

Supplemental info about phylodiversity in evolutionary computation

Jose Guadalupe Hernandez, Alexander Lalejini, Emily Dolson

2021-09-08

Contents

1	Introduction	5
1.1	About our supplemental material	5
1.2	Contributing authors	5
1.3	Research overview	5
1.4	Reproducing our work	8
2	Exploration Diagnostic	11
2.1	Setup	11
2.2	Performance	13
2.3	Phylogenetic diversity	21
2.4	Phenotypic diversity	48
2.5	Relationship between phenotypic and phylogenetic diversity	57
2.6	Relationship between diversity and success	64
2.7	Causality analysis	79
3	Other fitness landscapes	111
3.1	Setup	111
3.2	Performance	113
3.3	Phylogenetic diversity	116
3.4	Phenotypic diversity	132
3.5	Relationship between phenotypic and phylogenetic diversity	136
3.6	Relationship between diversity and success	137
3.7	Causality analysis	144

Chapter 1

Introduction

This is the supplemental material for our work, ‘What can phylogenetic metrics tell us about useful diversity in evolutionary algorithms?’. This is not intended as a stand-alone document, but as a companion to our paper.

1.1 About our supplemental material

As you may have noticed (unless you’re reading a pdf version of this), our supplemental material is hosted using GitHub pages. We compiled our data analyses and supplemental documentation into this nifty web-accessible book using bookdown.

The source code/configuration files for this supplemental material and all experiments in the paper can be found in this GitHub repository.

1.2 Contributing authors

- Jose Guadalupe Hernandez
- Alexander Lalejini
- Emily Dolson

1.3 Research overview

1.3.1 Abstract

It is generally accepted that “diversity” is associated with success in evolutionary algorithms. However, diversity is a broad concept that can be measured and defined in a multitude of ways. To date, most evolutionary computation research has measured diversity using the

richness and/or evenness of a particular genotypic or phenotypic property. While these metrics are informative, we hypothesize that other diversity metrics are more strongly predictive of success. Phylogenetic diversity metrics are a class of metrics popularly used in biology, which take into account the evolutionary history of a population. Here, we investigate the extent to which 1) these metrics provide different information than those traditionally used in evolutionary computation, and 2) these metrics better predict the long-term success of a run of evolutionary computation. We find that, in most cases, phylogenetic metrics behave meaningfully differently from other diversity metrics. Moreover, our results suggest that phylogenetic diversity is indeed a better predictor of success.

1.3.2 Phenotypic diversity vs phylogenetic diversity.

In short, phenotypic diversity measures the diversity of phenotypes in the population at any one point in time. Phylogenetic diversity measures the diversity of evolutionary history represented in a population. We wrote a lot more about building phylogenies in the context of computational evolution in this paper.

As an example, the following figure shows two different phylogenies (ancestry trees). Arrows show parent-child relationships. Each node is a taxonomically unique phenotype (i.e., a phenotype with a unique evolutionary origin). For simplicity, leaf nodes in these diagrams are assumed to be the current set of taxa in the population; in reality, there could be non-leaf nodes corresponding to extant taxa. A) A population with high phenotypic diversity (phenotypic richness = 5) and low phylogenetic diversity (mean pairwise distance = 2). B) A population with low phenotypic diversity (phenotypic richness = 2) and high phylogenetic diversity (mean pairwise distance = 6).

1.3.3 Research questions

1. Is phylogenetic diversity meaningfully different from phenotypic diversity in the context of evolutionary computation?

The answer to this question is important. Intuitively, we might think that since these are both types of diversity, they should correlate pretty closely. Given that phylogenetic diversity is more computationally intensive to measure, if we're going to argue that it's something evolutionary computation researchers should pay attention to (spoilers: we are!), we need to show that it is meaningfully different.

2. Is phylogenetic diversity more informative about outcomes in evolutionary computation than phenotypic diversity?

The importance of this question is more obvious. We know that diversity is centrally linked to the success of evolutionary algorithms. There are hints

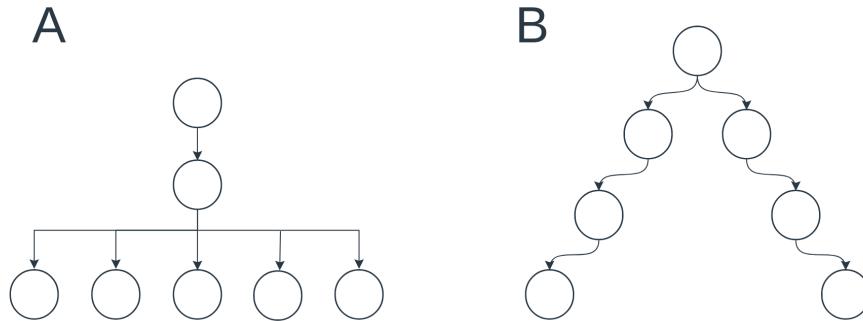


Figure 1.1: Example of populations with different levels of phenotypic and phylogenetic diversity

scattered across the literature that certain types of diversity are more “useful” to solving problems than others. So our goal is for this work to move us towards a better understanding of which types of diversity we should be promoting in evolutionary algorithms.

1.3.4 Study design

We ran 5 selection schemes (random, tournament, fitness sharing, lexicase selection, and eco-ea) on 5 different problems (one designed to be a clean test environment, and 4 chosen to evoke the messy realities of real problems) and gathered a ludicrous amount of data. Here and in the paper, we attempt to focus very closely on getting answers to the two specific questions that we asked above (to avoid overwhelming ourselves or the reader with a firehose of data). There are many intriguing aspects of this data set that raise further questions, which we look forward to addressing in the future.

1.3.5 Results

1. Phylogenetic diversity and phenotypic diversity behave differently to an extent that was even surprising to us.
2. Phylogenetic diversity is more predictive of success than phenotypic diversity in the vast majority of cases. The differences are often substantial (check out our effect sizes!).

1.3.6 Caveats/areas for future research

- Phylogenetic diversity and phenotypic diversity are both broad classes of metrics, and there is substantial variation in how different phylogenetic diversity metrics behave in different contexts.
- There is clearly variation in all of this over time and by fitness landscape.

1.4 Reproducing our work

1.4.1 Data availability

All data used in the paper is available via the Open Science Framework.

1.4.2 Code availability

All code used in the paper is available on github.

1.4.3 Dependencies

The C++ code to run these experiments requires:

- Empirical - The EC Ecology toolbox

1.4.4 Compilation

You can compile and run the code used in the paper as follows:

```
# Clone Empirical
git clone --recursive https://github.com/devosoft/Empirical.git

# Clone EC-ecology-toolbox
git clone https://github.com/emilydolson/ec_ecology_toolbox.git

# Clone the repo for this project
git clone --recursive https://github.com/emilydolson/phylodiversity-metrics-in-EC-GPTP

### Complex fitness landscapes

# Compile the executable to run experiments for this project
make

# Run an experiment. To set parameters, use command line flags
# e.g. to set the selection scheme, run ./ecology_parameter_sweep -SELECTION 2
# To see all options, run ./ecology_parameter_sweep --help
./ecology_parameter_sweep

### Exploration diagnostic
```

```
# all of the code for the exploration diagnostic lives in the exploration_diagnostic submodule
cd exploration_diagnostic
make
./dia_world

# the dia_world executable can be configured in the same way as the ecology_parameter_sweep execu
```


Chapter 2

Exploration Diagnostic

2.1 Setup

First, we need to do some set up to analyze our data

Include dependencies

```
library(ggplot2)      # For plotting
library(tidyverse)    # For data wrangling
library(knitr)        # For making nice rmarkdown documents
library(cowplot)      # For theme
library(viridis)       # For color scale
library(RColorBrewer)  # For more color scales
library(rstatix)
library(ggsignif)     # For adding pairwise significance to plots
library(Hmisc)         # For bootstrapping confidence intervals
library(kableExtra)   # For displaying nice tables
source("https://gist.github.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9")
library(readr)          # For reading in data
library(stringr)       # For manipulating string data
library(ggpubr)         # For displaying correlation statistics on plots
library(infotheo)      # For causality analysis
library(scales)         # For displaying scales nicely in faceted plots
library(osfr)           # For downloading the data for this project
```

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              0.4
## year               2021
## month              02
## day                15
## svn rev            80002
## language           R
## version.string     R version 4.0.4 (2021-02-15)
## nickname          Lost Library Book

```

Setup constants to be used across plots

```

# 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 <- 1
coverage_ylim <- 1.0

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

```

The data for this project are hosted on osf.io. The following chunk downloads them automatically if they haven't already been downloaded.

```

# Read in data
osf_retrieve_file("esm4r") %>% osf_download(conflicts = "skip") # Download data from osf
data_loc <- "final_exploration_diagnostic_data.csv"

data <- read_csv(data_loc, na=c("NONE", "NA", ""))
## Clean up data columns

```

```

# Make selection name column human readable
data <- data %>% mutate(selection_name = as.factor(case_when(
  selection_name == "EpsilonLexicase" ~ "Lexicase",
  TOUR_SIZE == 1 ~ "Random",
  selection_name == "Tournament" ~ "Tournament",
  selection_name == "FitnessSharing" ~ "Fitness Sharing",
  selection_name == "EcoEA" ~ "EcoEA"
)))

# Calculate performance statistics.
# Elite trait avg is the avg per-site performance of the best individual
data$elite_trait_avg <-
  data$ele_agg_per / data$OBJECTIVE_CNT
data$unique_start_positions_coverage <-
  data$uni_str_pos / data$OBJECTIVE_CNT

# Convert elite_trait_avg to percent of maximum possible
data$elite_trait_avg <- data$elite_trait_avg/data$TARGET

# Grab data from just the final time point
final_data <- filter(data, evaluations==max(data$evaluations))

```

2.2 Performance

For context, it's important to know how each selection scheme performed on the exploration diagnostic.

2.2.1 Over time

First we look at the dynamics of performance over time.

2.2.1.1 Trait performance

Here, we plot average trait performance (i.e. fitness) over time for each selection scheme. We log the x-axis because Eco-EA gains fitness over a very long time scale, whereas the interesting dynamics for the other selection schemes occur relatively quickly.

```

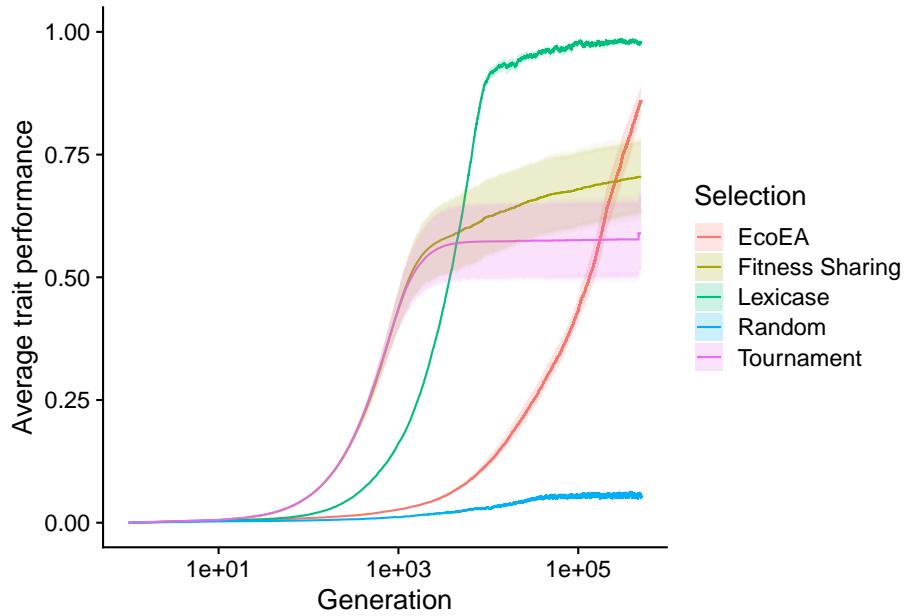
ggplot(
  data,
  aes(
    x=gen,                      # Generations
    y=elite_trait_avg,          # Performance
    color=selection_name,       # Selection scheme
    fill=selection_name
)

```

```

)
) +
stat_summary(geom="line", fun=mean) + # Plot line showing mean for each selection scheme
stat_summary( # Add shading around each line indicating 95% confidence interval
  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", # Set y axis title
  limits=c(0, performance_ylim) # Set y axis range to include all possible performance values
) +
scale_x_log10( # Log x axis
  name="Generation" # Set x axis title
) +
scale_color_discrete("Selection") + # Set legend title
scale_fill_discrete("Selection") # Set legend title

```



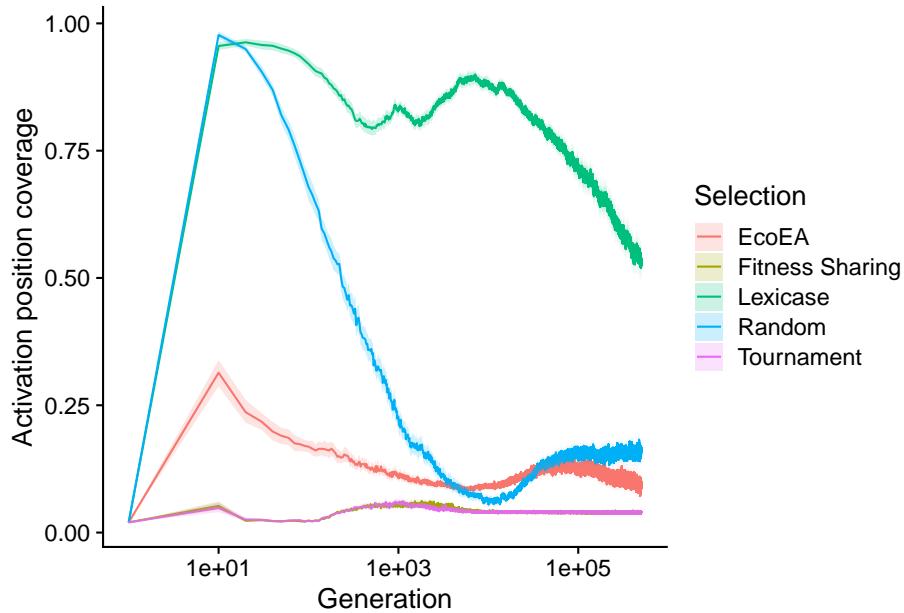
As observed by Hernandez et al. in their original paper on the exploration diagnostic (Hernandez et al., 2021), fitness in tournament selection initially increases quickly and then plateaus. Fitness in lexicase selection increases slightly slower but plateaus at a much higher value (nearly 100%). Fitness sharing behaves similarly to tournament selection, but maintains a slight upward

trajectory (note that, because the x axis is on a log scale, this trajectory is very gradual). Eco-EA takes substantially longer to increase in fitness but ultimately surpasses fitness sharing and tournament selection. It is unclear whether it would pass lexicase selection if these runs were allowed to continue for slightly longer; they do not appear to have plateaued yet. We chose to cut them off at 500,000 generations due to resource constraints and the fact that the questions we're asking here are not really about final fitness.

2.2.1.2 Activation position coverage

Out of curiosity, we also ran the analysis of unique activation positions present in the population, used by Hernandez et. al. This analysis tells us about the diversity of start positions for the coding region represented in the population. As the set of start positions in the population tends to represent a meaningful constraint on the number of paths through the fitness landscape that are currently accessible, this is in some sense a metric of useful diversity in the population

```
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"
  ) +
  scale_x_log10(
    name="Generation"
  ) +
  scale_color_discrete("Selection")+
  scale_fill_discrete("Selection")
```



We see that lexicase selection maintains by far that largest number of unique start positions, even surpassing the number maintained by random drift. This suggests that lexicase selection is actively selecting for maintaining a diversity of start positions. Tournament selection and fitness sharing perform virtually identically, with Eco-EA falling in between.

2.2.2 Final

While trends over time are more informative, it can be hard to visualize the full distribution (particularly the extent of variation). Thus, we also conduct a more detailed analysis of performance at the final time point.

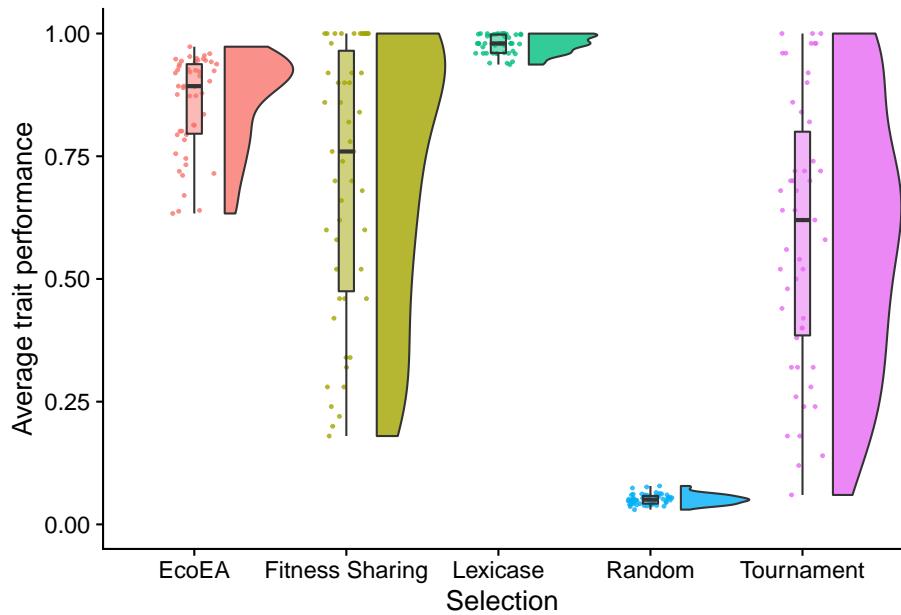
2.2.2.1 Trait performance

First we conduct statistics to identify which selection schemes are significantly different from each other.

```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(elite_trait_avg ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>% # Apply Bonferroni correction for multiple
  add_significance() %>%
  add_xy_position(x="selection_name", step.increase=1)
stat.test$label <- mapply(p_label, stat.test$p.adj)
```

Then we make raincloud plots (Allen et al., 2021) of each selection scheme.

```
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"
) +
  scale_fill_discrete(
    name="Selection"
) +
  scale_color_discrete(
    name="Selection"
) +
  theme(legend.position="none")
elite_final_performance_fig
```



These observations look fairly consistent with the timeseries plots.

Next, we output the results of our significance testing.

```
stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")
```

Fitness sharing did not perform significantly differently from Eco-EA or Tournament selection, but all other selection schemes are significantly different.

2.2.2.2 Final activation position Coverage

Now we do the same analysis for final activation position coverage.

First we calculate the statistics

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

.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj.signif	y
elite_trait_avg	EcoEA	Fitness Sharing	50	50	1561	3.20e-02	3.20e-01	ns	
elite_trait_avg	EcoEA	Lexicase	50	47	60	0.00e+00	0.00e+00	****	
elite_trait_avg	EcoEA	Random	50	50	2500	0.00e+00	0.00e+00	****	
elite_trait_avg	EcoEA	Tournament	50	50	1939	2.10e-06	2.07e-05	****	
elite_trait_avg	Fitness Sharing	Lexicase	50	47	593	2.69e-05	2.69e-04	***	
elite_trait_avg	Fitness Sharing	Random	50	50	2500	0.00e+00	0.00e+00	****	
elite_trait_avg	Fitness Sharing	Tournament	50	50	1549	4.00e-02	4.00e-01	ns	
elite_trait_avg	Lexicase	Random	47	50	2350	0.00e+00	0.00e+00	****	
elite_trait_avg	Lexicase	Tournament	47	50	2098	0.00e+00	0.00e+00	****	
elite_trait_avg	Random	Tournament	50	50	10	0.00e+00	0.00e+00	****	

```

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)

```

Then we make raincloud plots

```

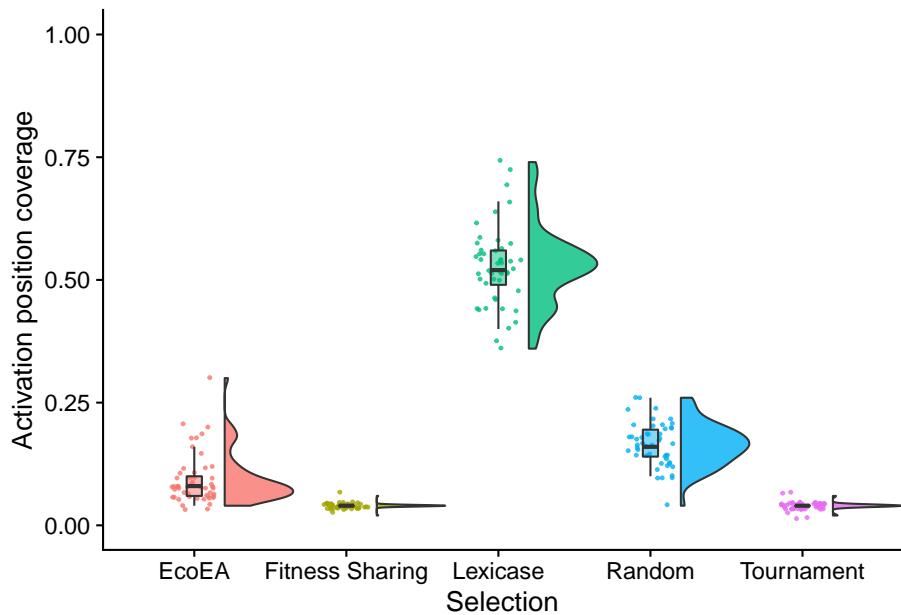
unique_start_positions_coverage_final_fig <- ggplot(
  final_data,
  aes(
    x=selection_name,
    y=unique_start_positions_coverage,
    fill=selection_name
  )
) +
  geom_flat_violin(
    position = position_nudge(x = .2, y = 0),
    alpha = .8,
    scale="width"
) +
  geom_point(
    mapping=aes(color=selection_name),
    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"
) +
scale_fill_discrete(
  name="Selection"
) +
scale_color_discrete(
  name="Selection"
) +
theme(
  legend.position="none"
)
unique_start_positions_coverage_final_fig

```



These also look unsurprising.

Lastly, we output the results of significance testing.

