Replaying the evolution of learning

Austin Ferguson

2022-08-23

Contents

Chapter 1

Introduction

Based on Alex Lalejini's wonderful example: https://github.com/amlalejini/auto-deploying-bookdown-example.

Chapter 2

Initial run - two cues

This experiment is located in `experiments/exploratory/2022_07_27__two_cues_phylo`.

The experiment aimed to see what behaviors evolved in an two-cue environment with naive parameters.

2.1 Dependencies

```
# External
library(ggplot2)
library(dplyr)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9
# Internal
source('../global_shared_files/constant_vars__two_cues.R')
source('../global_shared_files/shared_funcs__two_cues.R')
```

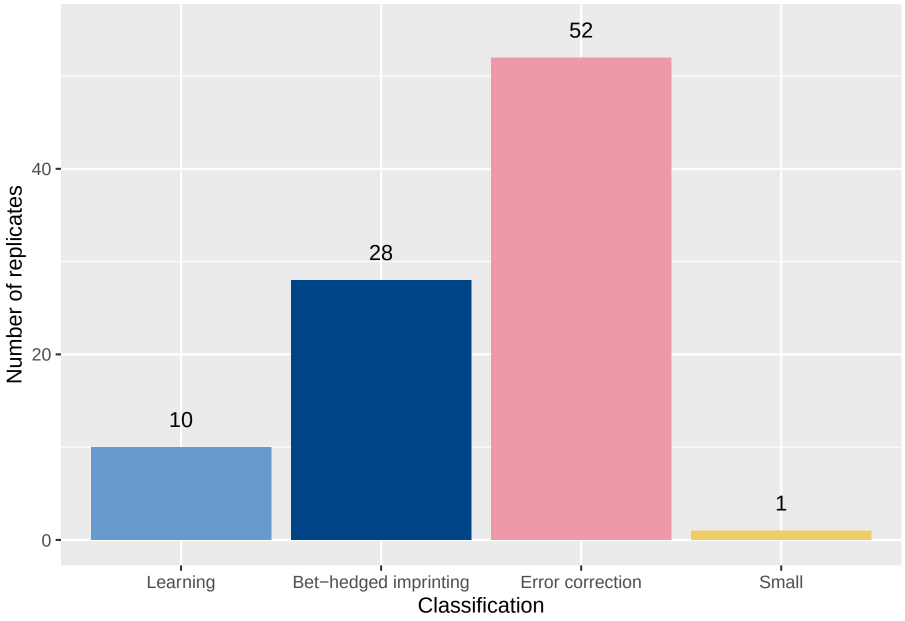
2.2 Data prep

```
df = read.csv('../experiments/exploration/2022_07_27__two_cues_phylo/data/combined_final_dominant
df = classify_individual_trials(df)
df = classify_seeds(df)
df_summary = summarize_final_dominant_org_data(df)
classification_summary = summarize_classifications(df_summary)
```

2.3 Classification summary

How many replicates evolved each behavior?

```
ggplot(classification_summary, aes(x = seed_classification_factor, y = count, fill = se
  geom_col() +
  geom_text(aes(y = count + 3, label = count)) +
  scale_fill_manual(values = color_map) +
  xlab('Classification') +
  ylab('Number of replicates') +
  theme(legend.position = 'none')
```



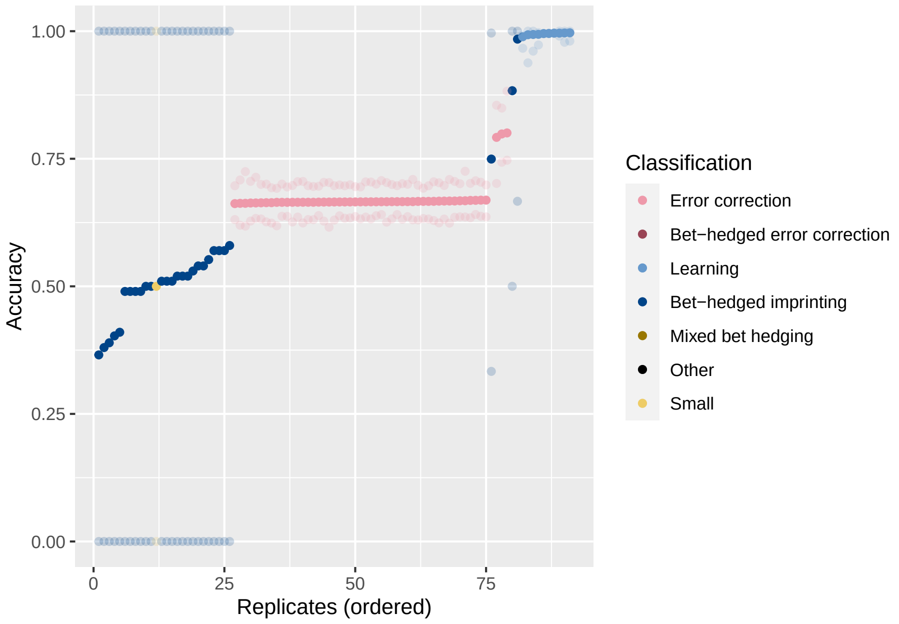
2.4 Accuracy plots

Order the points by mean accuracy

```
df_summary = df_summary[order(df_summary$accuracy_mean),]
df_summary$seed_order = 1:nrow(df_summary)
df$seed_order = NA
for(seed in unique(df$seed)){
  df[df$seed == seed,]$seed_order = df_summary[df_summary$seed == seed,]$seed_order
}
```

