggtitle("Performance over time") +
    theme(legend.position="none"),
  elite_final_performance_fig +
    ggtitle("Final performance") +
    theme(),
  unique_start_position_coverage_fig +
    ggtitle("Activation position coverage over time") +
  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

```



Bibliography

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