```

stat.test %>%
  kbl() %>%
  kable_styling(
  bootstrap_options = c(
    "striped",

```

.y.	group1	group2	n1	n2	statistic	p	p.adj	p.ad
unique_start_positions_coverage	EcoEA	Fitness Sharing	50	50	2392.5	0.000	0	****
unique_start_positions_coverage	EcoEA	Lexicase	50	47	0.0	0.000	0	****
unique_start_positions_coverage	EcoEA	Random	50	50	339.0	0.000	0	****
unique_start_positions_coverage	EcoEA	Tournament	50	50	2387.0	0.000	0	****
unique_start_positions_coverage	Fitness Sharing	Lexicase	50	47	0.0	0.000	0	****
unique_start_positions_coverage	Fitness Sharing	Random	50	50	25.0	0.000	0	****
unique_start_positions_coverage	Fitness Sharing	Tournament	50	50	1274.5	0.708	1	ns
unique_start_positions_coverage	Lexicase	Random	47	50	2350.0	0.000	0	****
unique_start_positions_coverage	Lexicase	Tournament	47	50	2350.0	0.000	0	****
unique_start_positions_coverage	Random	Tournament	50	50	2475.5	0.000	0	****

```

    "hover",
    "condensed",
    "responsive"
)
) %>%
scroll_box(width="600px")

```

2.3 Phylogenetic diversity

Next, we analyze the behavior of phylogenetic diversity on the exploration diagnostic.

2.3.1 Relationship between different types of phylogenetic diversity

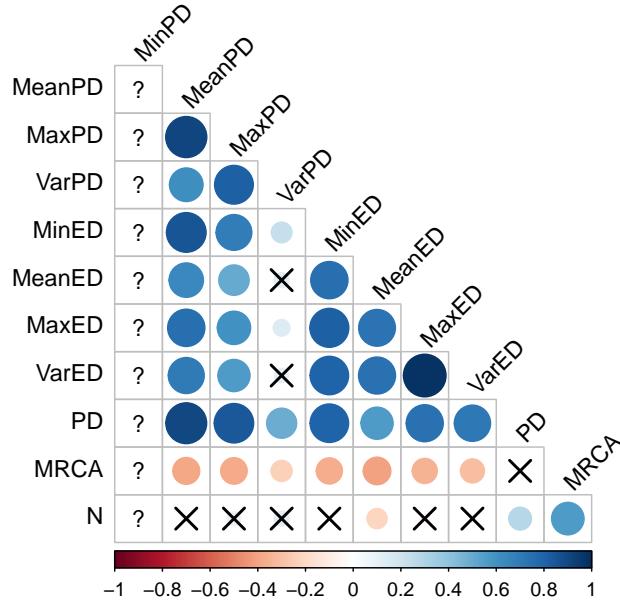
First, to get a big-picture overview, we make correlation matrices of all the different phylogenetic diversity metrics:

```

final_data %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
) %>%
  cor_mat() %>%

```

```
pull_lower_triangle() %>%
cor_plot()
```



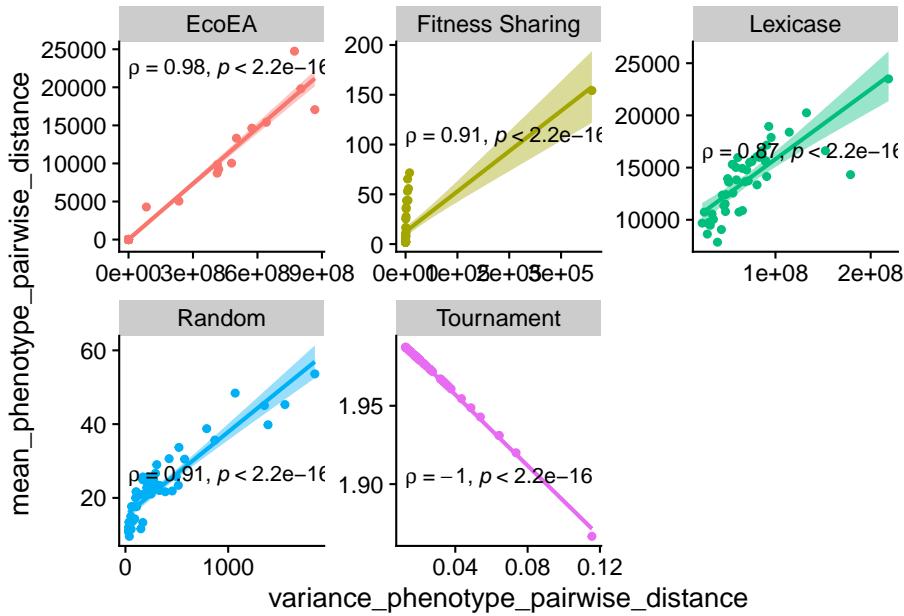
However, these correlations may well vary by selection scheme, and even over time within a selection scheme. Let's take a look at some scatter plots.

```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=variance_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
```

```

stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



Mean and variance of pairwise distances appears to correlate fairly closely with each other across selection schemes. The exception to this is tournament selection, where the range of values for both of these metrics are very small and the correlation is directly inverse. This is likely being driven by there being a small number of taxa that are mostly siblings of each other.

```

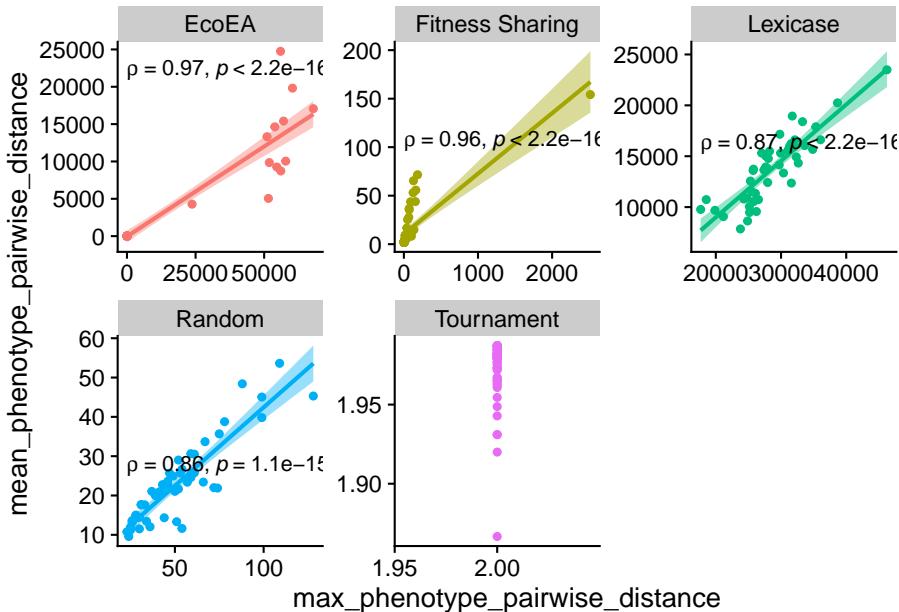
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
) +
  facet_wrap(

```

```

    ~selection_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



Maximum and mean pairwise distance also correlate pretty closely, with the exception again being tournament. These data shed further light on the previous graph as well - the maximum pairwise distance for tournament selection is always 2.

```

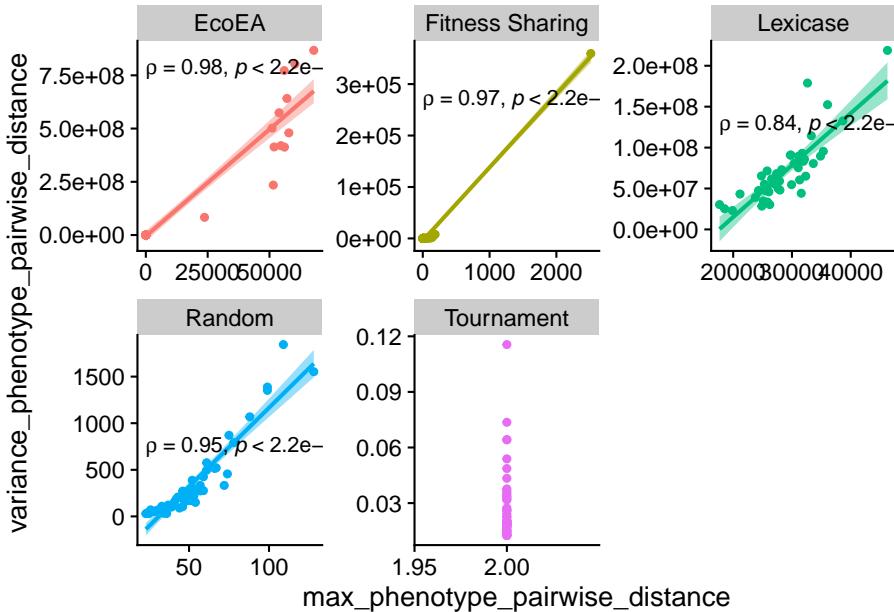
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=variance_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
geom_point() +

```

```

scale_x_continuous(
  breaks = breaks_extended(4)
) +
facet_wrap(~selection_name, scales="free")
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



Max and variance of pairwise distances behave similarly.

And let's check that it doesn't look radically different early on in the run:

```

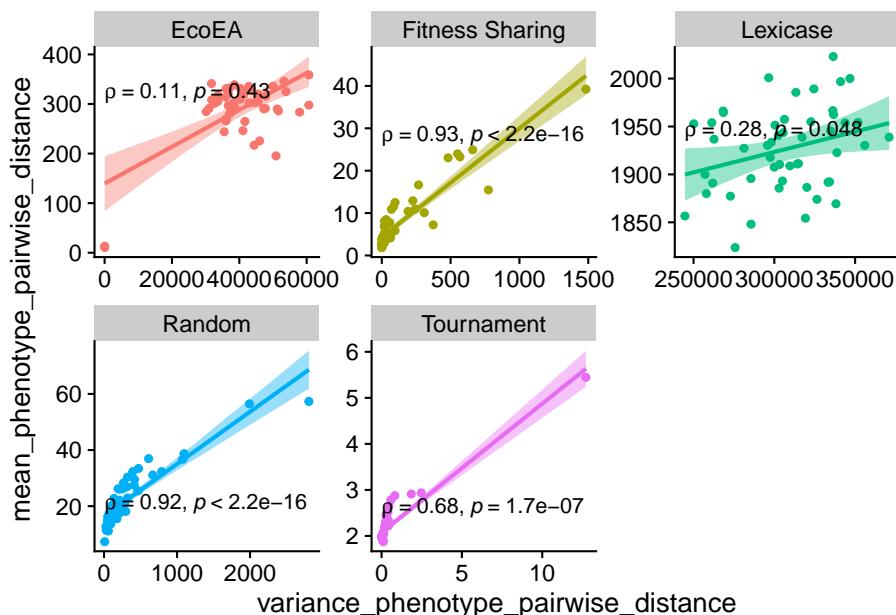
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=variance_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )

```

```

) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



The correlation between these two metrics is substantially weaker for Eco-EA and lexicase selection at the early time point.

```

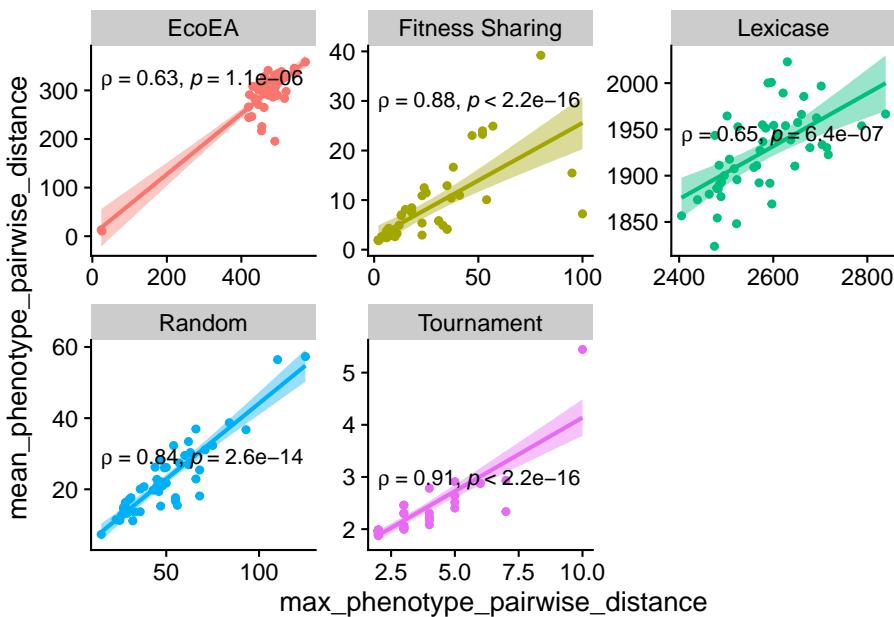
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,
    color=selection_name,

```

```

        fill=selection_name
    )
) +
geom_point() +
scale_x_continuous(
    breaks = breaks_extended(4)
) +
facet_wrap(
    ~selection_name, scales="free"
) +
stat_smooth(
    method="lm"
) +
stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



This relationship is also weaker, although not as much weaker.

```

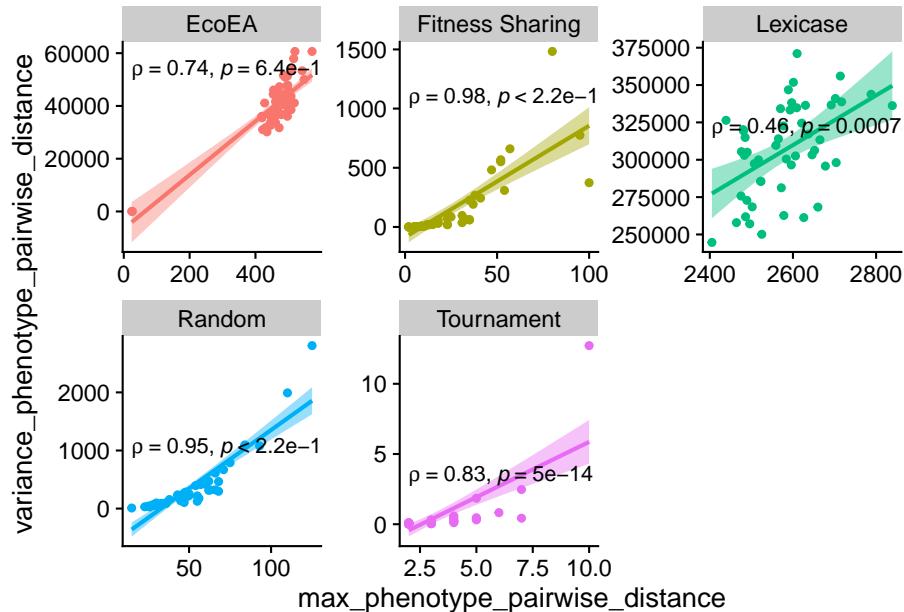
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=variance_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,

```

```

        color=selection_name,
        fill=selection_name
    )
) +
geom_point() +
scale_x_continuous(
    breaks = breaks_extended(4)
) +
facet_wrap(
    ~selection_name, scales="free"
) +
stat_smooth(
    method="lm"
) +
stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```

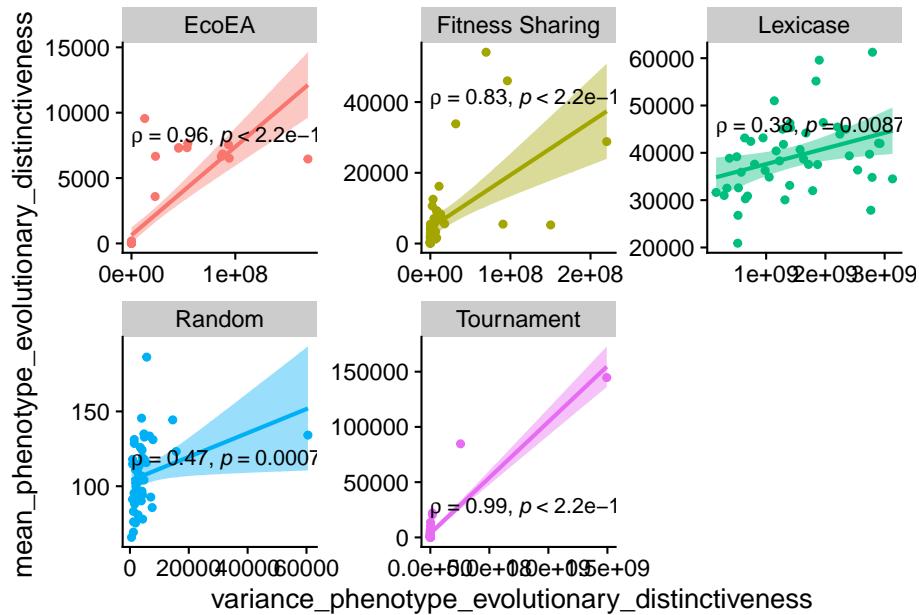


Similarly, the correlation between max and variance of pairwise distances is weakened, but still robust. So the relationship between mean and variance of pairwise distances is the only one that is really weaker at the earlier time points. Understanding exactly what drives these discrepancies is a promising angle for future work, as it may shed further light on the dynamics occurring in the run. It is particularly interesting that EcoEA and Lexicase selection are the scenarios

where they diverge.

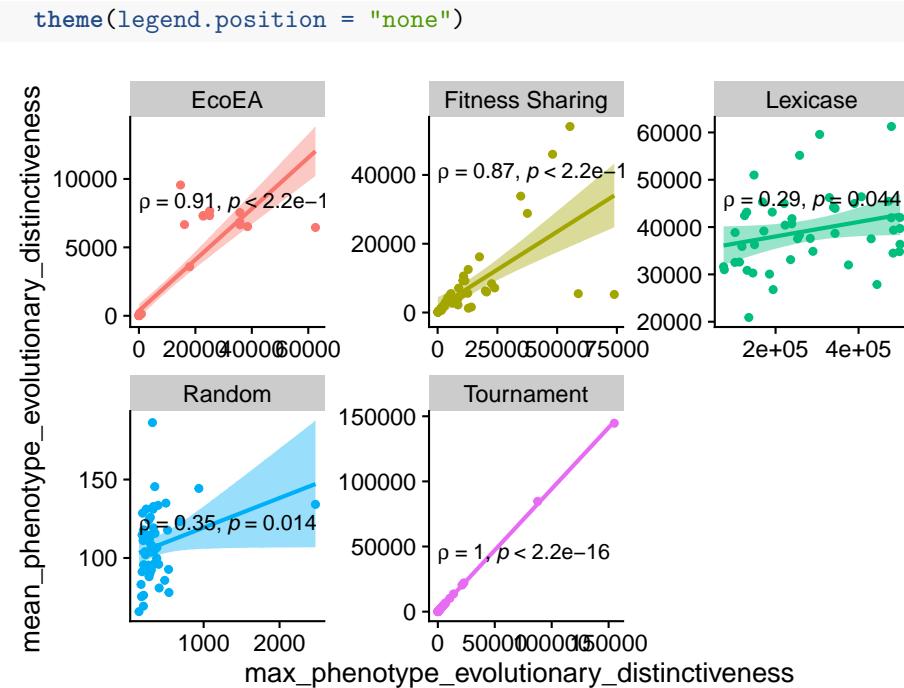
Similarly, the evolutionary distinctiveness metrics look largely similar to each other, but let's spot check that too.

```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=variance_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(
    ~selection_name, scales="free"
  ) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



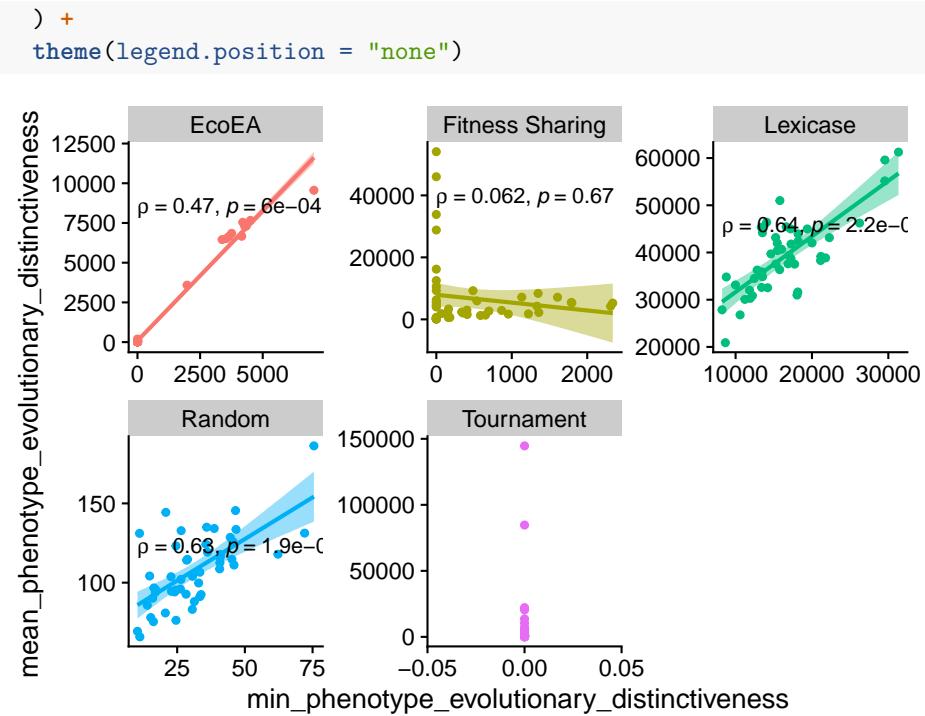
These aren't the strongest correlations; different information is definitely being captured by each metric. However, they still correlate to a fair extent.

```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=max_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
```



Again, for the most part there is a fairly high degree of correlation here.

```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=min_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
```

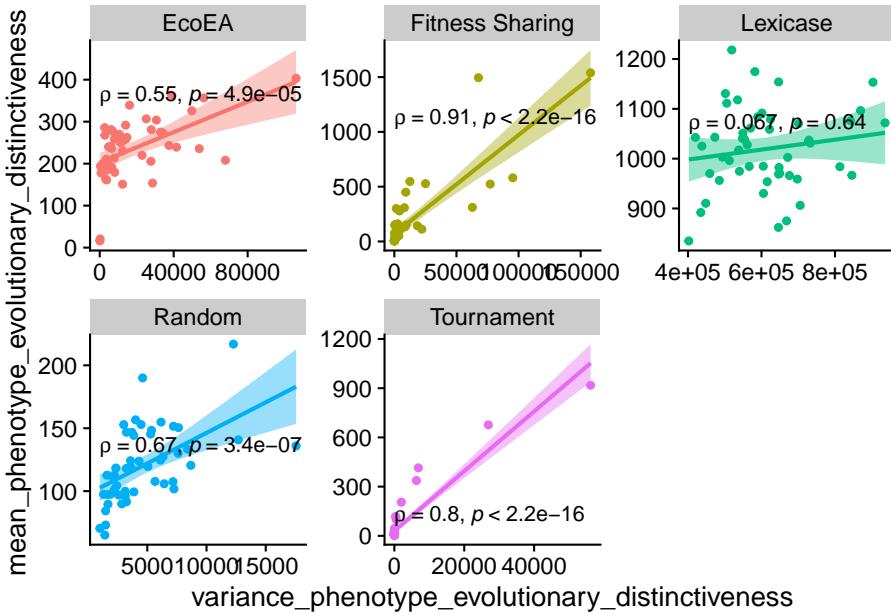


And here too, although it's getting kind of weak for lexicase.

And let's check out an earlier time point

```
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=variance_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
```

```
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



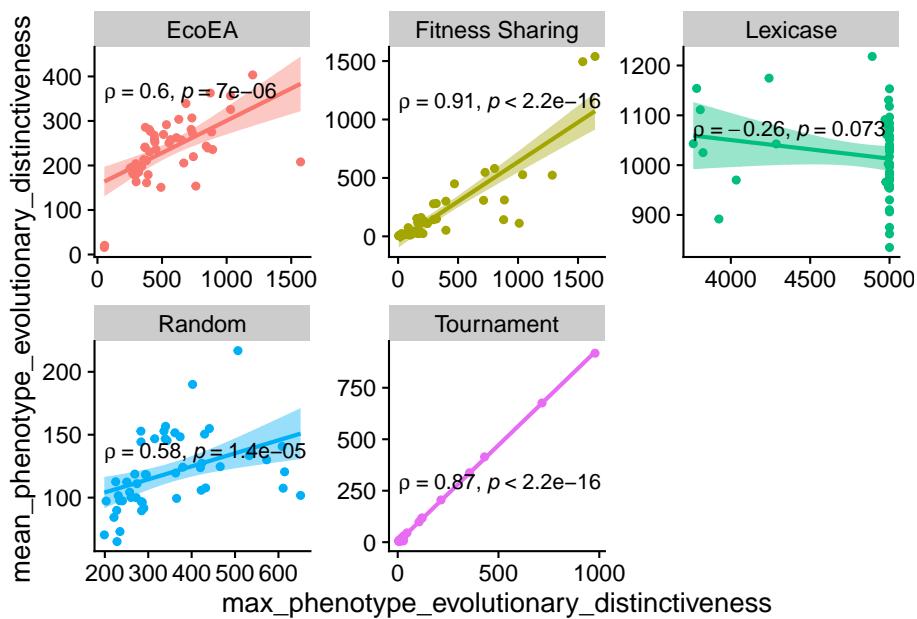
The relationship between min and mean evolutionary distinctiveness holds up at the earlier time point for everything except fitness sharing.