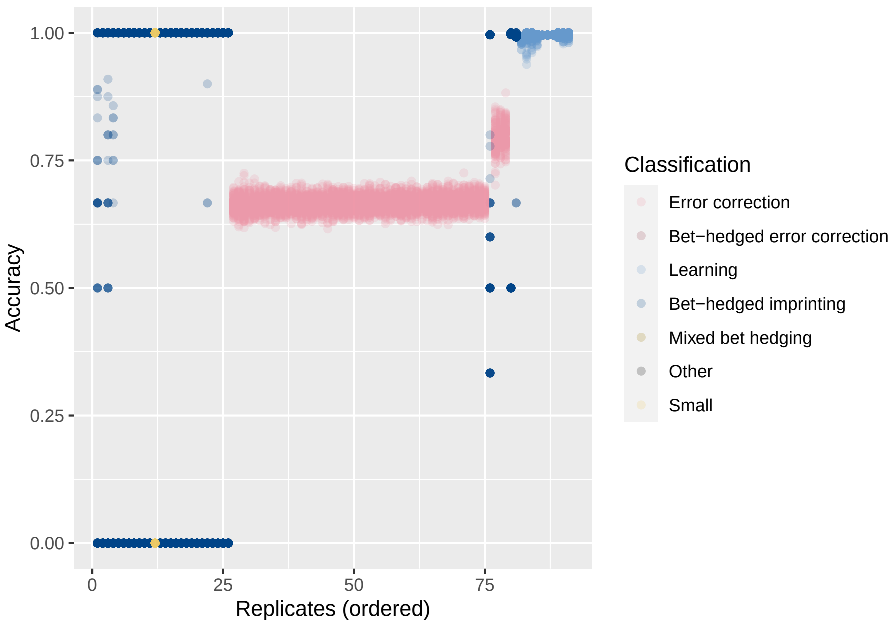
For each replicate, plot mean accuracy as a solid point and min/max as partially transparent points.

```
ggplot(df_summary, aes(x = seed_order, color = seed_classification)) +
  geom_point(aes(y = accuracy_mean)) +
  geom_point(aes(y = accuracy_min), alpha = 0.2) +
  geom_point(aes(y = accuracy_max), alpha = 0.2) +
  scale_color_manual(values = color_map) +
  xlab('Replicates (ordered)') +
  ylab('Accuracy') +
  labs(color = 'Classification')
```

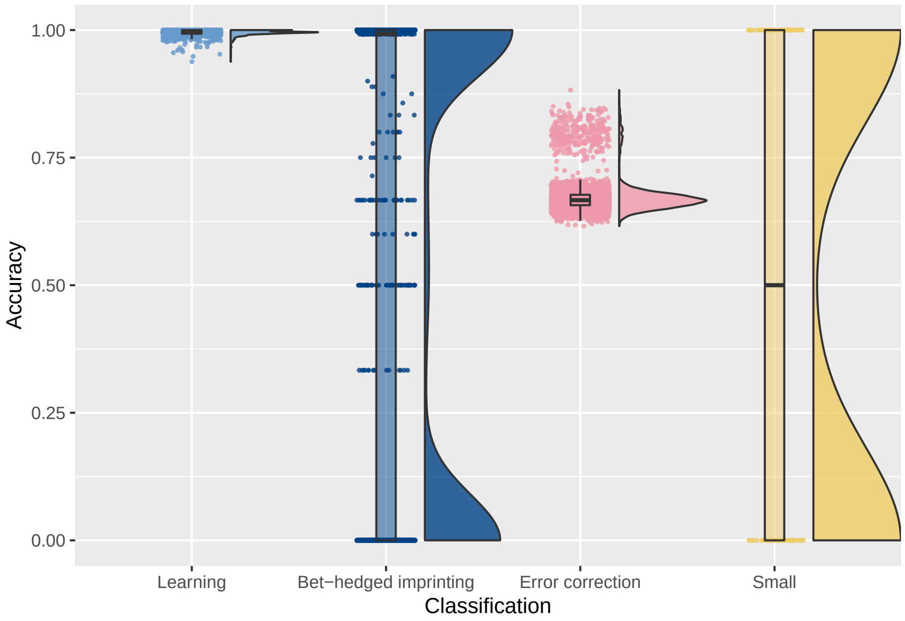


Now plot the accuracy of each sample, maintaining the same order on the x-axis.

