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

2021-06-16

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

Diagnostic cardinality

4.1 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)
```

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
```

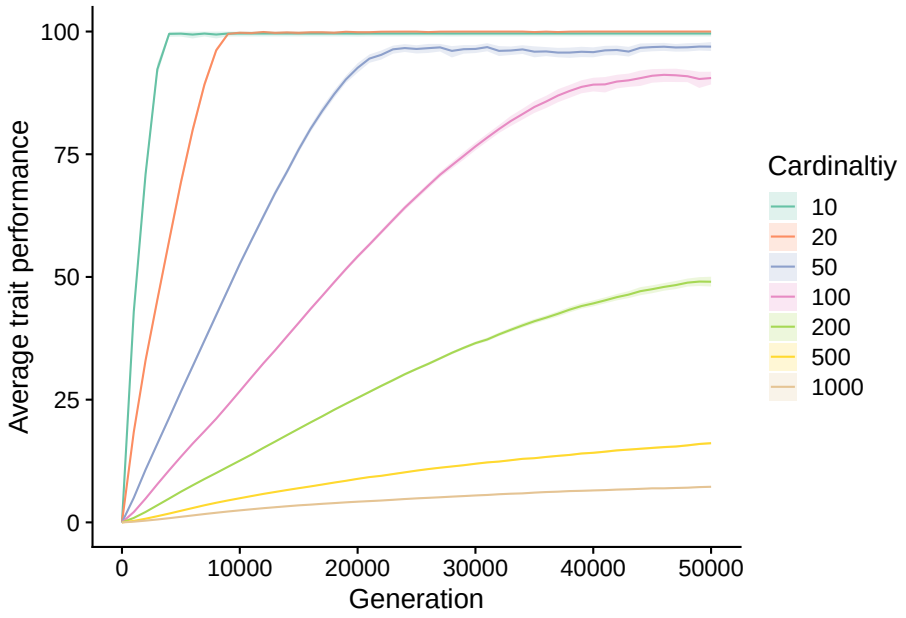
```
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          4
## minor          1.0
## year           2021
## month          05
## day            18
## svn rev        80317
## language       R
## version.string R version 4.1.0 (2021-05-18)
## nickname       Camp Pontanezen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

```
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())
```

4.4 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
```