```
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=max_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth()
```

```

    method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



And here, for everything except lexicase

```

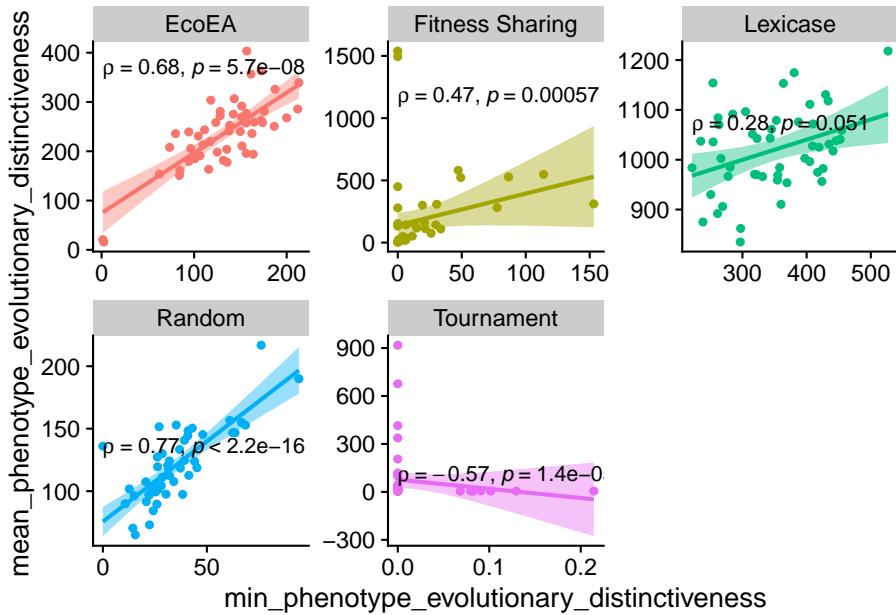
ggplot(
  data %>% filter(gen==5000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=min_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name, scales="free"
) +

```

```

stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



These are relatively weak for the most part.

Looks like the evolutionary distinctiveness statistics capture more different information from each other than the pairwise distance statistics (in general the correlations are weaker, and there are more deviations from the pattern of positive correlations). However, in a lot of cases they still correlate. There is definitely more exploration to be done on what differences in these metrics imply, but for simplicity we'll stick to one pairwise distance metric and one evolutionary distinctiveness metric.

Since it's hard to get a gestalt impression from the scatterplots, let's take a look at the correlation matrices for each selection scheme. Note that these are at the final time point, which clearly misses some dynamics that occur earlier.

2.3.1.1 Random

```

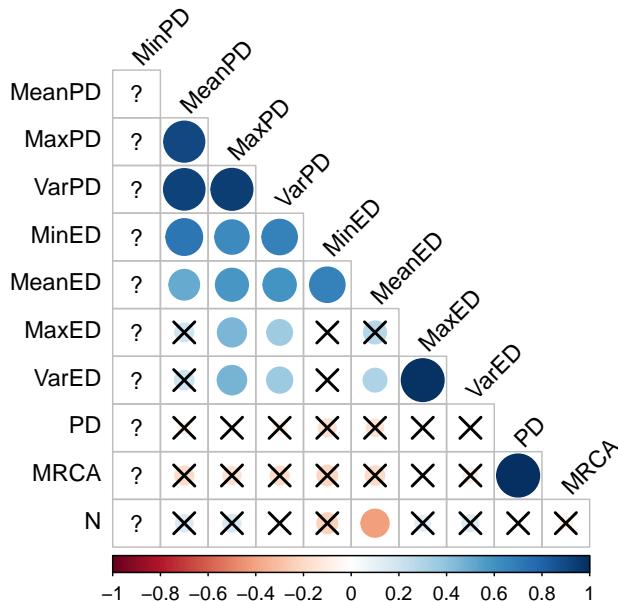
final_data %>%
  filter(selection_name=="Random") %>%

```

```

transmute(MinPD=min_phenotype_pairwise_distance,
          MeanPD=mean_phenotype_pairwise_distance,
          MaxPD=max_phenotype_pairwise_distance,
          VarPD=variance_phenotype_pairwise_distance,
          MinED = min_phenotype_evolutionary_distinctiveness,
          MeanED= mean_phenotype_evolutionary_distinctiveness,
          MaxED=max_phenotype_evolutionary_distinctiveness,
          VarED=variance_phenotype_evolutionary_distinctiveness,
          PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
          MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
          N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
) %>%
cor_mat() %>%
pull_lower_triangle() %>%
cor_plot()

```

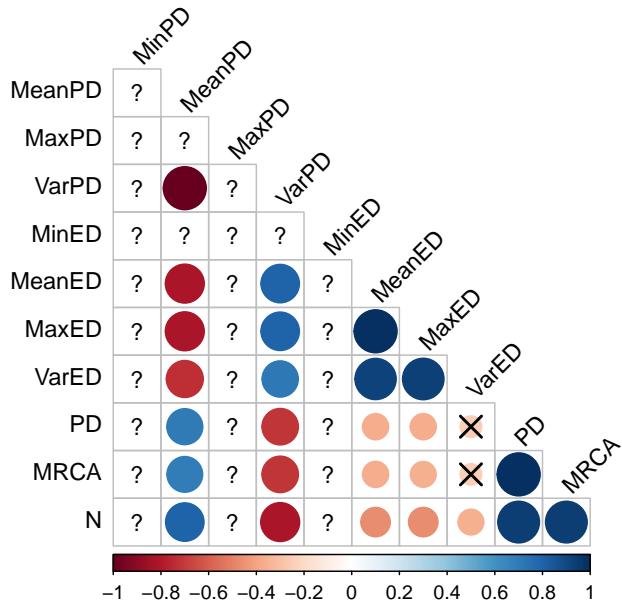


In random selection, all the pairwise distance metrics correlate with each other, and also more weakly with some of the evolutionary distinctiveness metrics, but not max and variance (although they correlate strongly with each other). It is somewhat surprising that the depth of the most recent common ancestor and Faith's Phylogenetic Diversity metric have so little correlation with the others.

```

final_data %>%
  filter(selection_name=="Tournament") %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
  ) %>%
  cor_mat() %>%
  pull_lower_triangle() %>%
  cor_plot()

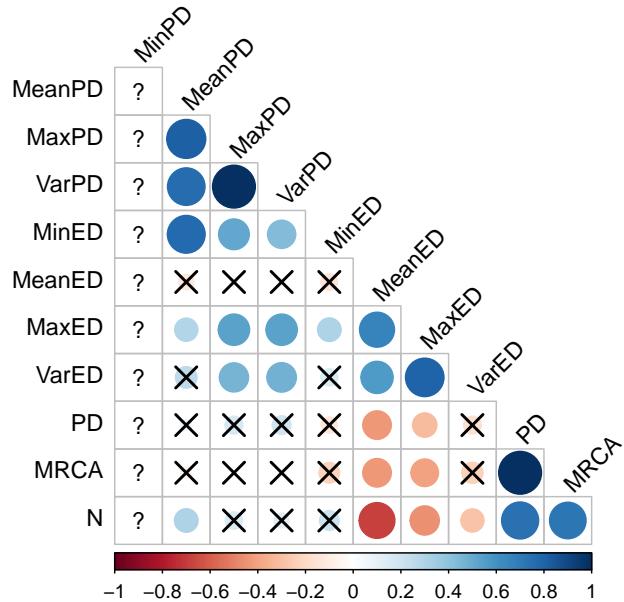
```



Here, all the evolutionary distinctiveness metrics corelate strongly with each other, but the pairwise distance metrics less so (although this is likely an artifact of the extremely low phylogenetic diversity in tournament selection)

2 3 1 3 Fitness Sharing

```
final_data %>%
  filter(selection_name=="Fitness Sharing") %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
  ) %>%
  cor_mat() %>%
  pull_lower_triangle() %>%
  cor_plot()
```

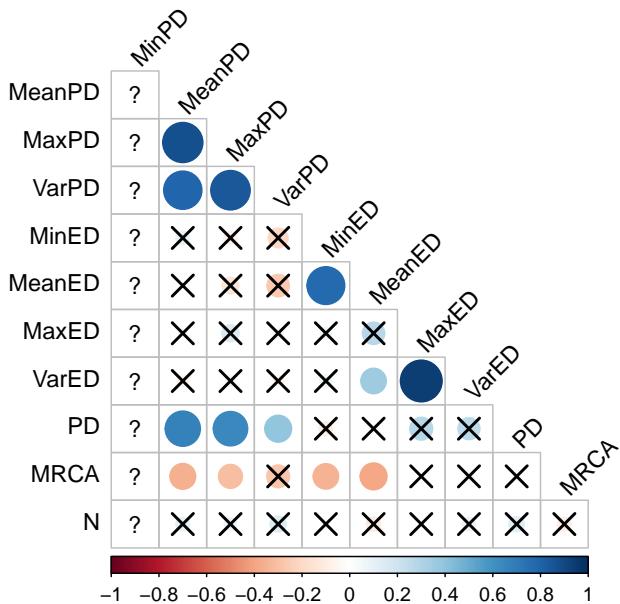


Here, the pairwise distance metrics correlate with each other and the evolutionary distinctiveness metrics correlated with each other. There are some correlations between pairwise distance and evolutionary distinctiveness metrics, but they're weaker.

```

final_data %>%
  filter(selection_name=="Lexicase") %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
  ) %>%
  cor_mat() %>%
  pull_lower_triangle() %>%
  cor_plot()

```

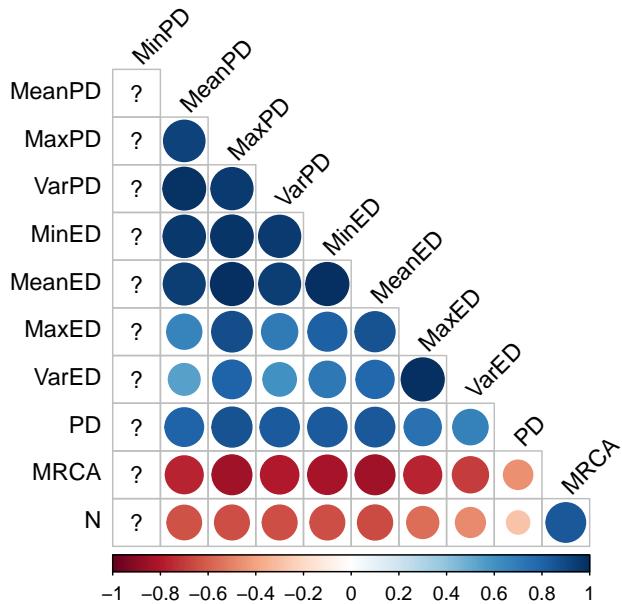


Again, the pairwise distance metrics correlate with each other but not wth the volutionary distinctivness metrics. The evolutionary distinctiveness metrics sort of correlate with each other.

```

2 3 1 5  Eco-EA
final_data %>%
  filter(selection_name=="EcoEA") %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phen_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phen_num_taxa      # Number of taxonomically-distinct phenotypes
) %>%
cor_mat() %>%
pull_lower_triangle() %>%
cor_plot()

```



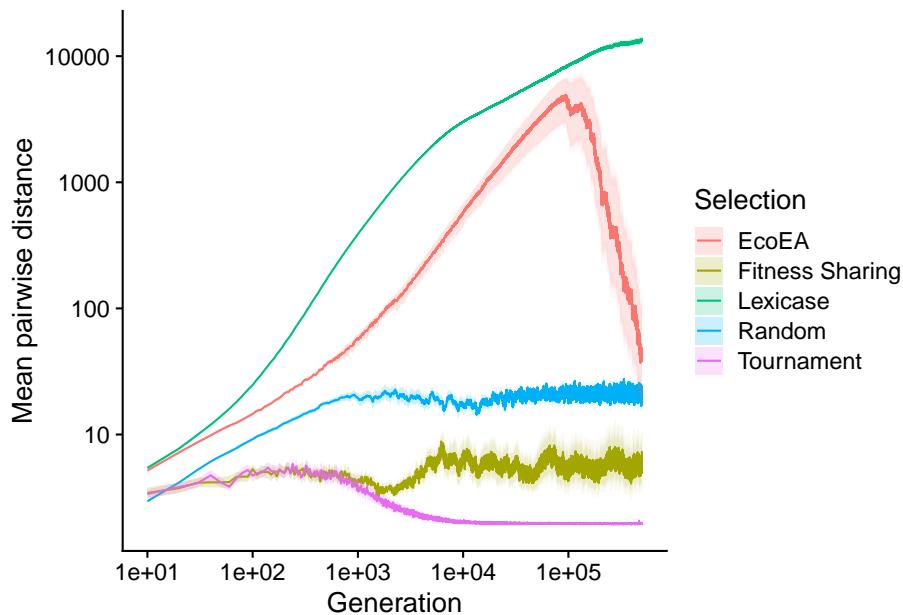
In Eco-EA, there is an impressively strong positive correlation among all the phylodiversity metrics. Understanding what causes Eco-EA to deviate from lexicase selection (and to a lesser extent fitness sharing) in this way would be worthy of further research.

2.3.2 Over time

2.3.2.1 Mean pairwise distance

First we plot mean pairwise distance over time. We log the y axis because there is such variation in mean pairwise distance across selection schemes, and the x-axis for the same reason as before.

```
ggplot(
  data,
  aes(
    x=gen,
    y=mean_phenotype_pairwise_distance,
    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_log10(
    name="Mean pairwise distance"
) +
  scale_x_log10(
    name="Generation"
) +
  scale_color_discrete("Selection") +
  scale_fill_discrete("Selection")
```



Lexicase selection maintains a monotonic increase in phylogenetic diversity over the course of the entire experiment. It likely never experiences a full coalescence event (where the most-recent common ancestor changes). Eco-EA nearly keeps pace with lexicase selection until towards the end, when its phylogenetic diversity crashes. This is likely the result of selective sweeps that begin to occur as the population discovers high fitness solutions. Fitness sharing shows a slight dip at the same time that its fitness plateaus (likely also the result of a selective sweep), but phylogenetic diversity recovers afterwards, making for a relatively constant level. over time. Tournament selection, on the other hand, maintains the same (low) level of phylogenetic diversity as fitness sharing, up until the point that fitness plateaus, at which point tournament selection's phylodiversity drops to nearly 0. Interestingly, lexicase selection and Eco-EA both maintain more phylodiversity than random selection, whereas fitness sharing and tournament selection maintain less.

2.3.2.2 Mean evolutionary distinctiveness

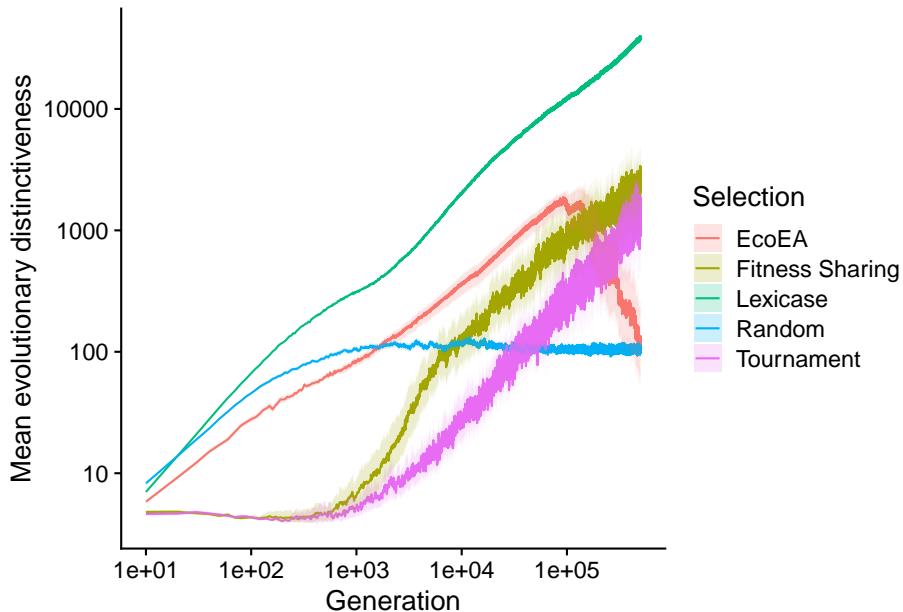
For comparison, we make the same plot with mean evolutionary distinctiveness.

```
ggplot(
  data,
  aes(
    x=gen,
    y=mean_phenotype_evolutionary_distinctiveness,
    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_log10(
  name="Mean evolutionary distinctiveness"
) +
scale_x_log10(
  name="Generation"
) +
scale_color_discrete("Selection") +
scale_fill_discrete("Selection")

```



Interestingly, Fitness Sharing and tournament selection both start increasing in evolutionary distinctiveness only after their fitnesses have plateaued. This seems likely to be due to some sort of pathological behavior of mean evolutionary distinctiveness on small trees, but more investigation would be necessary to figure out exactly what's going on. Trends in other selection schemes are largely similar.

.y.	group1	group2	n1	n2	statistic	
mean_phenotype_pairwise_distance	EcoEA	Fitness Sharing	50	50	1824.0	7.70
mean_phenotype_pairwise_distance	EcoEA	Lexicase	50	47	227.0	0.00
mean_phenotype_pairwise_distance	EcoEA	Random	50	50	690.0	1.19
mean_phenotype_pairwise_distance	EcoEA	Tournament	50	50	2500.0	0.00
mean_phenotype_pairwise_distance	Fitness Sharing	Lexicase	50	47	0.0	0.00
mean_phenotype_pairwise_distance	Fitness Sharing	Random	50	50	536.0	9.00
mean_phenotype_pairwise_distance	Fitness Sharing	Tournament	50	50	2232.5	0.00
mean_phenotype_pairwise_distance	Lexicase	Random	47	50	2350.0	0.00
mean_phenotype_pairwise_distance	Lexicase	Tournament	47	50	2350.0	0.00
mean_phenotype_pairwise_distance	Random	Tournament	50	50	2500.0	0.00

2.3.3 Final

Next, we perform a more in-depth analysis of phylogenetic diversity distributions at the final time point.

2.3.3.1 Mean pairwise distance

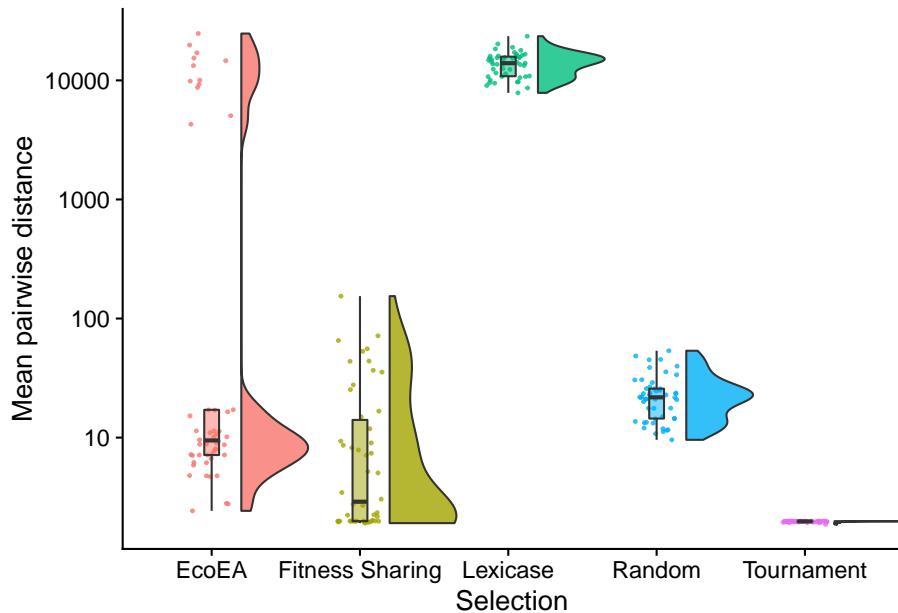
First, we test which selection schemes end up with significantly different final distributions of mean pairwise distance.

```
# Pairwise wilcoxon test set to determine which conditions are significantly different from each other
stat.test <- final_data %>%
  wilcox_test(mean_phenotype_pairwise_distance ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name", step.increase=1)
stat.test$label <- mapply(p_label, stat.test$p.adj)

# Output stats
stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")
```

Looks like they are all significantly different from each other.

```
# Raincloud plot of final mean pairwise distance
final_phylogeny_fig <- ggplot(
  final_data,
  aes(
    x=selection_name,
    y=mean_phenotype_pairwise_distance,
    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_log10(
    name="Mean pairwise distance"
) +
  scale_x_discrete(
    name="Selection"
) +
  scale_fill_discrete(
    name="Selection"
) +
  scale_color_discrete(
    name="Selection"
) +
  theme(legend.position = "none")
final_phylogeny_fig
```



This shows something interesting! Final phylogenetic diversity in Eco-EA is heavily bimodal. In later analysis, we will see that the runs with high phylogenetic diversity are the ones with lower fitness, suggesting that they have no yet experienced a selective sweep resulting from the discovery of a high-fitness solution.

2.3.3.2 Mean evolutionary distinctiveness

Tests for significant differences:

```
# Pairwise wilcoxon test to determine which conditions are significantly different fr
stat.test <- final_data %>%
  wilcox_test(mean_phenotype_evolutionary_distinctiveness ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name", step.increase=1)
stat.test$label <- mapply(p_label, stat.test$p.adj)

# Output stats

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
```

.y.	group1	group2	n1	n2	statistic	
mean_phenotype_evolutionary_distinctiveness	EcoEA	Fitness Sharing	50	50	469	1.00e+
mean_phenotype_evolutionary_distinctiveness	EcoEA	Lexicase	50	47	0	0.00e+
mean_phenotype_evolutionary_distinctiveness	EcoEA	Random	50	50	711	2.05e+
mean_phenotype_evolutionary_distinctiveness	EcoEA	Tournament	50	50	569	2.70e+
mean_phenotype_evolutionary_distinctiveness	Fitness Sharing	Lexicase	50	47	100	0.00e+
mean_phenotype_evolutionary_distinctiveness	Fitness Sharing	Random	50	50	2428	0.00e+
mean_phenotype_evolutionary_distinctiveness	Fitness Sharing	Tournament	50	50	1614	1.20e+
mean_phenotype_evolutionary_distinctiveness	Lexicase	Random	47	50	2350	0.00e+
mean_phenotype_evolutionary_distinctiveness	Lexicase	Tournament	47	50	2255	0.00e+
mean_phenotype_evolutionary_distinctiveness	Random	Tournament	50	50	173	0.00e+

```

    "condensed",
    "responsive"
)
) %>%
scroll_box(width="600px")

```

Looks like everything except fitness sharing and tournament are significantly different from each other.

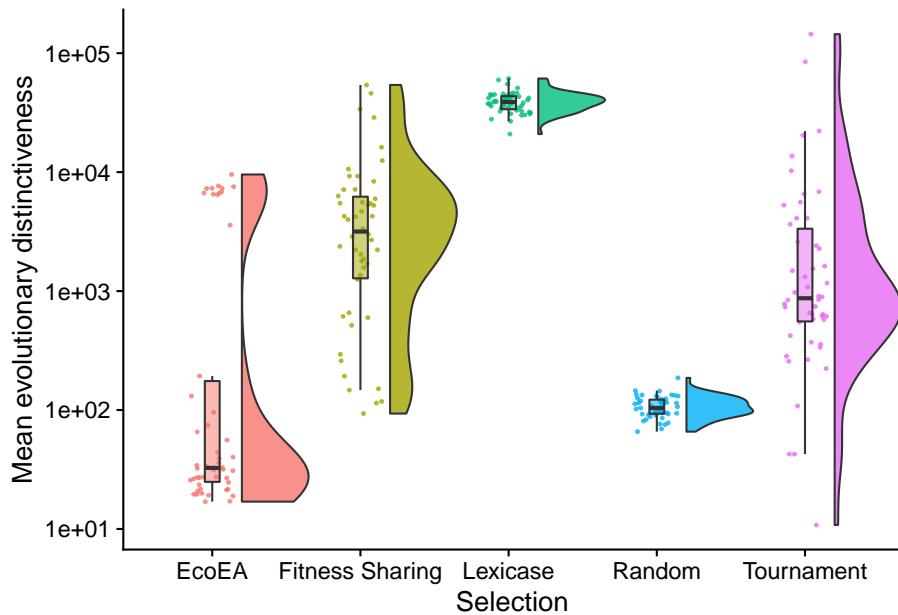
```

# Raincloud plot of final mean evolutionary distinctiveness
ggplot(
  final_data,
  aes(
    x=selection_name,
    y=mean_phenotype_evolutionary_distinctiveness,
    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_log10(
  name="Mean evolutionary distinctiveness"
) +
scale_x_discrete(
  name="Selection"
) +
scale_fill_discrete(
  name="Selection"
) +
scale_color_discrete(
  name="Selection"
) +
theme(legend.position = "none")

```



Again, this looks fairly similar to MPD, except that fitness sharing and tournament are higher.

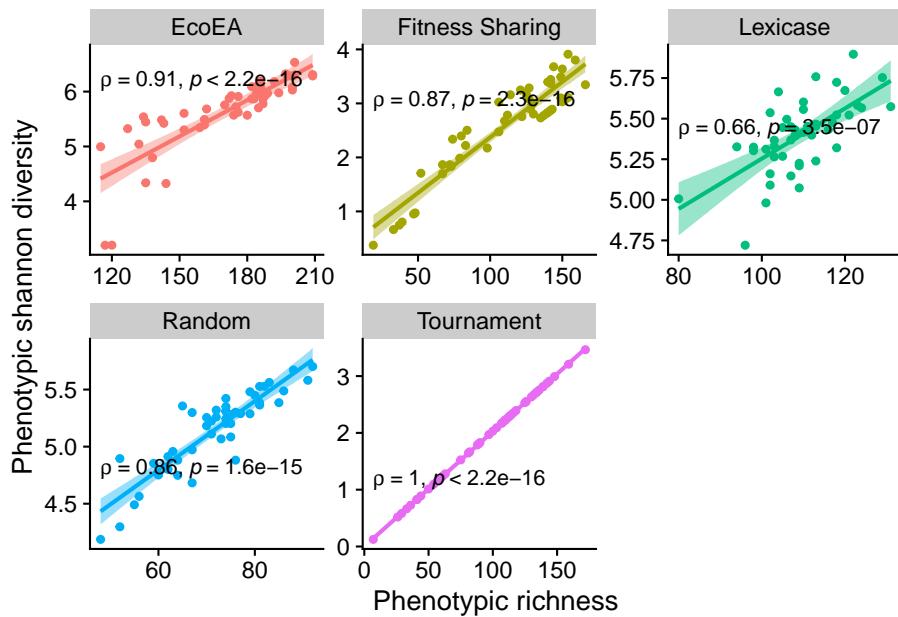
2.4 Phenotypic diversity

Now we analyze phenotypic (i.e. population-level) diversity. Here, we're defining phenotypes to only include the sites that are actively contributing to fitness. So the phenotype of [1,4,2,6,5,4,3,6,7] would be [0,0,0,6,5,4,3,0,0]. Note that phylogenetic trees are also built using this conception of phenotypes.

2.4.1 Relationship between different types of phenotypic diversity

First, we should assess the extent to which different metrics of phenotypic diversity are capturing different information.