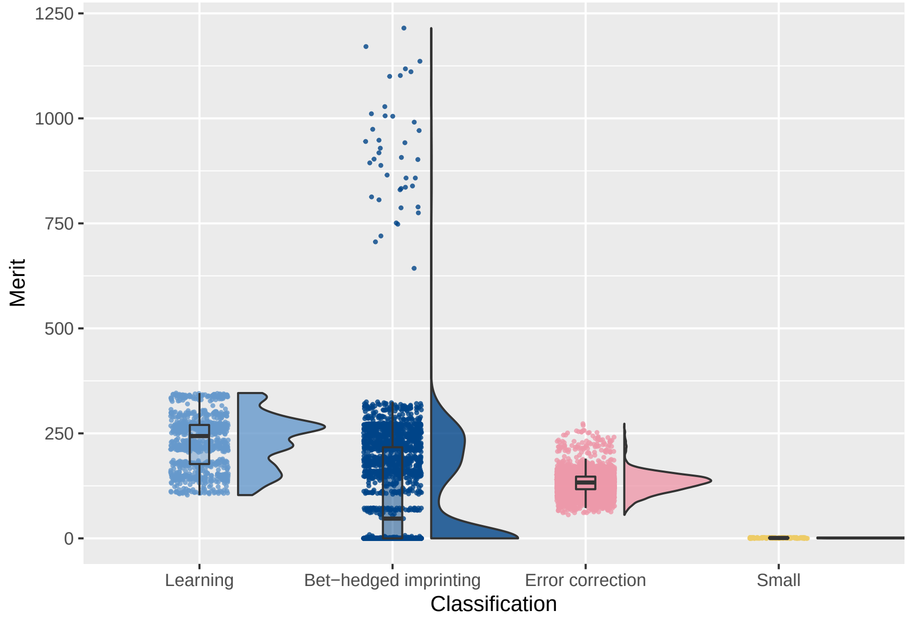
```
ggplot(df, aes(x = seed_order, y = accuracy, color = seed_classification)) +
  geom_point(alpha = 0.2) +
  scale_color_manual(values = color_map) +
  xlab('Replicates (ordered)') +
  ylab('Accuracy') +
  labs(color = 'Classification')
```



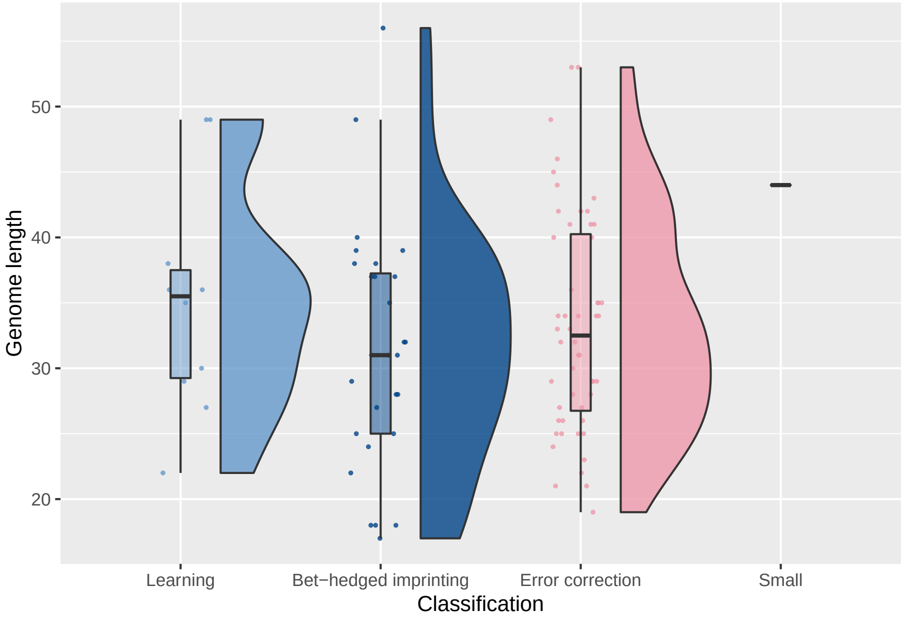
```
# Raincloud plot of accuracy
ggplot(df, aes(x = seed_classification_factor, y = accuracy, fill = seed_classificatio
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Accuracy') +
  theme(legend.position = 'none')
```