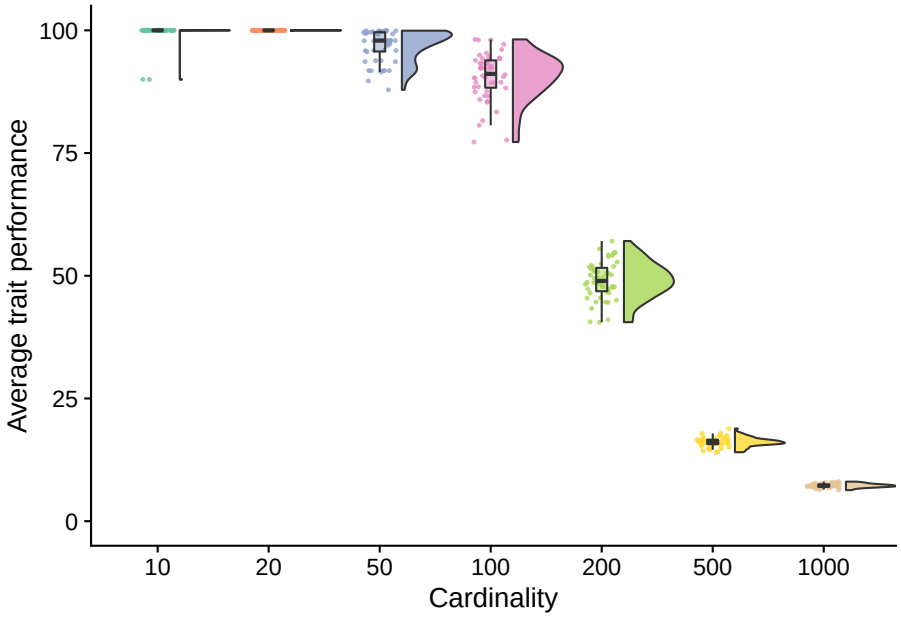


4.4.1 Final performance

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
  ) +
```

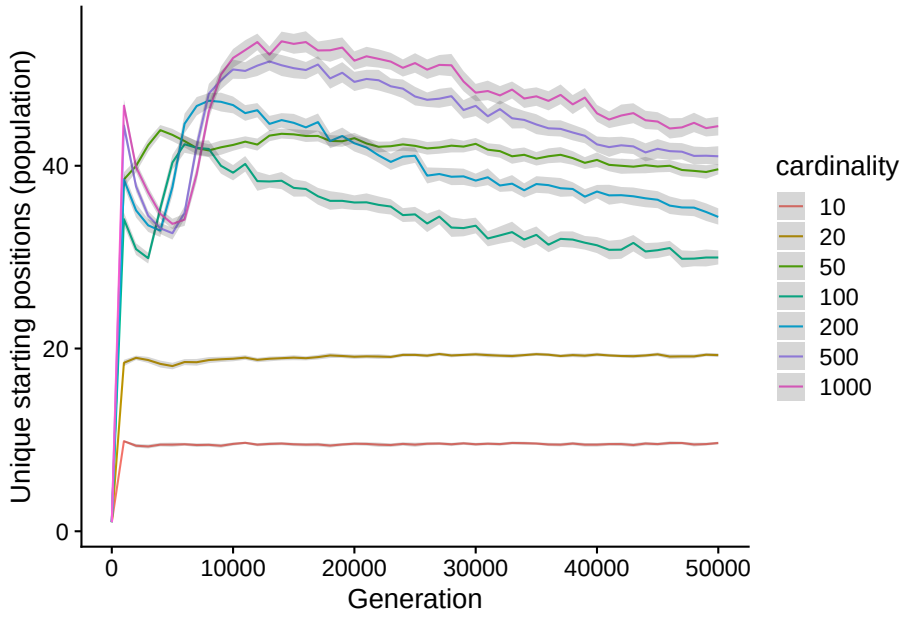
```
  scale_y_continuous(
    name="Average trait performance",
    limits=c(0, 100)
  ) +
  scale_x_discrete(
    name="Cardinality"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
elite_trait_ave_fit_final
```



4.5 Unique starting positions

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

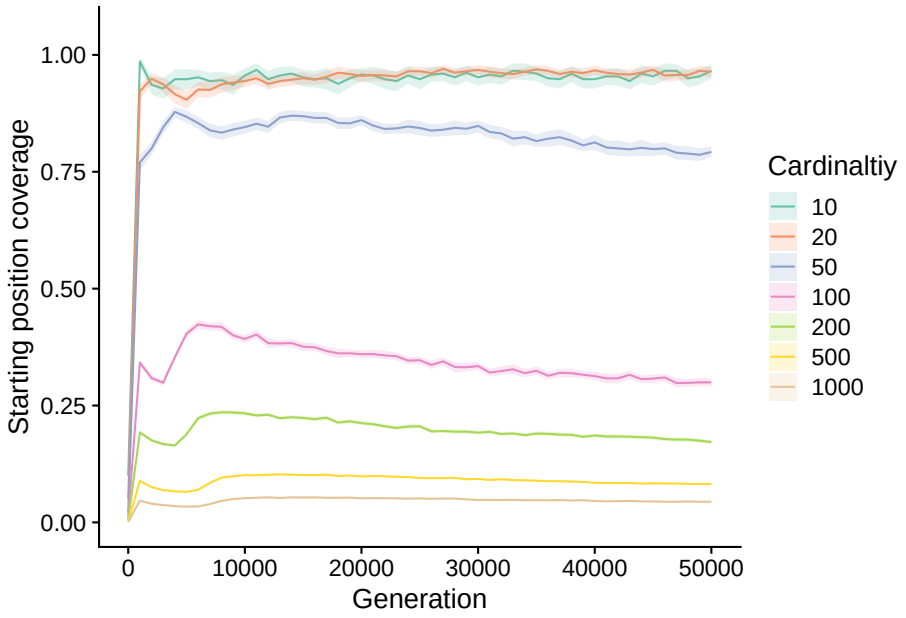
```
ggplot(data, aes(x=gen, y=uni_str_pos, color=cardinality)) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Unique starting positions (population)",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