```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=phen_diversity,
    x=phen_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Phenotypic shannon diversity"
) +
  scale_x_continuous(
    name="Phenotypic richness",
    breaks = breaks_extended(4)
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



Looks like at the final time point they are pretty much always closely correlated, although this correlation is weaker for lexicase selection than for other selection schemes.

2.4.2 Over time

Now we examine the behavior of each phenotypic diversity metric over time.

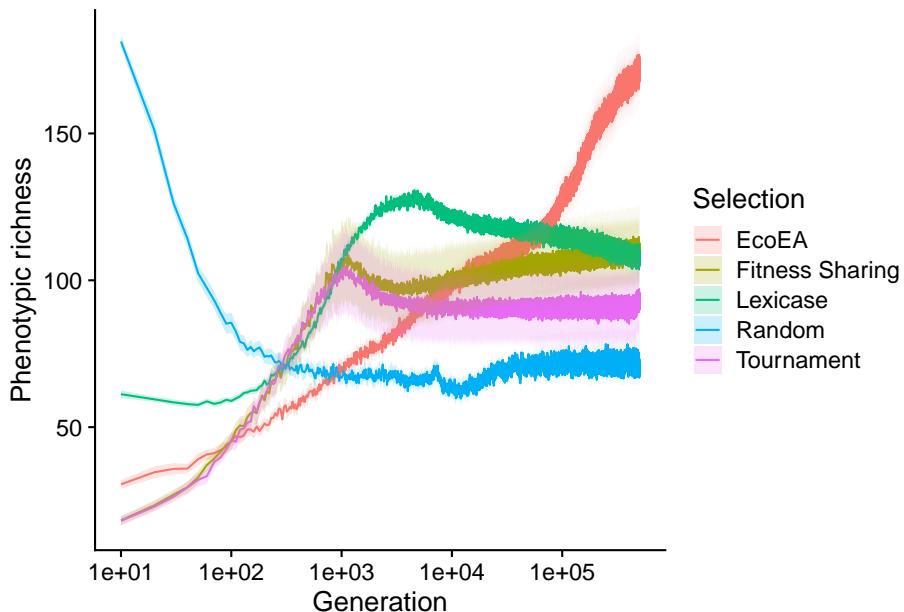
2.4.2.1 Richness

```
ggplot(
  data,
  aes(
    x=gen,
    y=phen_num_taxa,
    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="Phenotypic richness"
) +
scale_x_log10(
  name="Generation"
) +
scale_color_discrete("Selection") +
scale_fill_discrete("Selection")

```



In contrast to the phylodiversity results, phenotypic richness in all selection schemes (even tournament selection) ultimately exceeds that of random selection. Eco-EA monotonically increases while lexicase selection reaches a maximum around the same time it reaches its fitness plateau. The only real similarity to the phylodiversity results is the behavior tournament selection and fitness sharing relative to each other.

2.4.2.2 Shannon diversity

We also looked at phenotypic shannon diversity:

```

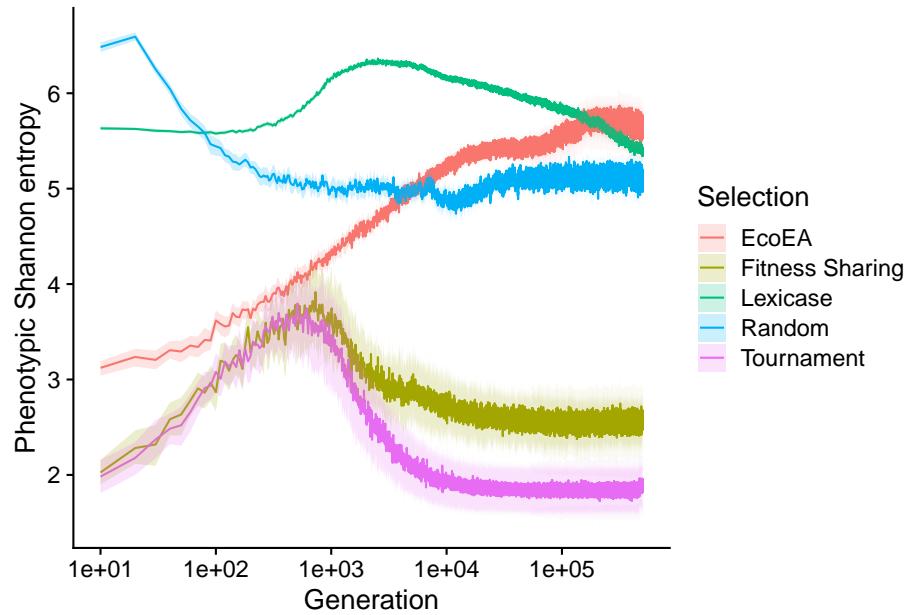
ggplot(
  data,
  aes(
    x=gen,

```

```

y=phen_diversity,
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="Phenotypic Shannon entropy"
) +
scale_x_log10(
  name="Generation"
) +
scale_color_discrete("Selection") +
scale_fill_discrete("Selection")

```



This is more different from the richness results than we might have expected based on the correlation at the final time point. There is a much more pronounced drop off in Shannon entropy for tournament and fitness sharing than in richness. This

difference is probably driven by the fact that, after plateauing, these selection schemes are likely both at mutation-selection balance. Thus, there are very many single-mutant phenotypes with only one copy in the population. These do not contribute much to Shannon entropy or phylogenetic diversity, but it does show up in richness.

We also see that, after plateauing, lexicase selection does actually start to decrease in Shannon entropy, but slowly. Similarly, in Eco-EA, the increase in Shannon entropy towards the end is much more modest than the increase in richness.

2.4.3 Final

Now we assess the final phenotypic diversity

2.4.3.1 Richness

Hypothesis-testing differences between groups:

```
# Determine which conditions are significantly different from each other
stat.test <- final_data %>%
  wilcox_test(phen_num_taxa ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name",step.increase=1)
stat.test$label <- mapply(p_label,stat.test$p.adj)

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")
```

Raincloud plot:

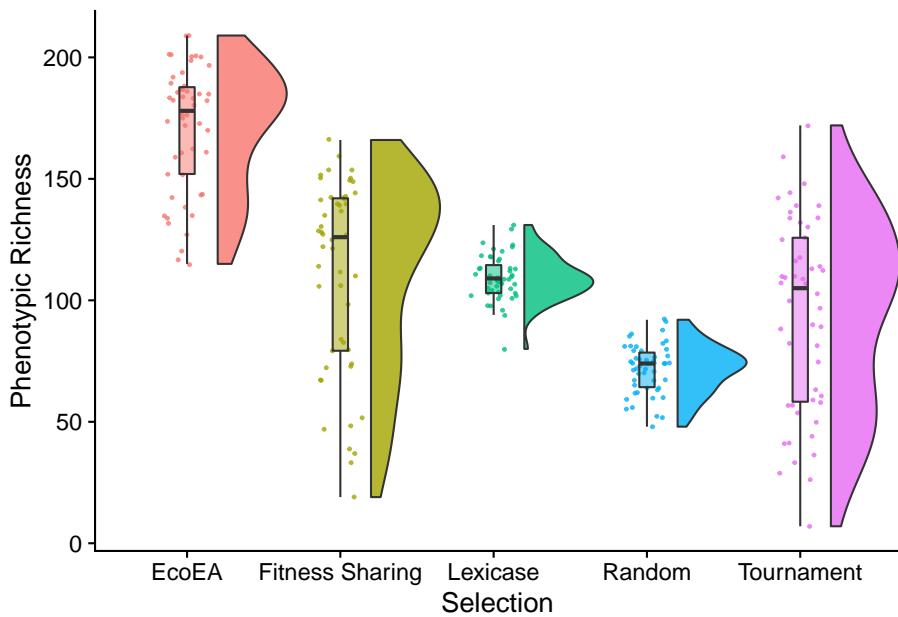
```
# Raincloud plot of final phenotypic diversity
final_phenotypic_fig <- ggplot(
  final_data,
  aes(
    x=selection_name,
    y=phen_num_taxa,
    fill=selection_name
```

.y.	group1	group2	n1	n2	statistic	p	p.adj	p.
phen_num_taxa	EcoEA	Fitness Sharing	50	50	2249.0	0.00e+00	0.0e+00	**
phen_num_taxa	EcoEA	Lexicase	50	47	2319.0	0.00e+00	0.0e+00	**
phen_num_taxa	EcoEA	Random	50	50	2500.0	0.00e+00	0.0e+00	**
phen_num_taxa	EcoEA	Tournament	50	50	2378.5	0.00e+00	0.0e+00	**
phen_num_taxa	Fitness Sharing	Lexicase	50	47	1428.0	6.80e-02	6.8e-01	ns
phen_num_taxa	Fitness Sharing	Random	50	50	1973.0	6.00e-07	6.3e-06	**
phen_num_taxa	Fitness Sharing	Tournament	50	50	1585.0	2.10e-02	2.1e-01	ns
phen_num_taxa	Lexicase	Random	47	50	2339.5	0.00e+00	0.0e+00	**
phen_num_taxa	Lexicase	Tournament	47	50	1359.0	1.85e-01	1.0e+00	ns
phen_num_taxa	Random	Tournament	50	50	797.0	2.00e-03	2.0e-02	*

```

)
) +
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="Phenotypic Richness"
) +
scale_x_discrete(
  name="Selection"
) +
scale_fill_discrete(
  name="Selection"
) +
scale_color_discrete(
  name="Selection"
) +
theme(legend.position = "none")
final_phenotypic_fig

```



Nothing particularly surprising here, but we should note that, based on the over time plot, this would look a lot different if we had selected a different time point.

2.4.3.2 Shannon diversity

```
# Determine which conditions are significantly different from each other
stat.test <- final_data %>%
  wilcox_test(phen_diversity ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x = "selection_name", step.increase=1)
stat.test$label <- mapply(p_label, stat.test$p.adj)

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")
```

.y.	group1	group2	n1	n2	statistic	p	p.adj	p.a
phen_diversity	EcoEA	Fitness Sharing	50	50	2478.0	0.00e+00	0.00e+00	***
phen_diversity	EcoEA	Lexicase	50	47	1772.0	1.66e-05	1.66e-04	**
phen_diversity	EcoEA	Random	50	50	2089.5	0.00e+00	1.00e-07	***
phen_diversity	EcoEA	Tournament	50	50	2496.0	0.00e+00	0.00e+00	***
phen_diversity	Fitness Sharing	Lexicase	50	47	0.0	0.00e+00	0.00e+00	***
phen_diversity	Fitness Sharing	Random	50	50	0.0	0.00e+00	0.00e+00	***
phen_diversity	Fitness Sharing	Tournament	50	50	1856.0	2.99e-05	2.99e-04	***
phen_diversity	Lexicase	Random	47	50	1714.0	1.01e-04	1.01e-03	**
phen_diversity	Lexicase	Tournament	47	50	2350.0	0.00e+00	0.00e+00	***
phen_diversity	Random	Tournament	50	50	2500.0	0.00e+00	0.00e+00	***

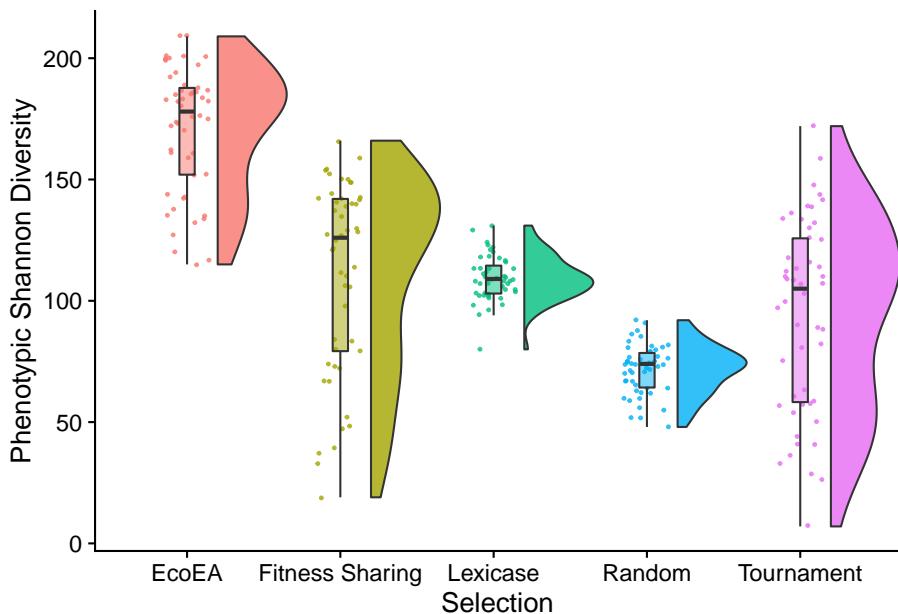
Interestingly, the final shannon diversity values of different selection schemes are much more distinguishable from each other than the final richness values.

```
# Raincloud plot of final phenotypic diversity
ggplot(
  final_data,
  aes(
    x=selection_name,
    y=phen_num_taxa,
    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="Phenotypic Shannon Diversity"
  ) +
  scale_x_discrete(
    name="Selection"
```

```

) +
  scale_fill_discrete(
    name="Selection"
) +
  scale_color_discrete(
    name="Selection"
) +
  theme(legend.position = "none")

```



Again, we know from the time series plots that these relative relationships varied a lot over time. Eco-EA is only higher than lexicase selection at the very end.

2.5 Relationship between phenotypic and phylogenetic diversity

Now, we can finally begin to address the main questions. Do phenotypic diversity and phylogenetic diversity capture different information?

2.5.1 Phenotypic richness vs. mean pairwise distance

```

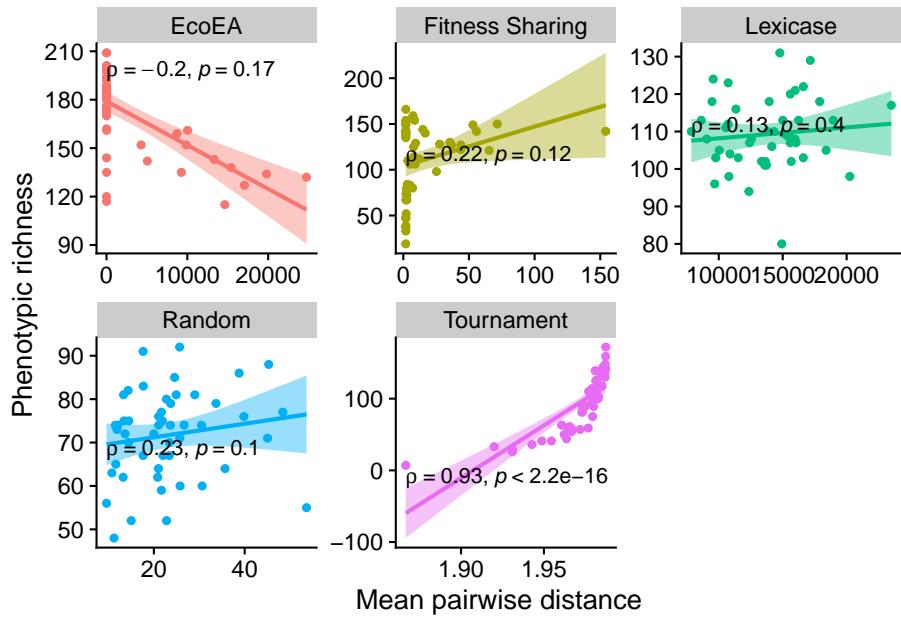
ggplot(
  data %>% filter(gen==500000),
  aes(

```

```

y=phen_num_taxa,
x=mean_phenotype_pairwise_distance,
color=selection_name,
fill=selection_name
)
)
) +
geom_point() +
scale_y_continuous(
  name="Phenotypic richness"
) +
scale_x_continuous(
  name="Mean pairwise distance",
  breaks = breaks_extended(4)
) +
facet_wrap(
  ~selection_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

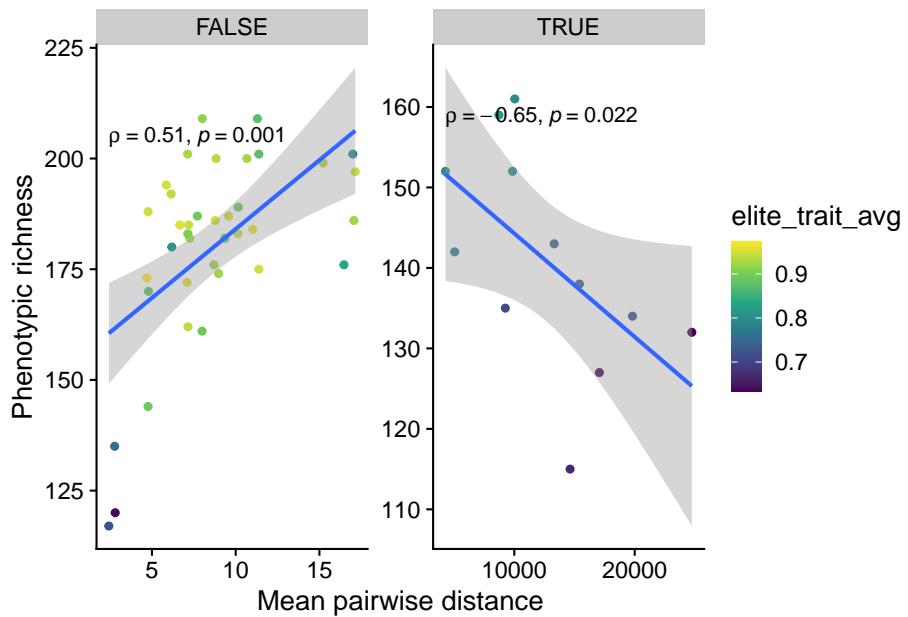
```



The linear models for some of these are questionable, but the Spearman correlation coefficient should be fine because it does not require linearity. Only tournament selection is significantly different from 0. Eco-EA is even negative (although non-significant), which is probably driven by the fact that there are really two groups of runs in Eco-EA: those that have found a good solution and had their diversity crash, and those that haven't yet.

Lets take a look at how each of those groups behave (we'll show fitness as color, just to better understand what's happening):

```
ggplot(
  data %>% filter(gen==500000, selection_name == "EcoEA"),
  aes(
    y=phen_num_taxa,
    x=mean_phenotype_pairwise_distance,
    color = elite_trait_avg
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Phenotypic richness"
) +
  scale_x_continuous(
    name="Mean pairwise distance",
    breaks = breaks_extended(4)
) +
  facet_wrap(
    ~mean_phenotype_pairwise_distance > 100, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  scale_color_continuous(type="viridis")
```



2.5.2 Phenotypic richness vs. mean evolutionary distinctiveness

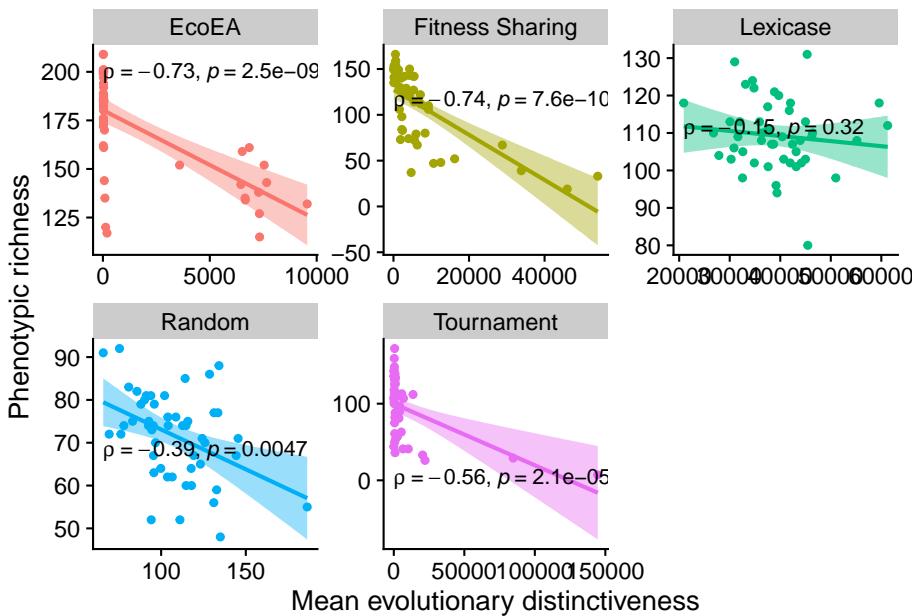
```
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=phen_num_taxa,
    x=mean_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Phenotypic richness"
  ) +
  scale_x_continuous(
    name="Mean evolutionary distinctiveness",
    breaks = breaks_extended(4)
  ) +
  facet_wrap(
    ~selection_name, scales="free"
  ) +
  stat_smooth()
```

2.5. RELATIONSHIP BETWEEN PHENOTYPIC AND PHYLOGENETIC DIVERSITY61

```

    method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



These are mostly significant, but negative. That's interesting and worthy of further exploration (but it's a little beyond the scope of this analysis).

2.5.3 Phenotypic shannon diversity vs. mean pairwise distance

```

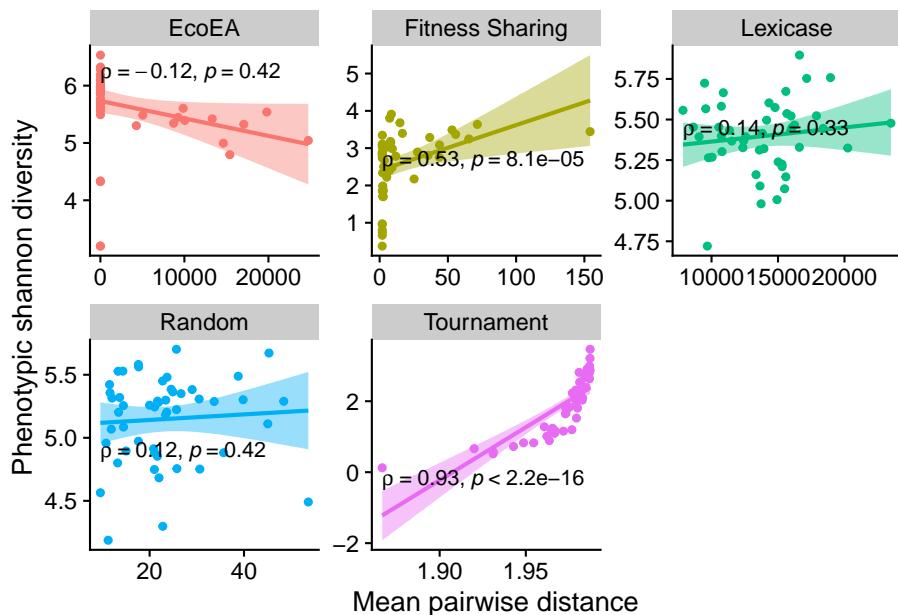
ggplot(
  data %>% filter(gen==500000),
  aes(
    y=phen_diversity,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(

```

```

        name="Phenotypic shannon diversity"
) +
scale_x_continuous(
  name="Mean pairwise distance",
  breaks = breaks_extended(4)
) +
facet_wrap(~selection_name, scales="free")
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



Again, these are mostly not significant (although fitness sharing notably now is).