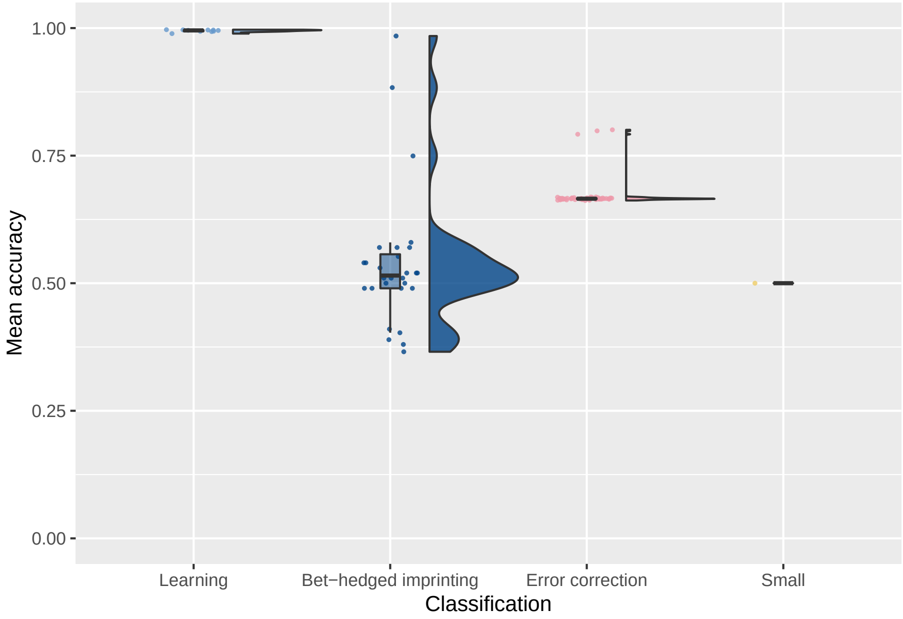
```
# Raincloud plot of merit
ggplot(df, aes(x = seed_classification_factor, y = merit, fill = seed_classification_factor)) +
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8 ) +
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter(width = .1
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Merit') +
  theme(legend.position = 'none')
```



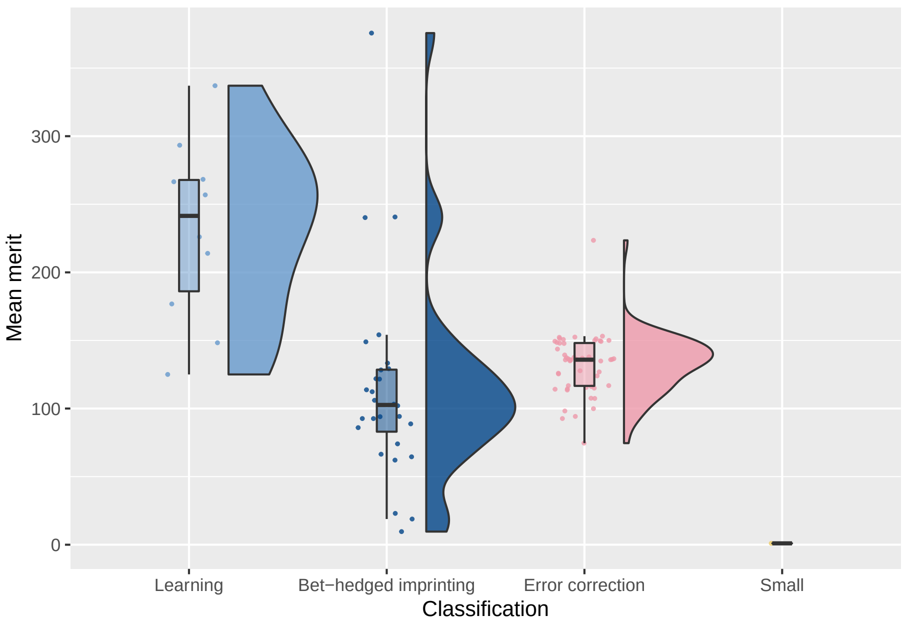
```
# Raincloud plot of genome length
ggplot(df_summary, aes(x = seed_classification_factor, y = genome_length, fill = seed_
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Genome length') +
  theme(legend.position = 'none')
```



```
# Raincloud plot of mean accuracy of replicates
ggplot(df_summary, aes(x = seed_classification_factor, y = accuracy_mean, fill = seed_classificat
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8 ) +
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter(width = .1
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  scale_y_continuous(limits = c(0,1)) +
  xlab('Classification') +
  ylab('Mean accuracy') +
  theme(legend.position = 'none')
```



```
# Raincloud plot of mean merit of replicates
ggplot(df_summary, aes(x = seed_classification_factor, y = merit_mean, fill = seed_clas
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Mean merit') +
  theme(legend.position = 'none')
```


Mean merit
0
100
200
300
Learning
Bet−hedged imprinting
Error correction
Small
Classification

Chapter 3

Initial run - three cues, one set

This experiment is located in `experiments/exploratory/2022_07_28__three_cues_one_set`.

The experiment aimed to see what behaviors evolved in an environment with three cues (with one being set to always be 1) and with naive parameters.