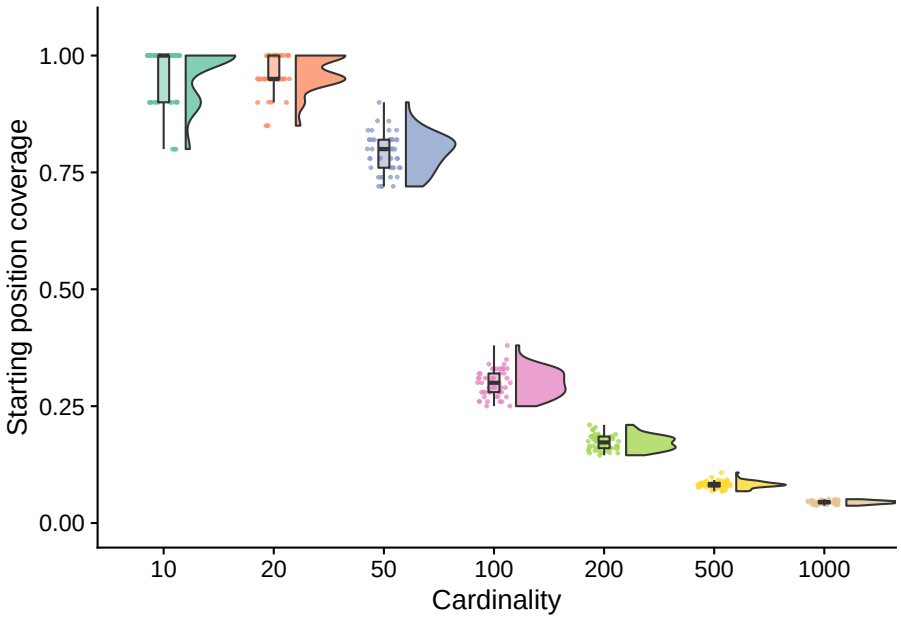
```
      color=cardinality,
      fill=cardinality
    )
  ) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),
    alpha=0.2,
    linetype=0
  ) +
  scale_y_continuous(
    name="Starting position coverage",
    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
```



4.5.1 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,
```

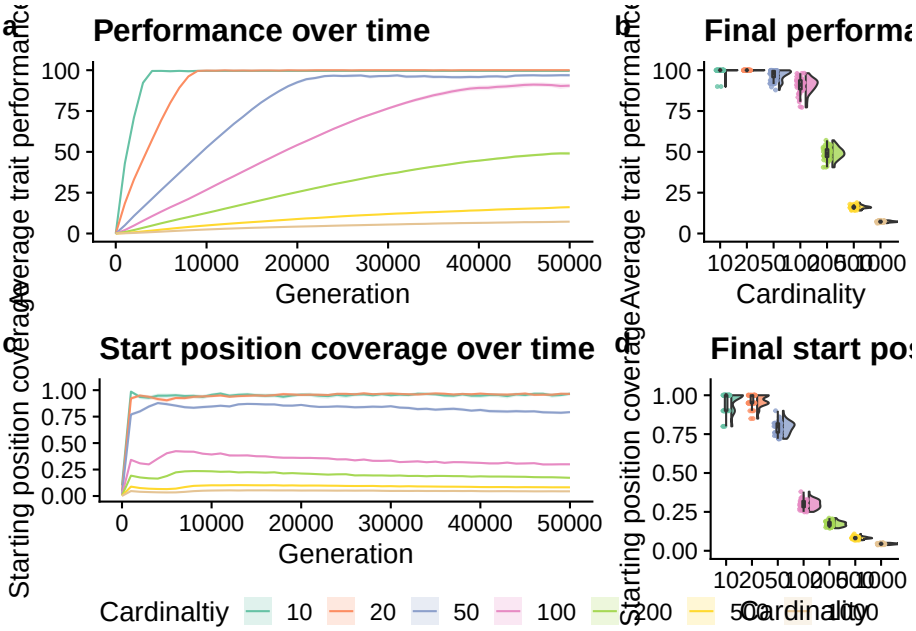
```
    alpha = 0.5
  ) +
  scale_y_continuous(
    name="Starting position coverage",
    limits=c(0, 1.05)
  ) +
  scale_x_discrete(
    name="Cardinality"
  ) +
  scale_fill_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  scale_color_brewer(
    name="Cardinaltiy",
    palette=cb_palette
  ) +
  theme(
    legend.position="none"
  )
final_unique_start_positions_coverage_fig
```



4.6 Manuscript figures

Combine figures for the manuscript.

```
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("Start position coverage over time") +
    guides(color=guide_legend(nrow = 1), fill=guide_legend(nrow=1)) +
    theme(
      legend.position="bottom",
      legend.box="horizontal"
    ),
  final_unique_start_positions_coverage_fig +
    ggtitle("Final start position coverage") +
    theme(),
  nrow=2,
  ncol=2,
  rel_widths=c(2,1),
  labels="auto"
)

save_plot(
  paste(working_directory, "imgs/cardinality-panel.pdf", sep=""),
  grid,
  base_width=12,
  base_height=10
)

grid
```


A
Performance over time
Average trait performance
100
75
50
25
0
0
10000
20000
30000
40000
50000
Generation
B
Final performa
Average trait performanc
Cardinality
C
Start position coverage over time
Starting position coverage
1.00
0.75
0.50
0.25
0.00
Generation
D
Final start po
Cardinality
Cardinaltiy
10
20
50
100
200
500
1000

Bibliography

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