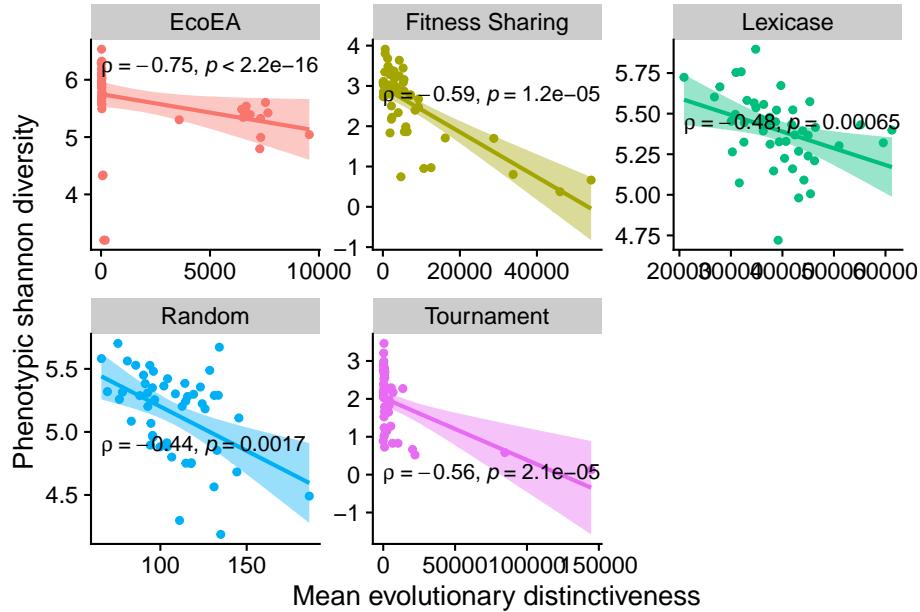
2.5.4 Phenotypic shannon diversity vs. mean evolutionary distinctiveness

```

ggplot(
  data %>% filter(gen==500000),

```

```
aes(  
    y=phen_diversity,  
    x=mean_phenotype_evolutionary_distinctiveness,  
    color=selection_name,  
    fill=selection_name  
)  
) +  
geom_point() +  
scale_y_continuous(  
    name="Phenotypic shannon diversity"  
) +  
scale_x_continuous(  
    name="Mean evolutionary distinctiveness",  
    breaks = breaks_extended(4)  
) +  
facet_wrap(  
    ~selection_name, scales="free"  
) +  
stat_smooth(  
    method="lm"  
) +  
stat_cor(  
    method="spearman", cor.coef.name = "rho", color="black"  
) +  
theme(legend.position = "none")
```



Again, all significant but negative. From this we can conclude that there is substantially more overlap between the information captured by phenotypic diversity and mean evolutionary distinctiveness than between phenotypic diversity metrics and mean pairwise distance. However, thus information has wildly different implications in the two contexts. Moreover, the long term trends are still very different. Thus, we still feel confident in saying that these metrics are meaningfully different. More investigation here may be worthwhile in the future.

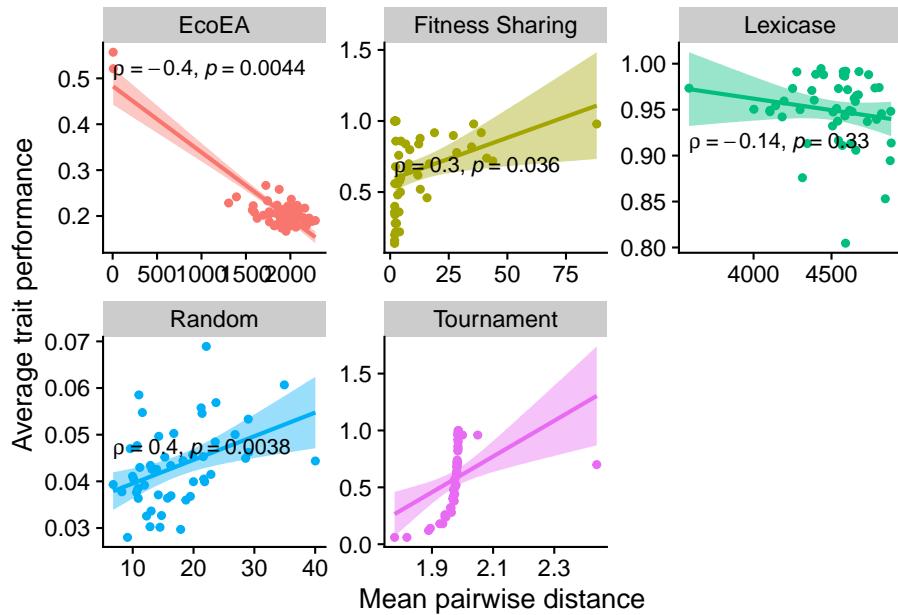
Overall, based on these end of time analyses and qualitative comparison of the temporal graphs, we conclude that phenotypic and phylogenetic diversity are meaningfully different from each other in the context of evolutionary computation. In retrospect, this finding may seem obvious. However a lot of people tend to assume that these two classes of diversity are closely related.

2.6 Relationship between diversity and success

At last, we can finally assess whether diversity leads to actually solving problems! We start out by looking at correlations between diversity success across a few different time points, although we ultimately conclude that this approach is not particularly informative.

2.6.1 Very early in run

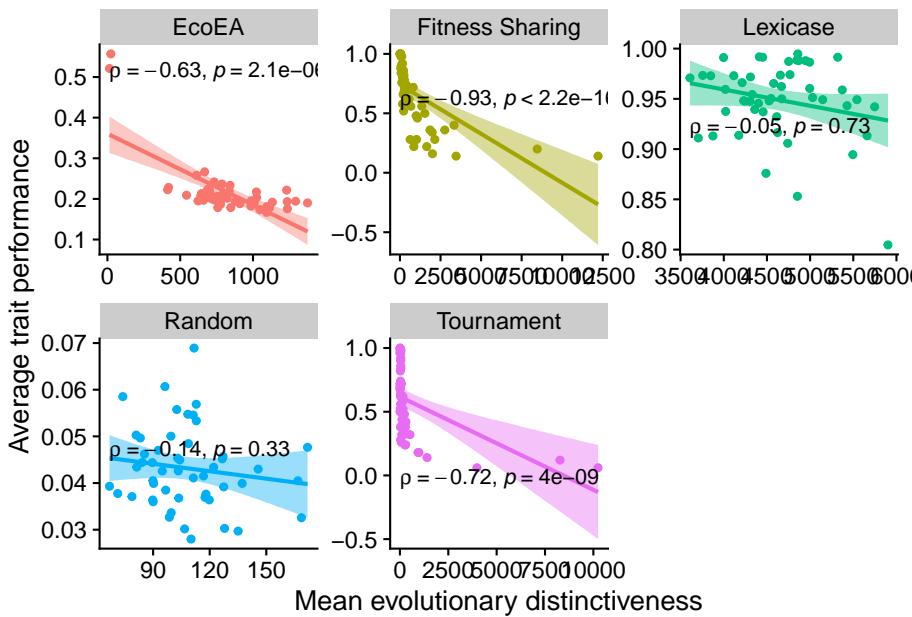
```
ggplot(
  data %>% filter(gen==25000),
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Mean pairwise distance"
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



2.6.1.2 Mean evolutionary distinctiveness

```
ggplot(
  data %>% filter(gen==25000),
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Mean evolutionary distinctiveness"
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
```

```
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



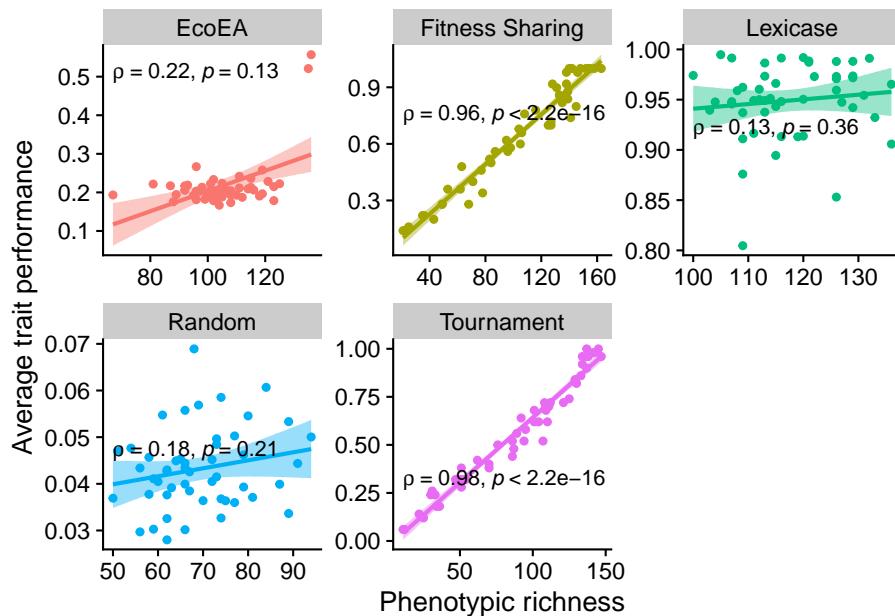
2.6.1.3 Richness

```
ggplot(
  data %>% filter(gen==25000),
  aes(
    y=elite_trait_avg,
    x=phen_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Phenotypic richness"
) +
  facet_wrap(
```

```

    ~selection_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



2.6.1.4 Shannon diversity

```

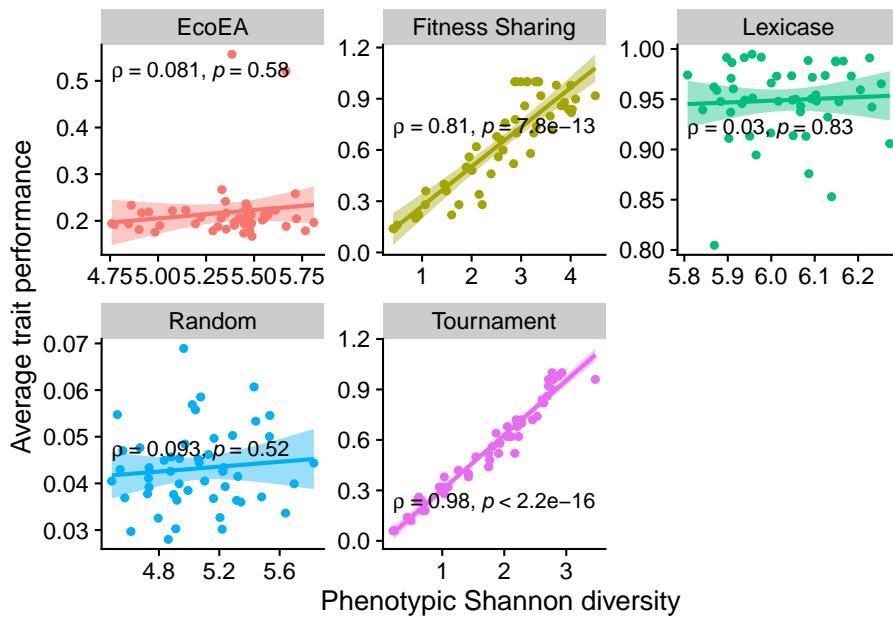
ggplot(
  data %>% filter(gen==25000),
  aes(
    y=elite_trait_avg,
    x=phen_diversity,
    color=selection_name,
    fill=selection_name
  )
) +
geom_point() +
scale_y_continuous(
  name="Average trait performance"
)

```

```

) +
  scale_x_continuous(
    name="Phenotypic Shannon diversity"
) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



2.6.2 Middle of run

2.6.2.1 Mean pairwise distance

```

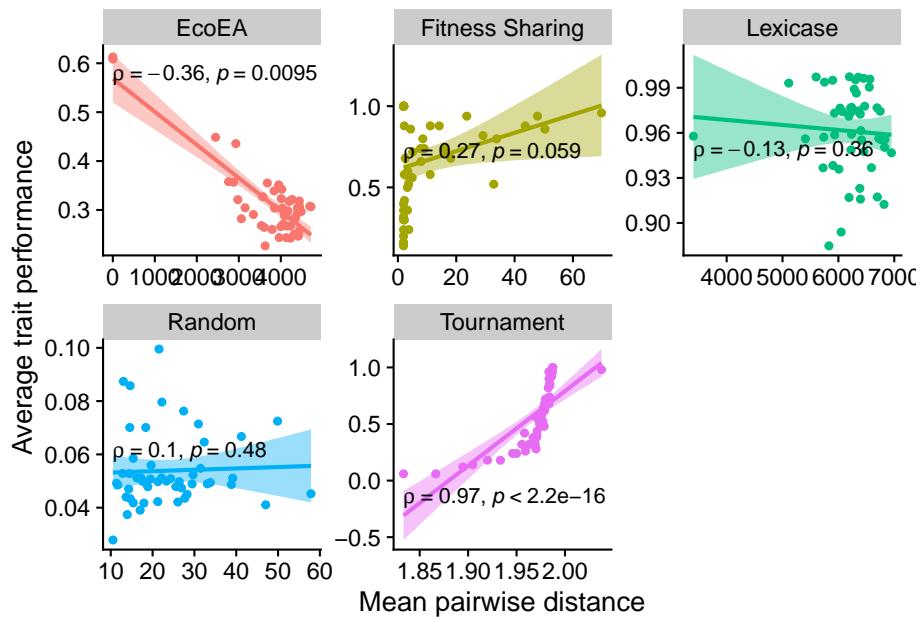
ggplot(
  data %>% filter(gen==50000),
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_pairwise_distance,

```

```

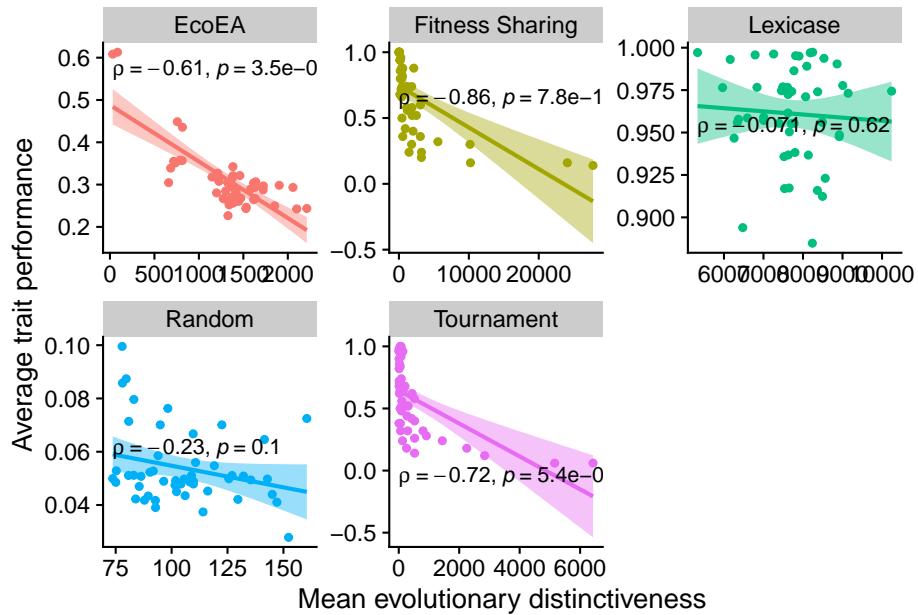
        color=selection_name,
        fill=selection_name
    )
) +
geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Mean pairwise distance"
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



2.6.2.2 Mean evolutionary distinctiveness

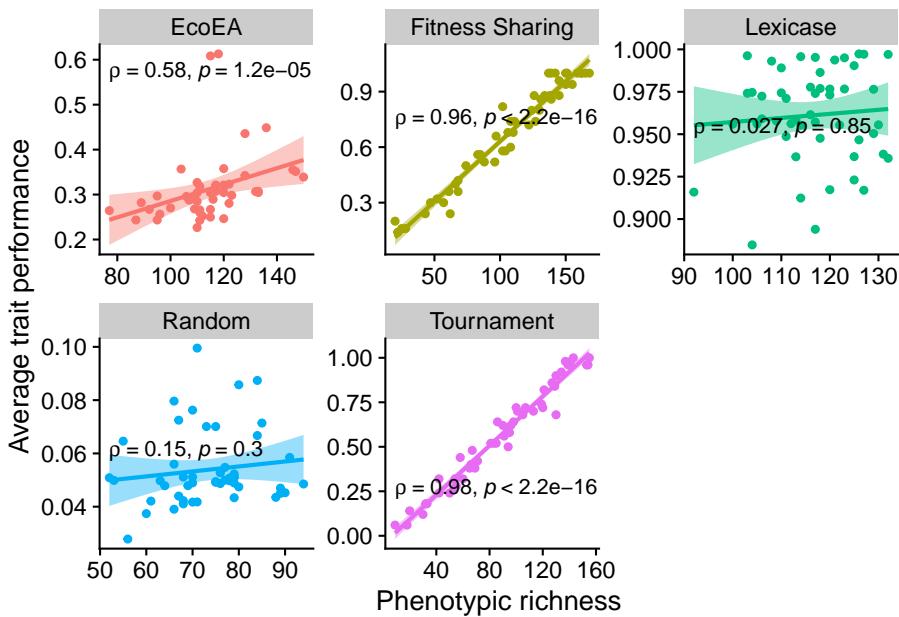
```
ggplot(
  data %>% filter(gen==50000),
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Mean evolutionary distinctiveness"
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



2.6.2.3 Richness

```
ggplot(
  data %>% filter(gen==50000),
  aes(
    y=elite_trait_avg,
    x=phen_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Phenotypic richness"
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
```

```
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



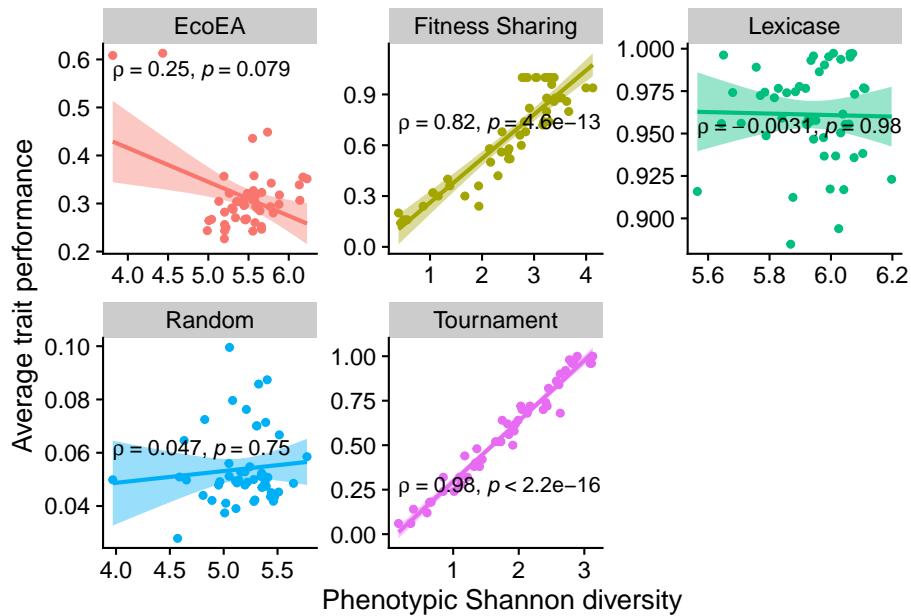
2.6.2.4 Shannon diversity

```
ggplot(
  data %>% filter(gen==50000),
  aes(
    y=elite_trait_avg,
    x=phen_diversity,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Phenotypic Shannon diversity"
) +
  facet_wrap(
```

```

    ~selection_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



2.6.3 End of run

2.6.3.1 Mean pairwise distance

```

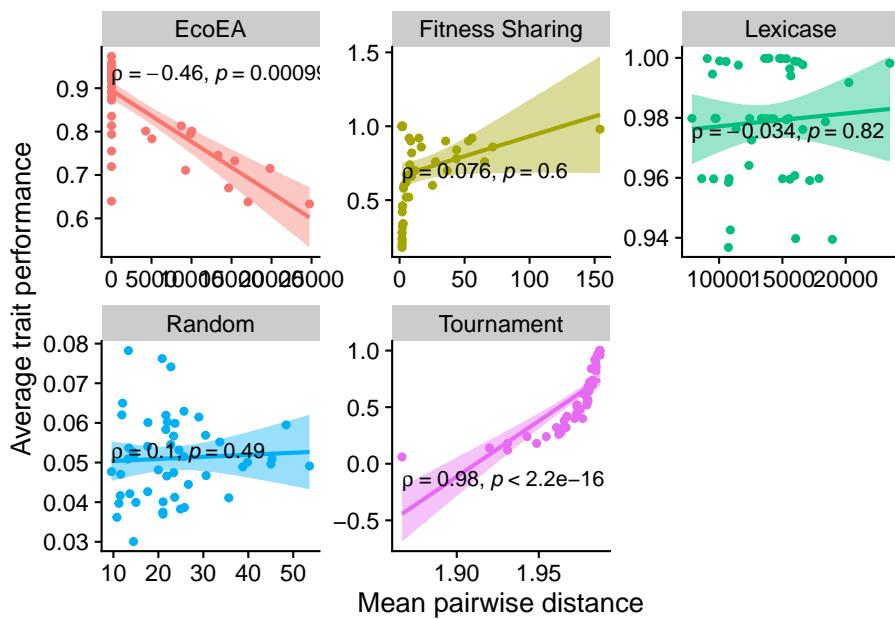
ggplot(
  final_data,
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
geom_point() +

```

```

  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Mean pairwise distance"
  ) +
  facet_wrap(
    ~selection_name, scales="free"
  ) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")