3.1 Dependencies

```
# External
library(ggplot2)
library(dplyr)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9
# Internal
source('../global_shared_files/constant_vars__three_cues_one_set.R')
source('../global_shared_files/shared_funcs__three_cues_one_set.R')
```

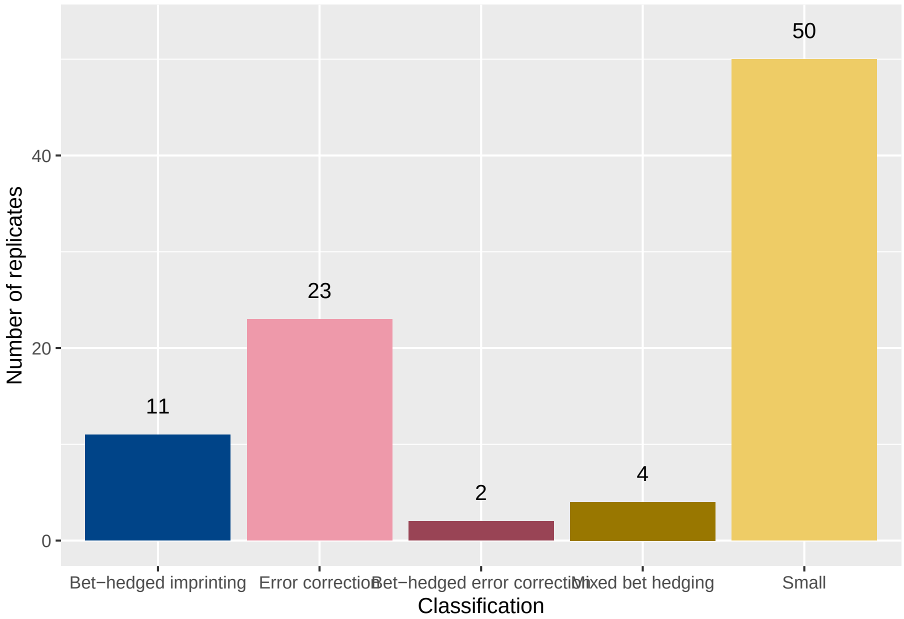
3.2 Data prep

```
df = read.csv('../experiments/exploration/2022_07_28__three_cues_one_set/data/combined_final_domi
df = classify_individual_trials(df)
df = classify_seeds(df)
df_summary = summarize_final_dominant_org_data(df)
classification_summary = summarize_classifications(df_summary)
```

3.3 Classification summary

How many replicates evolved each behavior?

```
ggplot(classification_summary, aes(x = seed_classification_factor, y = count, fill = se
  geom_col() +
  geom_text(aes(y = count + 3, label = count)) +
  scale_fill_manual(values = color_map) +
  xlab('Classification') +
  ylab('Number of replicates') +
  theme(legend.position = 'none')
```



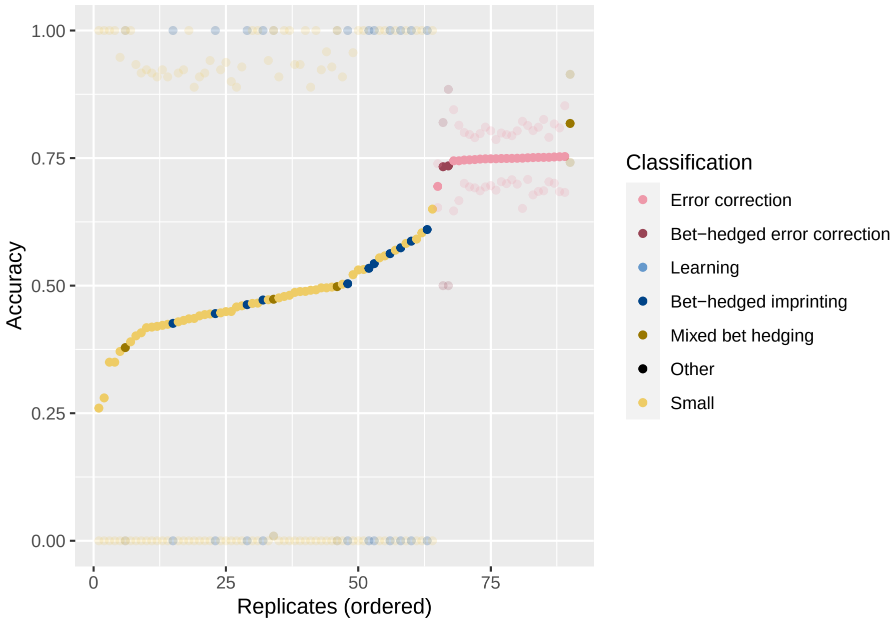
3.4 Accuracy plots

Order the points by mean accuracy

```
df_summary = df_summary[order(df_summary$accuracy_mean),]
df_summary$seed_order = 1:nrow(df_summary)
df$seed_order = NA
for(seed in unique(df$seed)){
  df[df$seed == seed,]$seed_order = df_summary[df_summary$seed == seed,]$seed_order
}
```