```



2.6.3.2 Mean evolutionary distinctiveness

```

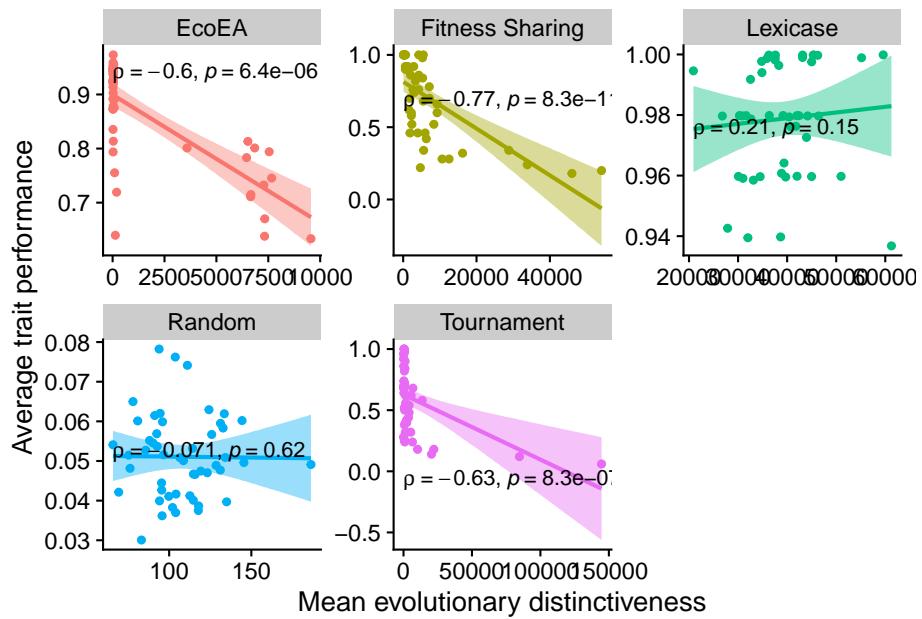
ggplot(
  final_data,
  aes(
    y=elite_trait_avg,
    x=mean_phenotype_evolutionary_distinctiveness,

```

```

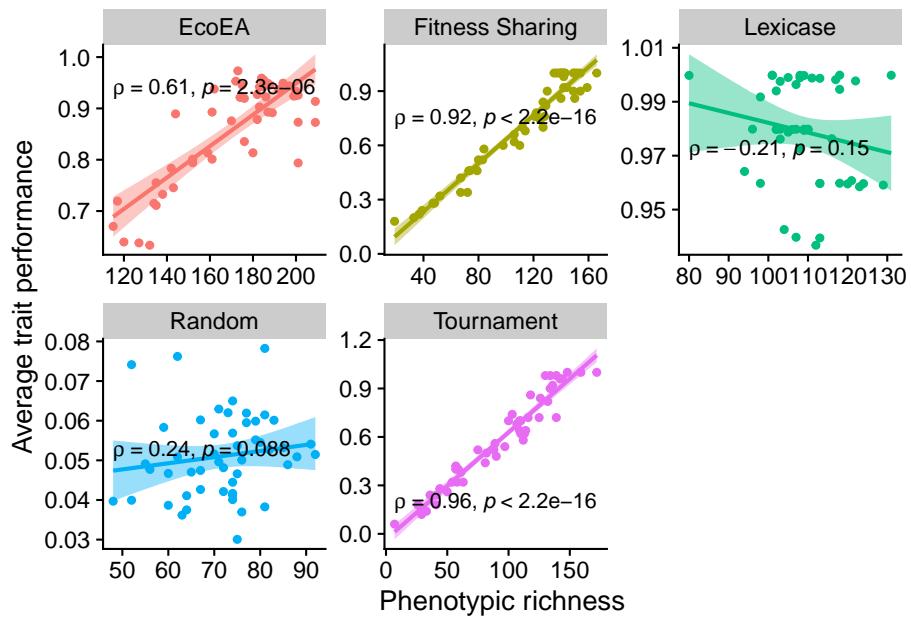
        color=selection_name,
        fill=selection_name
    )
) +
geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Mean evolutionary distinctiveness"
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



2.6.3.3 Richness

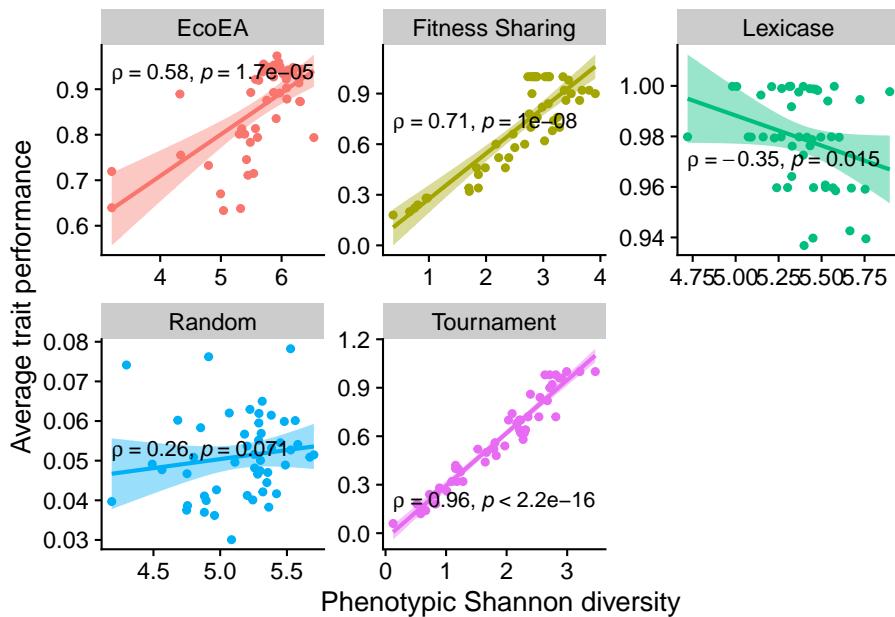
```
ggplot(
  final_data,
  aes(
    y=elite_trait_avg,
    x=phen_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Phenotypic richness"
) +
  facet_wrap(
    ~selection_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



2.6.3.4 Shannon diversity

```
ggplot(
  final_data,
  aes(
    y=elite_trait_avg,
    x=phen_diversity,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Phenotypic Shannon diversity"
  ) +
  facet_wrap(~selection_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
```

```
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



2.7 Causality analysis

Ultimately, it's hard to draw useful inferences either from these single time point analyses or from comparison of line plots. Due to the feedbacks between diversity and performance, there may be a time delay in when one affects the other.

To analyze the drivers of this feedback loop in a more rigorous way, we turn to Transfer Entropy as a metric of Granger Causality. For a more thorough description, see the paper associated with this supplement.

2.7.1 Setup

First let's define a function we'll use to calculate and output significance and effect size for these results:

```
transfer_entropy_stats <- function(res) {
  stat.test <- res %>%
    group_by(selection_name, offset) %>%
    wilcox_test(value ~ Type) %>%
    adjust_pvalue(method = "bonferroni") %>%
```

```

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

# Calculate effect sizes for these differences
effect_sizes <- res %>%
  group_by(selection_name, offset) %>%
  wilcox_effsize(value ~ Type)

stat.test$effsize <- effect_sizes$effsize
stat.test$magnitude <- effect_sizes$magnitude

stat.test %>%
  kbl() %>%
  kable_styling(
  bootstrap_options = c(
    "striped",
    "hover",
    "condensed",
    "responsive"
  )
) %>%
  scroll_box()
}

```

2.7.2 Transfer entropy from diversity to fitness

First, we calculate the information that each type of diversity gives us about future fitness.

2.7.2.1 Max pairwise distance vs. phenotypic richness

We plot the differences in transfer entropy from phylogenetic diversity to fitness and from phenotypic diversity to fitness, at a range of different lags.

```

# Calculate transfer entropy for max pairwise distance
# Time points are 10 generations, so calculating lag 1 gives us lag 10
res <- data %>% group_by(directory, selection_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(max_phenotype_pairwise_distance, 1)),
                                    discretize(lag(elite_trait_avg, 1))),
    fit_phylo_100 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(max_phenotype_pairwise_distance, 10)),
                                    discretize(lag(elite_trait_avg, 10))),
    fit_phylo_1000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(max_phenotype_pairwise_distance, 100)),
                                    discretize(lag(elite_trait_avg, 100)))
  )
}

```

```

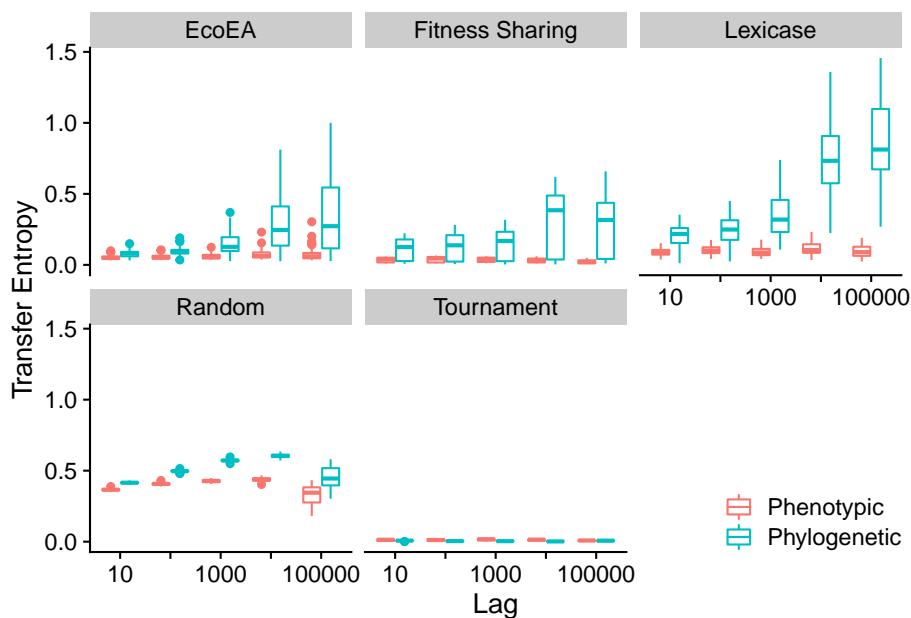
            discretize(lag(elite_trait_avg, 100))),
fit_phylo_10000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 10000
                                    discretize(lag(max_phenotype_pairwise_distance, 1000)),
                                    discretize(lag(elite_trait_avg, 1000))),
fit_phylo_100000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 100000
                                    discretize(lag(max_phenotype_pairwise_distance, 10000)),
                                    discretize(lag(elite_trait_avg, 10000))),
fit_fit_10000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
                                    discretize(lag(elite_trait_avg, 1000)),
                                    discretize(lag(max_phenotype_pairwise_distance, 1000))),
fit_fit_100000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
                                    discretize(lag(elite_trait_avg, 10000)),
                                    discretize(lag(max_phenotype_pairwise_distance, 10000))),
fit_pheno_10 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10
                                discretize(lag(pheno_num_taxa, 1)),
                                discretize(lag(elite_trait_avg, 1))),
fit_pheno_100 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100
                                discretize(lag(pheno_num_taxa, 10)),
                                discretize(lag(elite_trait_avg, 10))),
fit_pheno_1000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 1000
                                discretize(lag(pheno_num_taxa, 100)),
                                discretize(lag(elite_trait_avg, 100))),
fit_pheno_10000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10000
                                discretize(lag(pheno_num_taxa, 1000)),
                                discretize(lag(elite_trait_avg, 1000))),
fit_pheno_100000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100000
                                discretize(lag(pheno_num_taxa, 10000)),
                                discretize(lag(elite_trait_avg, 10000)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("o_10"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

# Plot transfer entropy
ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
)

```

```
) +
  geom_boxplot() +
  facet_wrap(~selection_name) +
  scale_x_discrete("Lag", labels=c("10", "", "1000", "", "100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0)) +
  scale_color_discrete("")
```



Next we calculate statistics to quantify these differences

```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.2.2 Mean pairwise distance vs. phenotypic richness

```
# Calculate transfer entropy for mean pairwise distance
# Time points are 10 generations, so calculating lag 1 gives us lag 10
res <- data %>% group_by(directory, selection_name) %>%
summarise(
  fit_phylo_10 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(mean_phenotype_pairwise_distance,
                                    discretize(lag(elite_trait_avg, 1))),
  fit_phylo_100 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(mean_phenotype_pairwise_distance,
```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.adj	p.a
EcoEA	10	value	Phenotypic	Phylogenetic	50	50	463	1.00e-07	0.0000015	***
EcoEA	100	value	Phenotypic	Phylogenetic	50	50	242	0.00e+00	0.0000000	***
EcoEA	1000	value	Phenotypic	Phylogenetic	50	50	195	0.00e+00	0.0000000	***
EcoEA	10000	value	Phenotypic	Phylogenetic	50	50	163	0.00e+00	0.0000000	***
EcoEA	100000	value	Phenotypic	Phylogenetic	50	50	247	0.00e+00	0.0000000	***
Fitness Sharing	10	value	Phenotypic	Phylogenetic	50	50	699	1.48e-04	0.0037000	**
Fitness Sharing	100	value	Phenotypic	Phylogenetic	50	50	739	4.33e-04	0.0108250	*
Fitness Sharing	1000	value	Phenotypic	Phylogenetic	50	50	751	5.89e-04	0.0147250	*
Fitness Sharing	10000	value	Phenotypic	Phylogenetic	50	50	598	7.10e-06	0.0001770	***
Fitness Sharing	100000	value	Phenotypic	Phylogenetic	50	50	415	0.00e+00	0.0000002	***
Lexicase	10	value	Phenotypic	Phylogenetic	50	50	194	0.00e+00	0.0000000	***
Lexicase	100	value	Phenotypic	Phylogenetic	50	50	183	0.00e+00	0.0000000	***
Lexicase	1000	value	Phenotypic	Phylogenetic	50	50	36	0.00e+00	0.0000000	***
Lexicase	10000	value	Phenotypic	Phylogenetic	50	50	1	0.00e+00	0.0000000	***
Lexicase	100000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	10	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	100	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	1000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	10000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	100000	value	Phenotypic	Phylogenetic	50	50	249	0.00e+00	0.0000000	***
Tournament	10	value	Phenotypic	Phylogenetic	50	50	1798	1.60e-04	0.0040000	**
Tournament	100	value	Phenotypic	Phylogenetic	50	50	2049	0.00e+00	0.0000009	***
Tournament	1000	value	Phenotypic	Phylogenetic	50	50	2232	0.00e+00	0.0000000	***
Tournament	10000	value	Phenotypic	Phylogenetic	50	50	2329	0.00e+00	0.0000000	***
Tournament	100000	value	Phenotypic	Phylogenetic	50	50	1418	2.48e-01	1.0000000	ns

```

            discretize(lag(elite_trait_avg, 10))),
fit_phylo_1000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                  discretize(lag(mean_phenotype_pairwise_distance,
                                  discretize(lag(elite_trait_avg, 100))),
fit_phylo_10000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                  discretize(lag(mean_phenotype_pairwise_distance,
                                  discretize(lag(elite_trait_avg, 1000))),
fit_phylo_100000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                  discretize(lag(mean_phenotype_pairwise_distance,
                                  discretize(lag(elite_trait_avg, 10000))),
fit_fit_10000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn.
                                  discretize(lag(elite_trait_avg, 1000)),
                                  discretize(lag(mean_phenotype_pairwise_distance,
fit_fit_100000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn.
                                  discretize(lag(elite_trait_avg, 10000)),
                                  discretize(lag(mean_phenotype_pairwise_distance,
fit_pheno_10 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                  discretize(lag(phen_num_taxa, 1)),
                                  discretize(lag(elite_trait_avg, 1))),
fit_pheno_100 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                  discretize(lag(phen_num_taxa, 10)),
                                  discretize(lag(elite_trait_avg, 10))),
fit_pheno_1000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                  discretize(lag(phen_num_taxa, 100)),
                                  discretize(lag(elite_trait_avg, 100))),
fit_pheno_10000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                  discretize(lag(phen_num_taxa, 1000)),
                                  discretize(lag(elite_trait_avg, 1000))),
fit_pheno_100000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                  discretize(lag(phen_num_taxa, 10000)),
                                  discretize(lag(elite_trait_avg, 10000)))
)

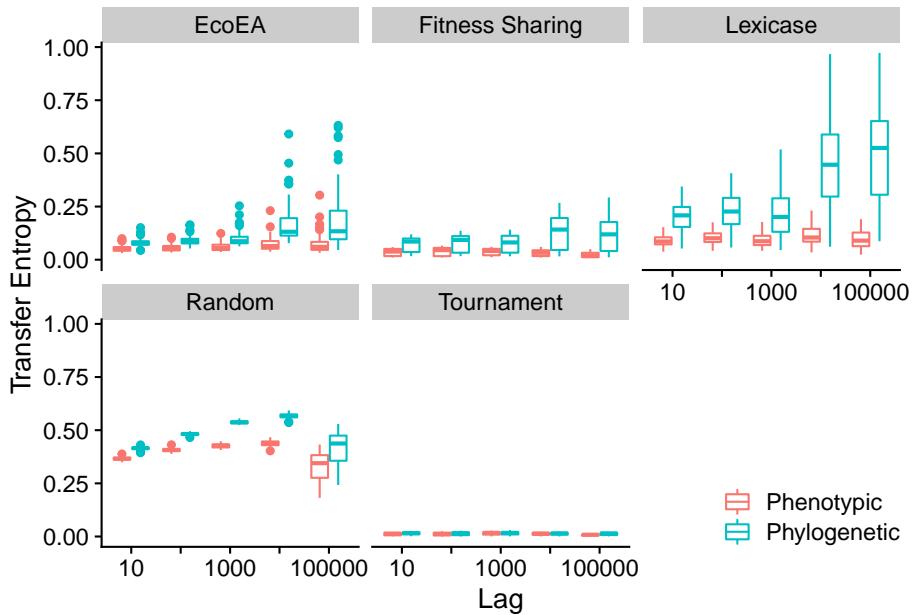
# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("o_10"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

# Plot transfer entropy
ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=.
```

```

y=value,
color=Type
)
) +
geom_boxplot() +
facet_wrap(~selection_name) +
scale_x_discrete("Lag",labels=c("10","", "1000","", "100000")) +
scale_y_continuous("Transfer Entropy") +
theme(legend.position = c(1, 0),
      legend.justification = c(1, 0)) +
scale_color_discrete("")

```



Mean and max pairwise distance appear to behave virtually identically.

```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.2.3 Mean pairwise distance vs. shannon entropy

```

# Calculate transfer entropy for mean pairwise distance
# Time points are 10 generations, so calculating lag 1 gives us lag 10
res <- data %>% group_by(directory, selection_name) %>%
summarise(
  fit_phylo_10 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 10
                                    discretize(lag(mean_phenotype_pairwise_distance, 1)),

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	
EcoEA	10	value	Phenotypic	Phylogenetic	50	50	292	0.00e+00	0.00
EcoEA	100	value	Phenotypic	Phylogenetic	50	50	251	0.00e+00	0.00
EcoEA	1000	value	Phenotypic	Phylogenetic	50	50	318	0.00e+00	0.00
EcoEA	10000	value	Phenotypic	Phylogenetic	50	50	211	0.00e+00	0.00
EcoEA	100000	value	Phenotypic	Phylogenetic	50	50	374	0.00e+00	0.00
Fitness Sharing	10	value	Phenotypic	Phylogenetic	50	50	522	5.00e-07	0.00
Fitness Sharing	100	value	Phenotypic	Phylogenetic	50	50	550	1.40e-06	0.00
Fitness Sharing	1000	value	Phenotypic	Phylogenetic	50	50	620	1.43e-05	0.00
Fitness Sharing	10000	value	Phenotypic	Phylogenetic	50	50	464	1.00e-07	0.00
Fitness Sharing	100000	value	Phenotypic	Phylogenetic	50	50	345	0.00e+00	0.00
Lexicase	10	value	Phenotypic	Phylogenetic	50	50	193	0.00e+00	0.00
Lexicase	100	value	Phenotypic	Phylogenetic	50	50	222	0.00e+00	0.00
Lexicase	1000	value	Phenotypic	Phylogenetic	50	50	336	0.00e+00	0.00
Lexicase	10000	value	Phenotypic	Phylogenetic	50	50	137	0.00e+00	0.00
Lexicase	100000	value	Phenotypic	Phylogenetic	50	50	140	0.00e+00	0.00
Random	10	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	100	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	1000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	10000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	100000	value	Phenotypic	Phylogenetic	50	50	504	3.00e-07	0.00
Tournament	10	value	Phenotypic	Phylogenetic	50	50	932	2.86e-02	0.73
Tournament	100	value	Phenotypic	Phylogenetic	50	50	1106	3.23e-01	1.00
Tournament	1000	value	Phenotypic	Phylogenetic	50	50	1209	7.80e-01	1.00
Tournament	10000	value	Phenotypic	Phylogenetic	50	50	1184	6.52e-01	1.00
Tournament	100000	value	Phenotypic	Phylogenetic	50	50	612	1.11e-05	0.00

```

            discretize(lag(elite_trait_avg, 1))),
fit_phylo_100 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 100
                                 discretize(lag(mean_phenotype_pairwise_distance, 10)),
                                 discretize(lag(elite_trait_avg, 10))),
fit_phylo_1000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 1000
                                 discretize(lag(mean_phenotype_pairwise_distance, 100)),
                                 discretize(lag(elite_trait_avg, 100))),
fit_phylo_10000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 10000
                                 discretize(lag(mean_phenotype_pairwise_distance, 1000)),
                                 discretize(lag(elite_trait_avg, 1000))),
fit_phylo_100000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 100000
                                 discretize(lag(mean_phenotype_pairwise_distance, 10000)),
                                 discretize(lag(elite_trait_avg, 10000))),
fit_fit_10000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
                                 discretize(lag(elite_trait_avg, 1000)),
                                 discretize(lag(mean_phenotype_pairwise_distance, 1000))),
fit_fit_100000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
                                 discretize(lag(elite_trait_avg, 10000)),
                                 discretize(lag(mean_phenotype_pairwise_distance, 10000))),
fit_pheno_10 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10
                                 discretize(lag(phen_diversity, 1)),
                                 discretize(lag(elite_trait_avg, 1))),
fit_pheno_100 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100
                                 discretize(lag(phen_diversity, 10)),
                                 discretize(lag(elite_trait_avg, 10))),
fit_pheno_1000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 1000
                                 discretize(lag(phen_diversity, 100)),
                                 discretize(lag(elite_trait_avg, 100))),
fit_pheno_10000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10000
                                 discretize(lag(phen_diversity, 1000)),
                                 discretize(lag(elite_trait_avg, 1000))),
fit_pheno_100000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100000
                                 discretize(lag(phen_diversity, 10000)),
                                 discretize(lag(elite_trait_avg, 10000)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("o_10"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

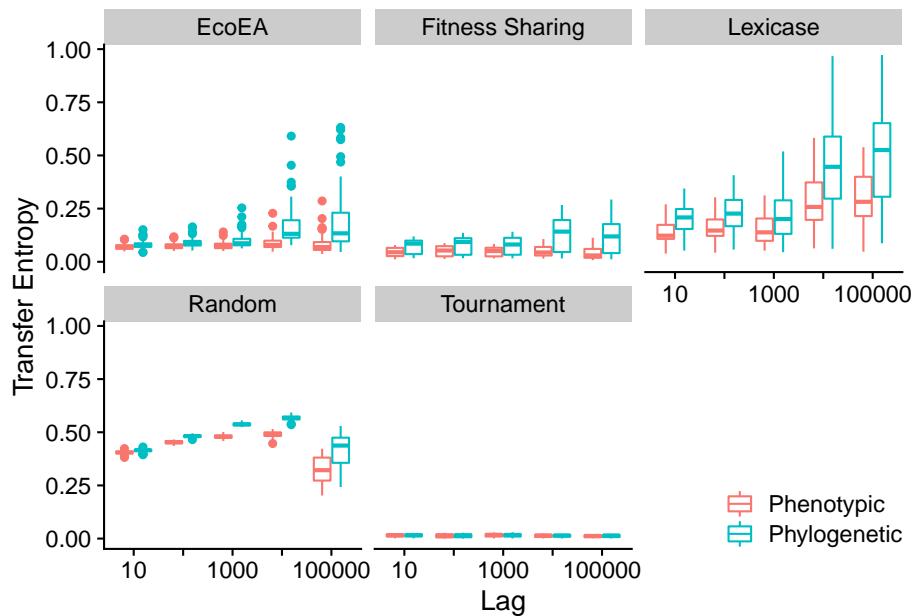
# Plot transfer entropy
ggplot(

```

```

res %>% filter(str_detect(name, "fit_ph*")),
aes(
  x=as.factor(offset),
  y=value,
  color=Type
)
) +
geom_boxplot() +
facet_wrap(~selection_name) +
scale_x_discrete("Lag",labels=c("10","","1000","","100000")) +
scale_y_continuous("Transfer Entropy") +
theme(legend.position = c(1, 0),
      legend.justification = c(1, 0)) +
scale_color_discrete("")

```



Looks like Shannon Diversity is more predictive of fitness than phenotypic richness (at least for lexicase), although still not as much as mean pairwise distance.

```

# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.adj	p.a
EcoEA	10	value	Phenotypic	Phylogenetic	50	50	789	1.50e-03	0.0375000	*
EcoEA	100	value	Phenotypic	Phylogenetic	50	50	671	6.66e-05	0.0016650	**
EcoEA	1000	value	Phenotypic	Phylogenetic	50	50	664	5.43e-05	0.0013575	**
EcoEA	10000	value	Phenotypic	Phylogenetic	50	50	293	0.00e+00	0.0000000	***
EcoEA	100000	value	Phenotypic	Phylogenetic	50	50	429	0.00e+00	0.0000004	***
Fitness Sharing	10	value	Phenotypic	Phylogenetic	50	50	705	1.74e-04	0.0043500	**
Fitness Sharing	100	value	Phenotypic	Phylogenetic	50	50	725	2.99e-04	0.0074750	**
Fitness Sharing	1000	value	Phenotypic	Phylogenetic	50	50	782	1.27e-03	0.0317500	*
Fitness Sharing	10000	value	Phenotypic	Phylogenetic	50	50	612	1.11e-05	0.0002775	***
Fitness Sharing	100000	value	Phenotypic	Phylogenetic	50	50	566	2.40e-06	0.0000612	***
Lexicase	10	value	Phenotypic	Phylogenetic	50	50	560	2.00e-06	0.0000500	***
Lexicase	100	value	Phenotypic	Phylogenetic	50	50	608	9.80e-06	0.0002440	***
Lexicase	1000	value	Phenotypic	Phylogenetic	50	50	827	3.58e-03	0.0895000	ns
Lexicase	10000	value	Phenotypic	Phylogenetic	50	50	608	9.80e-06	0.0002440	***
Lexicase	100000	value	Phenotypic	Phylogenetic	50	50	636	2.34e-05	0.0005850	***
Random	10	value	Phenotypic	Phylogenetic	50	50	402	0.00e+00	0.0000001	***
Random	100	value	Phenotypic	Phylogenetic	50	50	1	0.00e+00	0.0000000	***
Random	1000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	10000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***
Random	100000	value	Phenotypic	Phylogenetic	50	50	428	0.00e+00	0.0000004	***
Tournament	10	value	Phenotypic	Phylogenetic	50	50	1244	9.70e-01	1.0000000	ns
Tournament	100	value	Phenotypic	Phylogenetic	50	50	1220	8.39e-01	1.0000000	ns
Tournament	1000	value	Phenotypic	Phylogenetic	50	50	1280	8.39e-01	1.0000000	ns
Tournament	10000	value	Phenotypic	Phylogenetic	50	50	1250	1.00e+00	1.0000000	ns
Tournament	100000	value	Phenotypic	Phylogenetic	50	50	1009	9.73e-02	1.0000000	ns

2.7.2.4 Mean evolutionary distinctiveness vs. phenotypic richness

```

# Calculate transfer entropy for mean evolutionary distinctiveness
# Time points are 10 generations, so calculating lag 1 gives us lag 10
res <- data %>% group_by(directory, selection_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, l
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1)),
                                    discretize(lag(elite_trait_avg, 1))),
    fit_phylo_100 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, l
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10)),
                                    discretize(lag(elite_trait_avg, 10))),
    fit_phylo_1000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, l
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 100)),
                                    discretize(lag(elite_trait_avg, 100))),
    fit_phylo_10000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, l
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1000)),
                                    discretize(lag(elite_trait_avg, 1000))),
    fit_phylo_100000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, l
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10000)),
                                    discretize(lag(elite_trait_avg, 10000))),
    fit_fit_10000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. .
                                    discretize(lag(elite_trait_avg, 1000)),
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1000))),
    fit_fit_100000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. .
                                    discretize(lag(elite_trait_avg, 10000)),
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10000))),
    fit_pheno_10 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, l
                                    discretize(lag(pheno_num_taxa, 1)),
                                    discretize(lag(elite_trait_avg, 1))),
    fit_pheno_100 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                    discretize(lag(pheno_num_taxa, 10)),
                                    discretize(lag(elite_trait_avg, 10))),
    fit_pheno_1000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                    discretize(lag(pheno_num_taxa, 100)),
                                    discretize(lag(elite_trait_avg, 100))),
    fit_pheno_10000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                    discretize(lag(pheno_num_taxa, 1000)),
                                    discretize(lag(elite_trait_avg, 1000))),
    fit_pheno_100000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit,
                                    discretize(lag(pheno_num_taxa, 10000)),
                                    discretize(lag(elite_trait_avg, 10000)))
  )
# Turn Transfer Entropy columns into rows

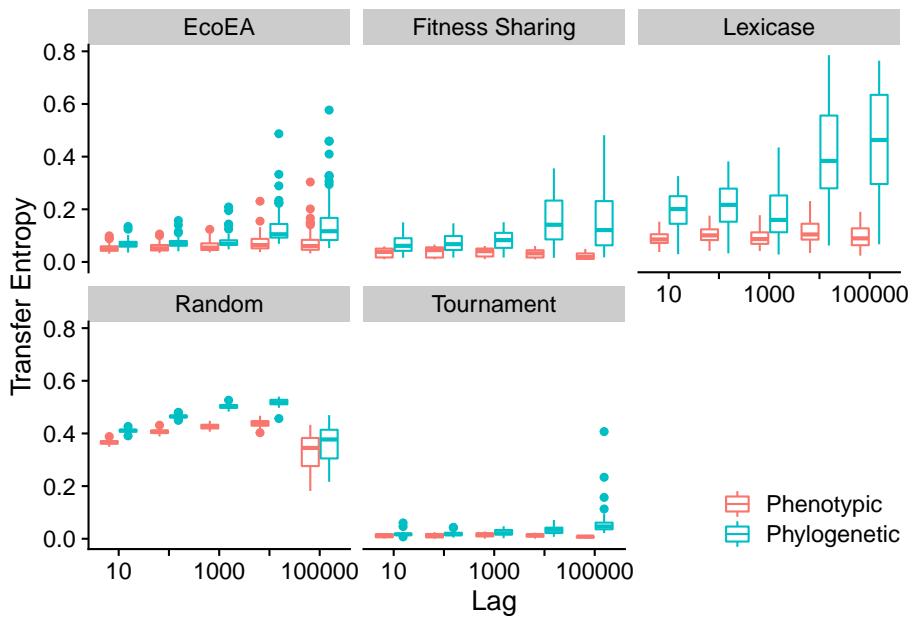
```

```

res <- res %>% pivot_longer(cols=contains("o_10"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res>Type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

# Plot transfer entropy
ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name) +
  scale_x_discrete("Lag", labels=c("10", "", "1000", "", "100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0)) +
  scale_color_discrete("")

```



Interestingly, mean evolutionary distinctiveness looks fairly comparable to mean pairwise distance (although note that the y axis doesn't go quite as far).

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	
EcoEA	10	value	Phenotypic	Phylogenetic	50	50	543	1.10e-06	0.00
EcoEA	100	value	Phenotypic	Phylogenetic	50	50	540	1.00e-06	0.00
EcoEA	1000	value	Phenotypic	Phylogenetic	50	50	629	1.89e-05	0.00
EcoEA	10000	value	Phenotypic	Phylogenetic	50	50	381	0.00e+00	0.00
EcoEA	100000	value	Phenotypic	Phylogenetic	50	50	462	1.00e-07	0.00
Fitness Sharing	10	value	Phenotypic	Phylogenetic	50	50	529	7.00e-07	0.00
Fitness Sharing	100	value	Phenotypic	Phylogenetic	50	50	553	1.60e-06	0.00
Fitness Sharing	1000	value	Phenotypic	Phylogenetic	50	50	428	0.00e+00	0.00
Fitness Sharing	10000	value	Phenotypic	Phylogenetic	50	50	330	0.00e+00	0.00
Fitness Sharing	100000	value	Phenotypic	Phylogenetic	50	50	146	0.00e+00	0.00
Lexicase	10	value	Phenotypic	Phylogenetic	50	50	237	0.00e+00	0.00
Lexicase	100	value	Phenotypic	Phylogenetic	50	50	272	0.00e+00	0.00
Lexicase	1000	value	Phenotypic	Phylogenetic	50	50	509	3.00e-07	0.00
Lexicase	10000	value	Phenotypic	Phylogenetic	50	50	208	0.00e+00	0.00
Lexicase	100000	value	Phenotypic	Phylogenetic	50	50	230	0.00e+00	0.00
Random	10	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	100	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	1000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00
Random	10000	value	Phenotypic	Phylogenetic	50	50	2	0.00e+00	0.00
Random	100000	value	Phenotypic	Phylogenetic	50	50	965	4.98e-02	1.00
Tournament	10	value	Phenotypic	Phylogenetic	50	50	655	4.16e-05	0.00
Tournament	100	value	Phenotypic	Phylogenetic	50	50	716	2.35e-04	0.00
Tournament	1000	value	Phenotypic	Phylogenetic	50	50	736	4.00e-04	0.00
Tournament	10000	value	Phenotypic	Phylogenetic	50	50	187	0.00e+00	0.00
Tournament	100000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.00

```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.2.5 Mean evolutionary distinctiveness vs. shannon entropy

```
# Calculate transfer entropy for mean evolutionary distinctiveness
# Time points are 10 generations, so calculating lag 1 gives us lag 10
res <- data %>% group_by(directory, selection_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1)),
                                    discretize(lag(elite_trait_avg, 1))),
    fit_phylo_100 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10)),
                                    discretize(lag(elite_trait_avg, 10))),
    fit_phylo_1000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit,
                                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 100)),
                                    discretize(lag(elite_trait_avg, 100)))
```

```

        discretize(lag(mean_phenotype_evolutionary_distinctiveness,
        discretize(lag(elite_trait_avg, 100))),
fit_phylo_10000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 10000
        discretize(lag(mean_phenotype_evolutionary_distinctiveness,
        discretize(lag(elite_trait_avg, 1000))),
fit_phylo_100000 = condinformation(discretize(elite_trait_avg), # TE phylo -> fit, lag 100000
        discretize(lag(mean_phenotype_evolutionary_distinctiveness,
        discretize(lag(elite_trait_avg, 10000))),
fit_fit_10000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
        discretize(lag(elite_trait_avg, 1000)),
        discretize(lag(mean_phenotype_evolutionary_distinctiveness,
fit_fit_100000 = condinformation(discretize(elite_trait_avg), # Mutual info btwn. fit and lag
        discretize(lag(elite_trait_avg, 10000)),
        discretize(lag(mean_phenotype_evolutionary_distinctiveness,
fit_pheno_10 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10
        discretize(lag(phen_diversity, 1)),
        discretize(lag(elite_trait_avg, 1))),
fit_pheno_100 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100
        discretize(lag(phen_diversity, 10)),
        discretize(lag(elite_trait_avg, 10))),
fit_pheno_1000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 1000
        discretize(lag(phen_diversity, 100)),
        discretize(lag(elite_trait_avg, 100))),
fit_pheno_10000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 10000
        discretize(lag(phen_diversity, 1000)),
        discretize(lag(elite_trait_avg, 1000))),
fit_pheno_100000 = condinformation(discretize(elite_trait_avg), # TE pheno -> fit, lag 100000
        discretize(lag(phen_diversity, 10000)),
        discretize(lag(elite_trait_avg, 10000)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("o_10"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res>Type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

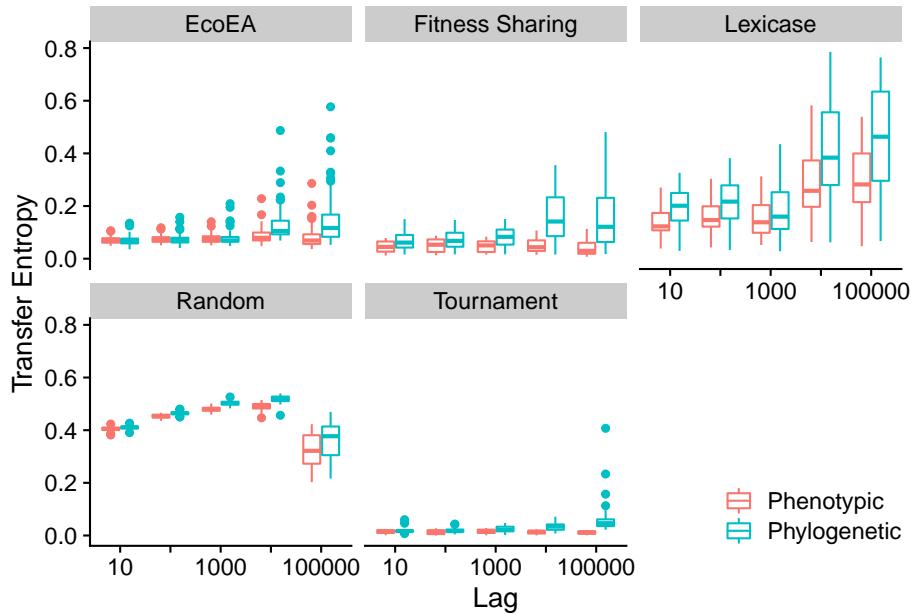
# Plot transfer entropy
ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type

```

```

    )
) +
geom_boxplot() +
facet_wrap(~selection_name) +
scale_x_discrete("Lag", labels=c("10", "", "1000", "", "100000")) +
scale_y_continuous("Transfer Entropy") +
theme(legend.position = c(1, 0),
      legend.justification = c(1, 0)) +
scale_color_discrete("")

```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.3 Transfer entropy between types of diversity

While we're calculating transfer entropy, we might as well also calculate it between phenotypic diversity and phylogenetic diversity, as these could potentially also be in a feedback loop.

2.7.3.1 Max pairwise distance and phenotypic richness

```
res <- data %>% group_by(directory, selection_name) %>%
summarise(
  phen_phylo_10 = condinformation(discretize(phen_num_taxa),
                                    discretize(lag(max_phenotype_pairwise_distance,
```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.adj	p.a
EcoEA	10	value	Phenotypic	Phylogenetic	50	50	1386	3.50e-01	1.0000000	ns
EcoEA	100	value	Phenotypic	Phylogenetic	50	50	1371	4.06e-01	1.0000000	ns
EcoEA	1000	value	Phenotypic	Phylogenetic	50	50	1352	4.84e-01	1.0000000	ns
EcoEA	10000	value	Phenotypic	Phylogenetic	50	50	517	4.00e-07	0.0000111	***
EcoEA	100000	value	Phenotypic	Phylogenetic	50	50	553	1.60e-06	0.0000392	***
Fitness Sharing	10	value	Phenotypic	Phylogenetic	50	50	813	2.62e-03	0.0655000	ns
Fitness Sharing	100	value	Phenotypic	Phylogenetic	50	50	839	4.66e-03	0.1165000	ns
Fitness Sharing	1000	value	Phenotypic	Phylogenetic	50	50	624	1.62e-05	0.0004050	***
Fitness Sharing	10000	value	Phenotypic	Phylogenetic	50	50	448	0.00e+00	0.0000008	***
Fitness Sharing	100000	value	Phenotypic	Phylogenetic	50	50	390	0.00e+00	0.0000001	***
Lexicase	10	value	Phenotypic	Phylogenetic	50	50	642	2.81e-05	0.0007025	***
Lexicase	100	value	Phenotypic	Phylogenetic	50	50	693	1.25e-04	0.0031250	**
Lexicase	1000	value	Phenotypic	Phylogenetic	50	50	1040	1.49e-01	1.0000000	ns
Lexicase	10000	value	Phenotypic	Phylogenetic	50	50	764	8.17e-04	0.0204250	*
Lexicase	100000	value	Phenotypic	Phylogenetic	50	50	715	2.29e-04	0.0057250	**
Random	10	value	Phenotypic	Phylogenetic	50	50	721	2.69e-04	0.0067250	**
Random	100	value	Phenotypic	Phylogenetic	50	50	278	0.00e+00	0.0000000	***
Random	1000	value	Phenotypic	Phylogenetic	50	50	89	0.00e+00	0.0000000	***
Random	10000	value	Phenotypic	Phylogenetic	50	50	152	0.00e+00	0.0000000	***
Random	100000	value	Phenotypic	Phylogenetic	50	50	881	1.11e-02	0.2775000	ns
Tournament	10	value	Phenotypic	Phylogenetic	50	50	1014	1.04e-01	1.0000000	ns
Tournament	100	value	Phenotypic	Phylogenetic	50	50	861	7.40e-03	0.1850000	ns
Tournament	1000	value	Phenotypic	Phylogenetic	50	50	774	1.05e-03	0.0262500	*
Tournament	10000	value	Phenotypic	Phylogenetic	50	50	211	0.00e+00	0.0000000	***
Tournament	100000	value	Phenotypic	Phylogenetic	50	50	0	0.00e+00	0.0000000	***

```

phen_phylo_100 = condinformation(discretize(lag(phen_num_taxa, 1))),
discretize(lag(max_phenotype_pairwise_distance,
discretize(lag(phen_num_taxa, 10))),
phen_phylo_1000 = condinformation(discretize(phen_num_taxa),
discretize(lag(max_phenotype_pairwise_distance,
discretize(lag(phen_num_taxa, 100))),
phen_phylo_10000 = condinformation(discretize(phen_num_taxa),
discretize(lag(max_phenotype_pairwise_distance,
discretize(lag(phen_num_taxa, 1000))),
phen_phylo_100000 = condinformation(discretize(phen_num_taxa),
discretize(lag(max_phenotype_pairwise_distance,
discretize(lag(phen_num_taxa, 10000))),

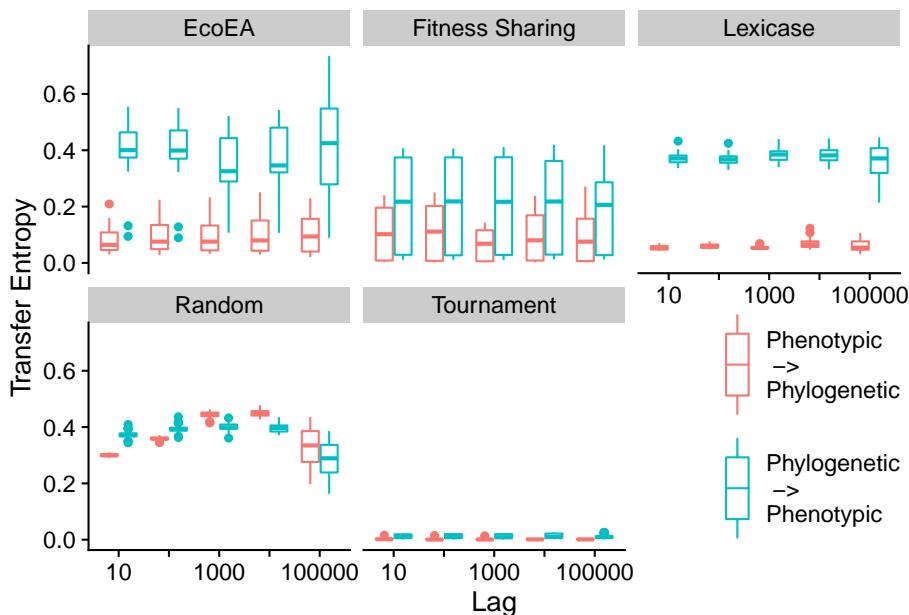
phylo_pheno_10 = condinformation(discretize(max_phenotype_pairwise_distance),
discretize(lag(phen_num_taxa, 1)),
discretize(lag(max_phenotype_pairwise_distance,
phylo_pheno_100 = condinformation(discretize(max_phenotype_pairwise_distance),
discretize(lag(phen_num_taxa, 10)),
discretize(lag(max_phenotype_pairwise_distance,
phylo_pheno_1000 = condinformation(discretize(max_phenotype_pairwise_distance),
discretize(lag(phen_num_taxa, 100)),
discretize(lag(max_phenotype_pairwise_distance,
phylo_pheno_10000 = condinformation(discretize(max_phenotype_pairwise_distance),
discretize(lag(phen_num_taxa, 1000)),
discretize(lag(max_phenotype_pairwise_distance,
phylo_pheno_100000 = condinformation(discretize(max_phenotype_pairwise_distance),
discretize(lag(phen_num_taxa, 10000)),
discretize(lag(max_phenotype_pairwise_distance,
))

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*\$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenic",
                       TRUE ~ "Phylogenetic\n\t->\nPhenotypic")

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
)

```

```
) +
  geom_boxplot() +
  facet_wrap(~selection_name) +
  scale_x_discrete("Lag", labels=c("10", "", "1000", "", "100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0)) +
  scale_color_discrete("")
```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.3.2 Mean pairwise distance and phenotypic richness

```
res <- data %>% group_by(directory, selection_name) %>%
  summarise(
    phen_phylo_10 = condinformation(discretize(phen_num_taxa),
                                      discretize(lag(mean_phenotype_pairwise_distance, 1)),
                                      discretize(lag(phen_num_taxa, 1))),
    phen_phylo_100 = condinformation(discretize(phen_num_taxa),
                                      discretize(lag(mean_phenotype_pairwise_distance, 10)),
                                      discretize(lag(phen_num_taxa, 10))),
    pheno_phylo_1000 = condinformation(discretize(phen_num_taxa),
                                         discretize(lag(mean_phenotype_pairwise_distance, 100)))
```

selection_name	offset	.y.	group1	group2	n1
EcoEA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50

```

            discretize(lag(phen_num_taxa, 100))),
pheno_phylo_10000 = condinformation(discretize(phen_num_taxa),
                                      discretize(lag(mean_phenotype_pairwise_distance, 1000)),
                                      discretize(lag(phen_num_taxa, 1000))),
pheno_phylo_100000 = condinformation(discretize(phen_num_taxa),
                                       discretize(lag(mean_phenotype_pairwise_distance, 10000)),
                                       discretize(lag(phen_num_taxa, 10000))),
```

```

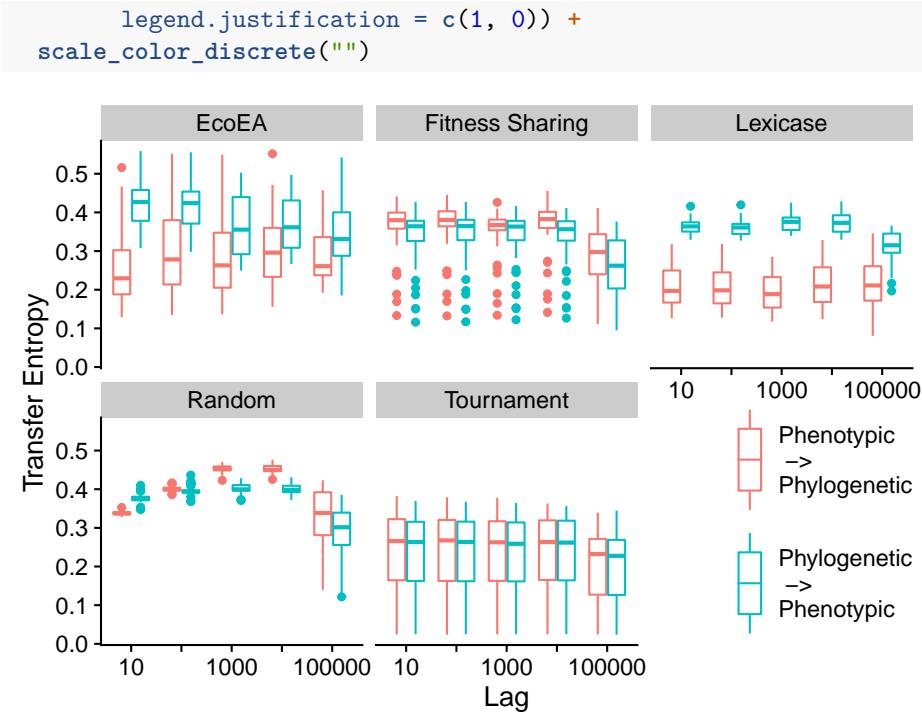
phylo_pheno_10 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                 discretize(lag(phen_num_taxa, 1)),
                                 discretize(lag(mean_phenotype_pairwise_distance, 1))),
phylo_pheno_100 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_num_taxa, 10)),
                                   discretize(lag(mean_phenotype_pairwise_distance, 10))),
phylo_pheno_1000 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phen_num_taxa, 100)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 100))),
phylo_pheno_10000 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                     discretize(lag(phen_num_taxa, 1000)),
                                     discretize(lag(mean_phenotype_pairwise_distance, 1000))),
phylo_pheno_100000 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                       discretize(lag(phen_num_taxa, 10000)),
                                       discretize(lag(mean_phenotype_pairwise_distance, 10000)))
)
```

```

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res>Type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n",
```

```

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name) +
  scale_x_discrete("Lag",labels=c("10","",""1000","",""100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
```