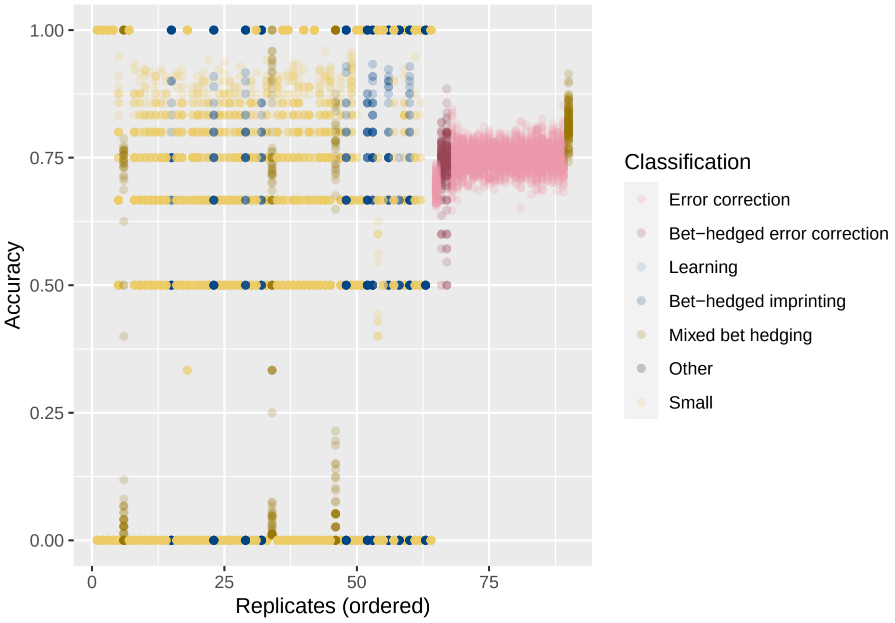
For each replicate, plot mean accuracy as a solid point and min/max as partially transparent points.

```
ggplot(df_summary, aes(x = seed_order, color = seed_classification)) +
  geom_point(aes(y = accuracy_mean)) +
  geom_point(aes(y = accuracy_min), alpha = 0.2) +
  geom_point(aes(y = accuracy_max), alpha = 0.2) +
  scale_color_manual(values = color_map) +
  xlab('Replicates (ordered)') +
  ylab('Accuracy') +
  labs(color = 'Classification')
```

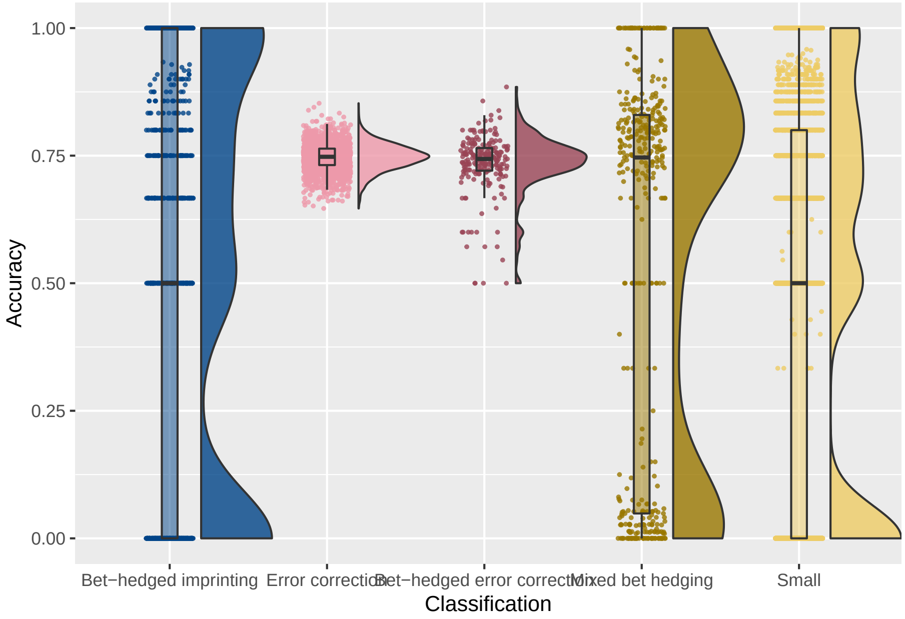


Now plot the accuracy of each sample, maintaining the same order on the x-axis.

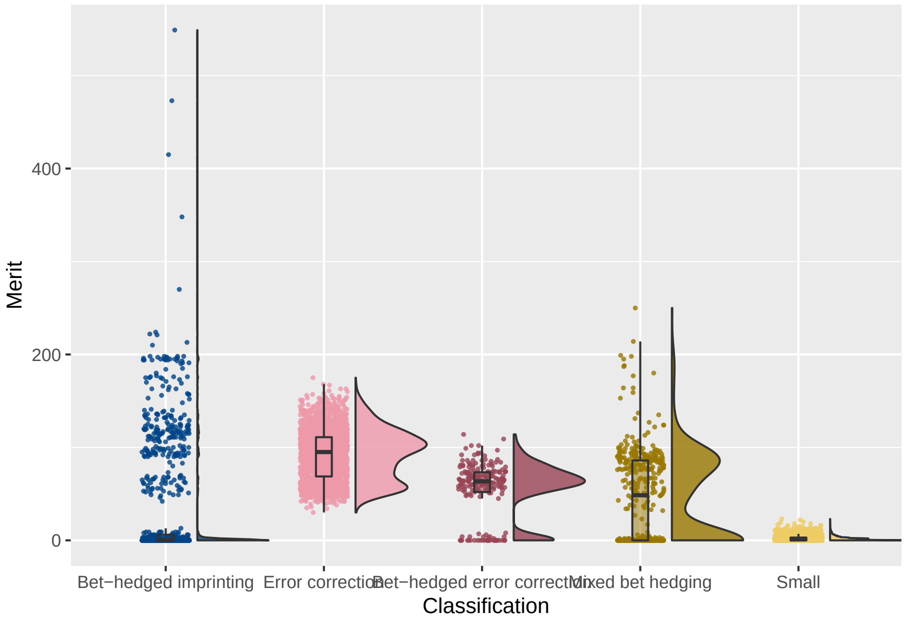
```
ggplot(df, aes(x = seed_order, y = accuracy, color = seed_classification)) +
  geom_point(alpha = 0.2) +
  scale_color_manual(values = color_map) +
  xlab('Replicates (ordered)') +
  ylab('Accuracy') +
  labs(color = 'Classification')
```



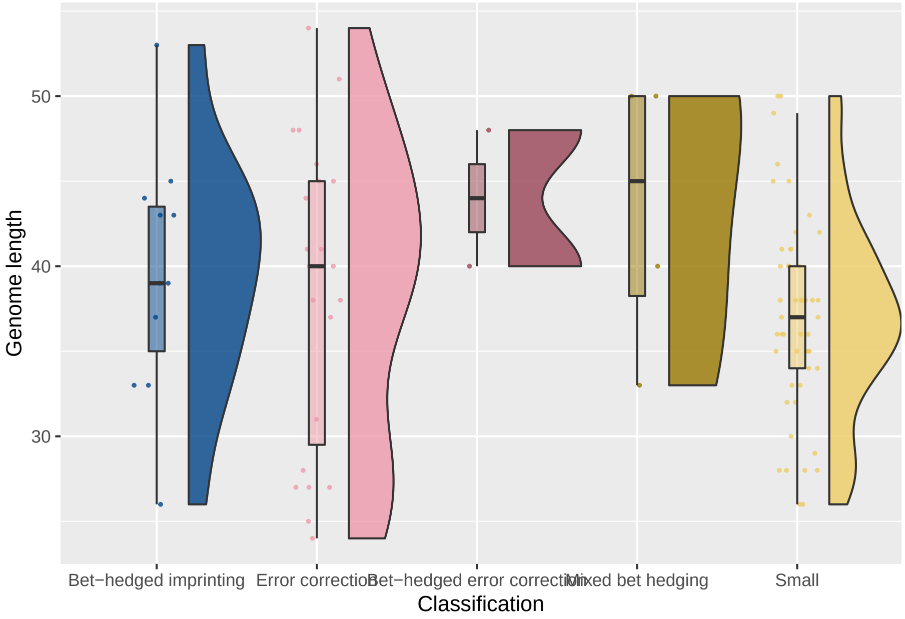
```
# Raincloud plot of accuracy
ggplot(df, aes(x = seed_classification_factor, y = accuracy, fill = seed_classificatio
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Accuracy') +
  theme(legend.position = 'none')
```



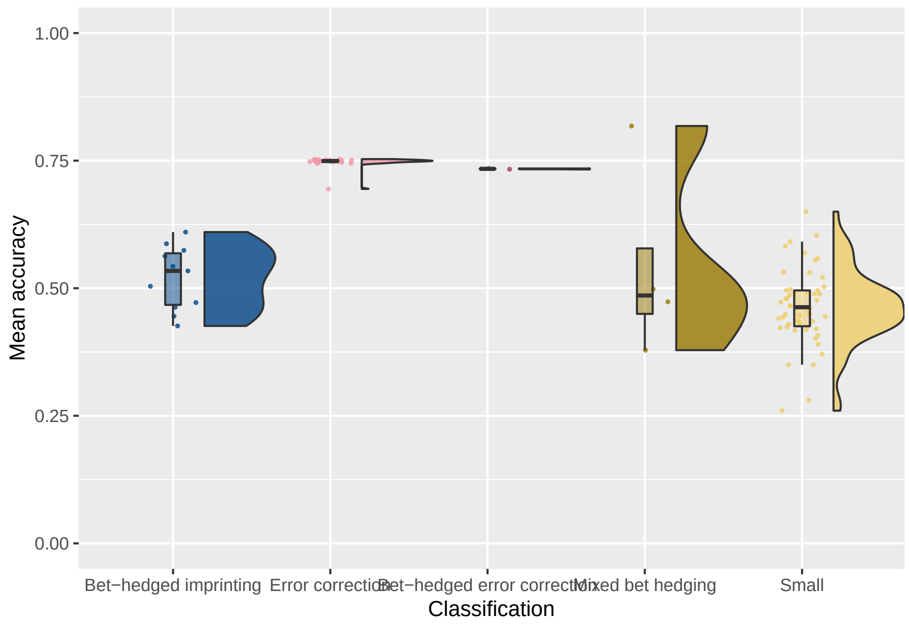
```r
# Raincloud plot of merit
ggplot(df, aes(x = seed_classification_factor, y = merit, fill = seed_classification_factor)) +
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8 ) +
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter(width = .1
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Merit') +
  theme(legend.position = 'none')
```



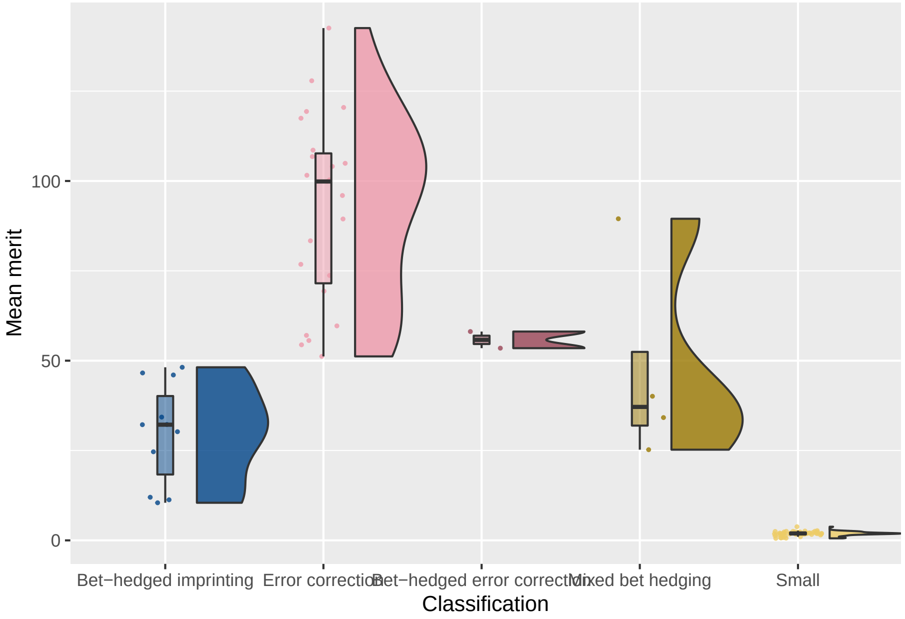
```
# Raincloud plot of genome length
ggplot(df_summary, aes(x = seed_classification_factor, y = genome_length, fill = seed_
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Genome length') +
  theme(legend.position = 'none')
```



```
# Raincloud plot of mean accuracy of replicates
ggplot(df_summary, aes(x = seed_classification_factor, y = accuracy_mean, fill = seed_classificat
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8 ) +
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter(width = .1
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  scale_y_continuous(limits = c(0,1)) +
  xlab('Classification') +
  ylab('Mean accuracy') +
  theme(legend.position = 'none')
```



```r
# Raincloud plot of mean merit of replicates
ggplot(df_summary, aes(x = seed_classification_factor, y = merit_mean, fill = seed_clas
  geom_flat_violin(scale="width", position = position_nudge(x = .2, y = 0), alpha = .8
  geom_point(mapping=aes(color=seed_classification_factor), position = position_jitter
  geom_boxplot( width = .1, outlier.shape = NA, alpha = 0.5 ) +
  scale_fill_manual(values = color_map) +
  scale_color_manual(values = color_map) +
  xlab('Classification') +
  ylab('Mean merit') +
  theme(legend.position = 'none')
```



Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Müller, Kirill, and Hadley Wickham. 2022. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2022a. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

———. 2022b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2022c. *Tidyverse: Easily Install and Load the Tidyverse*. https://CRAN .R-project.org/package=tidyverse.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington.

2022. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Maximilian Girlich. 2022. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Jim Hester, and Jennifer Bryan. 2022. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wilke, Claus O. 2020. *Cowplot: Streamlined Plot Theme and Plot Annotations for Ggplot2*. https://wilkelab.org/cowplot/.

Xie, Yihui. 2016. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/bookdown.

———. 2022. *Bookdown: Authoring Books and Technical Documents with R Markdown*. https://CRAN.R-project.org/package=bookdown.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2019. *R Markdown: The Definitive Guide*. Boca Raton: CRC Press, Taylor; Francis Group.