Here, on the other hand, there's a pretty notable difference between mean and max pairwise distance. Looks like max pairwise distance is in general more predictive of phenotypic richness than mean pairwise distance is (for fitness sharing, mean richness even has a high transfer entropy to MPD than the other way around).

```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.3.3 Mean pairwise distance and shannon diversity

```
res <- data %>% group_by(directory, selection_name) %>%
summarise(
  phen_phylo_10 = condinformation(discretize(pheno_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance,
                                    discretize(lag(pheno_diversity, 1))),
  phen_phylo_100 = condinformation(discretize(pheno_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance,
                                    discretize(lag(pheno_diversity, 10))),
  pheno_phylo_1000 = condinformation(discretize(pheno_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance,
                                    discretize(lag(pheno_diversity, 100))),
```

selection_name	offset	.y.	group1	group2	n1	n2	statist
EcoEA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	34
EcoEA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	48
EcoEA	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	66
EcoEA	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	70
EcoEA	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	73
Fitness Sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	160
Fitness Sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	165
Fitness Sharing	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	138
Fitness Sharing	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	174
Fitness Sharing	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	156
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	22
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	180
Random	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	249
Random	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	249
Random	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	168
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	129
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	130
Tournament	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	129
Tournament	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	129
Tournament	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	127

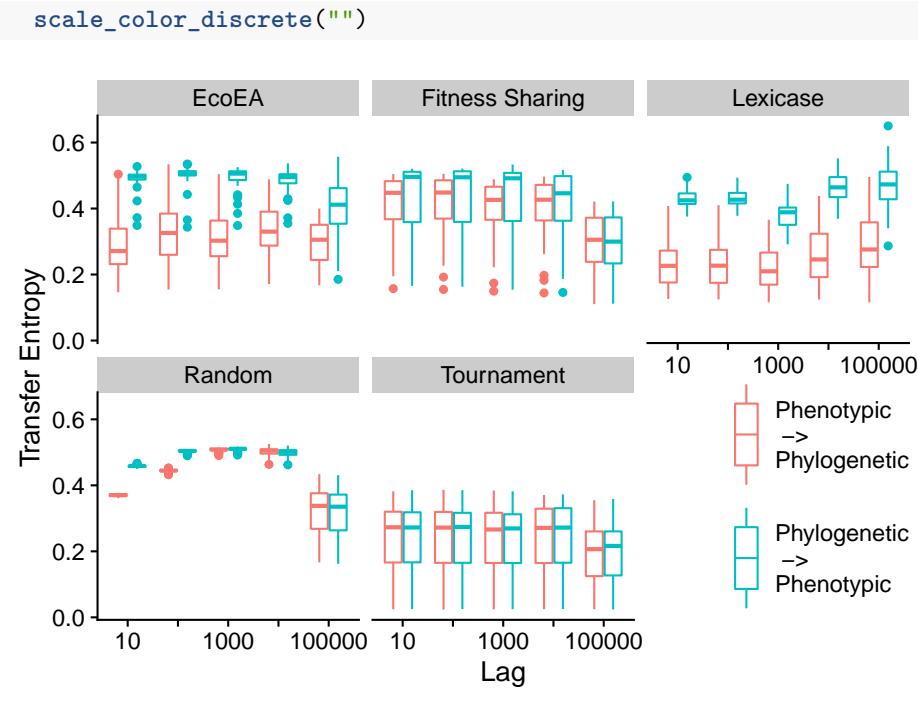
```

pheno_phylo_10000 = condinformation(discretize(phen_diversity),
                                      discretize(lag(mean_phenotype_pairwise_distance,
                                      discretize(lag(phen_diversity, 1000))),
pheno_phylo_100000 = condinformation(discretize(phen_diversity),
                                      discretize(lag(mean_phenotype_pairwise_distance,
                                      discretize(lag(phen_diversity, 10000))),
phylo_pheno_10 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_diversity, 1)),
                                   discretize(lag(mean_phenotype_pairwise_distance,
phylo_pheno_100 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_diversity, 10)),
                                   discretize(lag(mean_phenotype_pairwise_distance,
phylo_pheno_1000 = condinflation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_diversity, 100)),
                                   discretize(lag(mean_phenotype_pairwise_distance,
phylo_pheno_10000 = condinflation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_diversity, 1000)),
                                   discretize(lag(mean_phenotype_pairwise_distance,
phylo_pheno_100000 = condinflation(discretize(mean_phenotype_pairwise_distance),
                                   discretize(lag(phen_diversity, 10000)),
                                   discretize(lag(mean_phenotype_pairwise_distance
))

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylog

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name) +
  scale_x_discrete("Lag", labels=c("10","","1000","","100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0)) +

```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.3.4 Mean evolutionary distinctiveness and phenotypic richness

```
res <- data %>% group_by(directory, selection_name) %>%
summarise(
  phen_phylo_10 = condinformation(
    discretize(phen_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1)),
    discretize(lag(phen_num_taxa, 1))),
  phen_phylo_100 = condinformation(
    discretize(phen_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10)),
    discretize(lag(phen_num_taxa, 10))),
  pheno_phylo_1000 = condinformation(
    discretize(phen_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 100)),
    discretize(lag(phen_num_taxa, 100))),
  pheno_phylo_10000 = condinformation(
    discretize(phen_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1000))),
```

selection_name	offset	.y.	group1	group2	n1
EcoEA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50

```

            discretize(lag(pheno_num_taxa, 1000))),  

pheno_phylo_100000 = condinformation(  

                        discretize(pheno_num_taxa),  

                        discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10000)),  

                        discretize(lag(pheno_num_taxa, 1000))),  

phylo_pheno_10 = condinformation(  

                    discretize(mean_phenotype_evolutionary_distinctiveness),  

                    discretize(lag(pheno_num_taxa, 1)),  

                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1))),  

phylo_pheno_100 = condinformation(  

                    discretize(mean_phenotype_evolutionary_distinctiveness),  

                    discretize(lag(pheno_num_taxa, 10)),  

                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10))),  

phylo_pheno_1000 = condinformation(  

                    discretize(mean_phenotype_evolutionary_distinctiveness),  

                    discretize(lag(pheno_num_taxa, 100)),  

                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 100))),  

phylo_pheno_10000 = condinformation(  

                    discretize(mean_phenotype_evolutionary_distinctiveness),  

                    discretize(lag(pheno_num_taxa, 1000)),  

                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1000))),  

phylo_pheno_100000 = condinformation(  

                    discretize(mean_phenotype_evolutionary_distinctiveness),  

                    discretize(lag(pheno_num_taxa, 10000)),  

                    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10000)))  

)  

# Turn Transfer Entropy columns into rows  

res <- res %>% pivot_longer(cols=contains("phylo"))  

# Pull lag into a column  

res$offset <- str_extract(res$name, "[[:digit:]]*$")  

# Make column indicating direction of transfer entropy  

res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n",  

ggplot(  

  res,  

  aes(  

    x=as.factor(offset),  

    y=value,  

    color=Type  

  )  

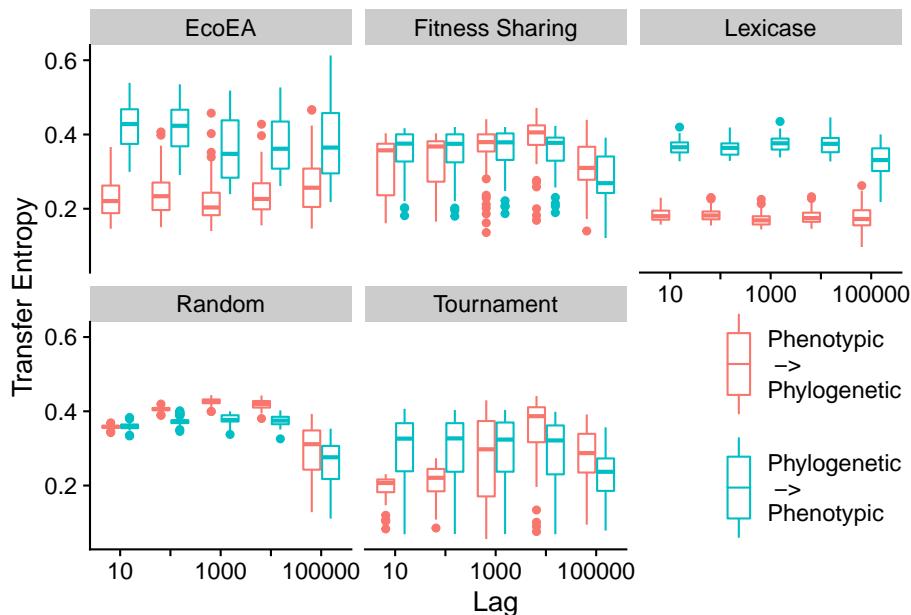
) +  

  geom_boxplot() +  

  facet_wrap(~selection_name) +

```

```
scale_x_discrete("Lag", labels=c("10", "", "1000", "", "100000")) +
  scale_y_continuous("Transfer Entropy") +
  theme(legend.position = c(1, 0),
        legend.justification = c(1, 0)) +
  scale_color_discrete("")
```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

2.7.3.5 Mean evolutionary distinctiveness and shannon diversity

```
res <- data %>% group_by(directory, selection_name) %>%
  summarise(
    phen_phylo_10 = condinformation(
      discretize(phen_diversity),
      discretize(lag(mean_phenotype_evolutionary_distinctiveness,
                    1)),
      discretize(lag(phen_diversity, 10))),
    phen_phylo_100 = condinformation(
      discretize(phen_diversity),
      discretize(lag(mean_phenotype_evolutionary_distinctiveness,
                    10)),
      discretize(lag(phen_diversity, 100))),
    pheno_phylo_1000 = condinformation(
      discretize(phen_diversity),
      discretize(lag(mean_phenotype_evolutionary_distinctiveness,
                    100)))
```

selection_name	offset	.y.	group1	group2	n1	n2	statist
EcoEA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	2
EcoEA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	9
EcoEA	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	17
EcoEA	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	19
EcoEA	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	44
Fitness Sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	81
Fitness Sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	94
Fitness Sharing	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	132
Fitness Sharing	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	181
Fitness Sharing	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	160
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	
Lexicase	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	1
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	117
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	247
Random	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	249
Random	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	247
Random	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	162
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	35
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	45
Tournament	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	115
Tournament	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	185
Tournament	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50	50	176

```

        discretize(lag(pheno_diversity, 100))),
pheno_phylo_10000 = condinformation(
    discretize(pheno_diversity),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
    discretize(lag(pheno_diversity, 1000))),
pheno_phylo_100000 = condinformation(
    discretize(pheno_diversity),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
    discretize(lag(pheno_diversity, 10000))),

phylo_pheno_10 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_diversity, 1)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
phylo_pheno_100 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_diversity, 10)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
phylo_pheno_1000 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_diversity, 100)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
phylo_pheno_10000 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_diversity, 1000)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
phylo_pheno_100000 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_diversity, 10000)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness,
))

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*\$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic", TRUE ~ "Phylogenetic\n\t->\nPhenotypic")

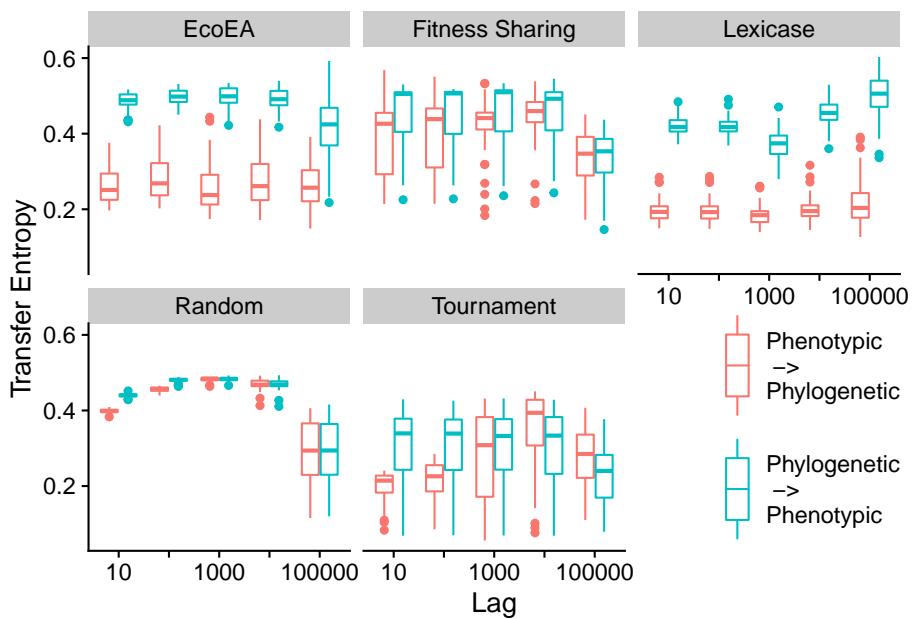
ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
)

```

```

)
) +
geom_boxplot() +
facet_wrap(~selection_name) +
scale_x_discrete("Lag", labels=c("10","", "1000","", "100000")) +
scale_y_continuous("Transfer Entropy") +
theme(legend.position = c(1, 0),
      legend.justification = c(1, 0)) +
scale_color_discrete("")

```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

selection_name	offset	.y.	group1	group2	n1
EcoEA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
EcoEA	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Fitness Sharing	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Lexicase	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Random	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	1000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	10000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50
Tournament	100000	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	50

Chapter 3

Other fitness landscapes

3.1 Setup

```
library(ggplot2)
library(tidyverse)
library(knitr)
library(cowplot)
library(viridis)
library(RColorBrewer)
library(rstatix)
library(ggsignif)
library(Hmisc)
library(kableExtra)
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce9")
library(readr)
library(stringr)
library(ggpubr)
library(infotheo)
library(osfr)
library(scales)
```

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          0.4
## year           2021
## month          02
## day            15
## svn rev        80002
## language       R
## version.string R version 4.0.4 (2021-02-15)
## nickname      Lost Library Book

# Labeled 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())

osf_retrieve_file("p79hx") %>% osf_download(conflicts = "skip") # Download data from osf

## # A tibble: 1 x 4
##   name              id      local_path      meta
##   <chr>             <chr>   <chr>        <list>
## 1 complex_fitness_lands~ 612fe4d84e5ee501~ ./complex_fitness_lands~ <named list~
data_loc <- "complex_fitness_landscapes.csv"

data <- read_csv(data_loc, na=c("NONE", "NA", ""))

data <- data %>%
  filter(N==20, generation %% 10 == 0) %>%
  mutate(
    selection_name = as.factor(case_when(
      SELECTION == 0 ~ "Tournament",
      SELECTION == 1 ~ "Fitness sharing",
      SELECTION == 2 ~ "Lexicase",
      SELECTION == 3 ~ "Eco-EA",
      SELECTION == 4 ~ "GA"
    )))

```

```

    SELECTION == 4 ~ "Random",
  )),
problem_name = as.factor(case_when(
  PROBLEM == 0 ~ "NK Landscape",
  PROBLEM == 1 ~ "Count Odds",
  PROBLEM == 2 ~ "Real-valued optimization",
  PROBLEM == 3 ~ "Sorting network",
  PROBLEM == 4 ~ "Logic-9"
))
)

data <- filter(data, generation <= 2000)
final_data <- filter(data, generation==max(data$generation))

```

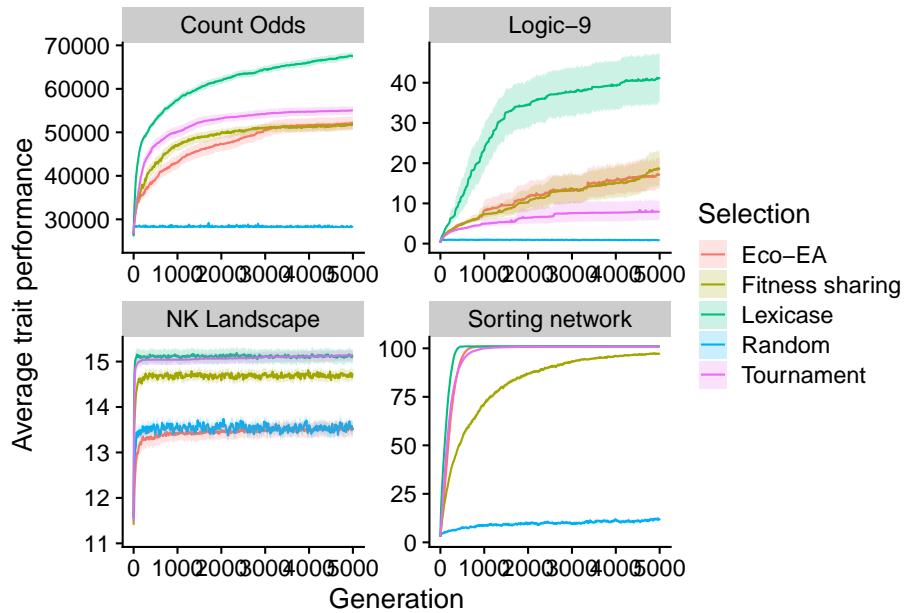
3.2 Performance

3.2.1 Over time

```

ggplot(
  data,
  aes(
    x=generation,
    y=max_performance,
    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"
) +
  scale_x_continuous(
    name="Generation"
) +
  scale_color_discrete("Selection") +
  scale_fill_discrete("Selection") +
  facet_wrap(~problem_name, scales="free")

```



3.2.2 Final

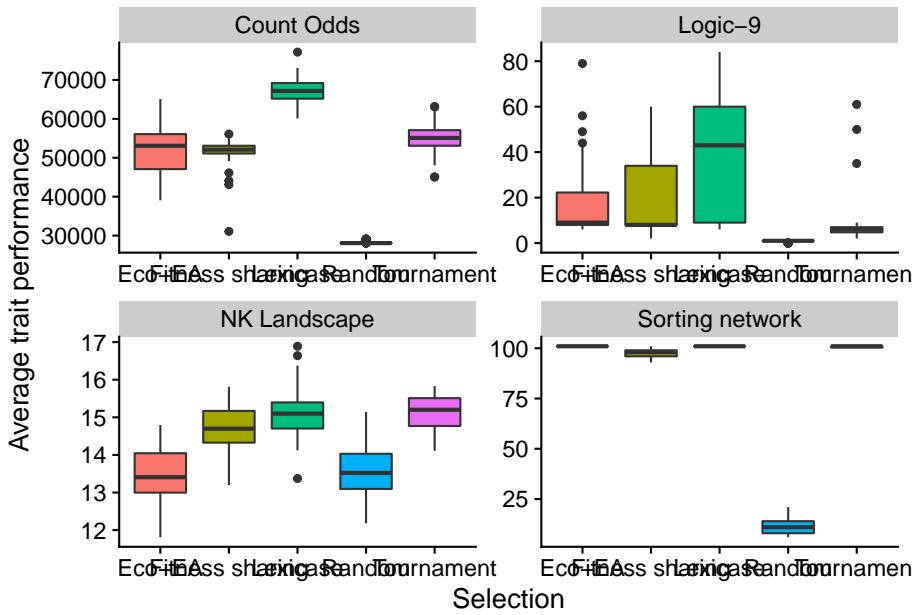
```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(max_performance ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name",step.increase=1)
#stat.test$manual_position <- stat.test$y.position * .5
#stat.test$manual_position <- c(110, 150, 170, 170, 130, 110)
stat.test$label <- mapply(p_label,stat.test$p.adj)

ggplot(
  final_data,
  aes(
    x=selection_name,
    y=max_performance,
    fill=selection_name
  )
) +
  geom_boxplot() +
  scale_y_continuous(
    name="Average trait performance"
) +
```

```

scale_x_discrete(
  name="Selection"
) +
scale_fill_discrete(
  name="Selection"
) +
scale_color_discrete(
  name="Selection"
) +
theme(legend.position="none") +
facet_wrap(~problem_name, scales="free")

```



```

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")

```

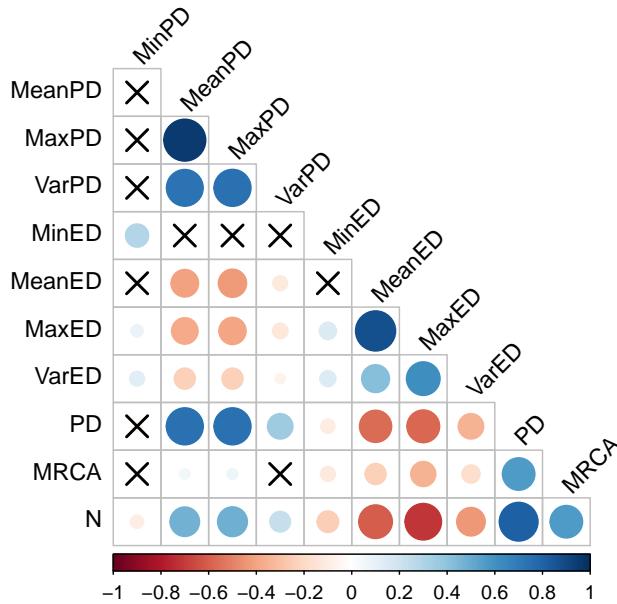
.y.	group1	group2	n1	n2	statistic	p	p.adj
max_performance	Eco-EA	Fitness sharing	240	240	29310.5	7.37e-01	1.000000
max_performance	Eco-EA	Lexicase	240	240	23075.0	1.46e-04	0.001460
max_performance	Eco-EA	Random	240	240	39696.5	0.00e+00	0.000000
max_performance	Eco-EA	Tournament	240	240	29346.0	7.18e-01	1.000000
max_performance	Fitness sharing	Lexicase	240	240	22328.5	2.01e-05	0.000201
max_performance	Fitness sharing	Random	240	240	41293.0	0.00e+00	0.000000
max_performance	Fitness sharing	Tournament	240	240	27500.5	3.92e-01	1.000000
max_performance	Lexicase	Random	240	240	44132.5	0.00e+00	0.000000
max_performance	Lexicase	Tournament	240	240	34831.5	6.61e-05	0.000661
max_performance	Random	Tournament	240	240	18254.0	0.00e+00	0.000000

3.3 Phylogenetic diversity

3.3.1 Relationship between different types of phylogenetic diversity

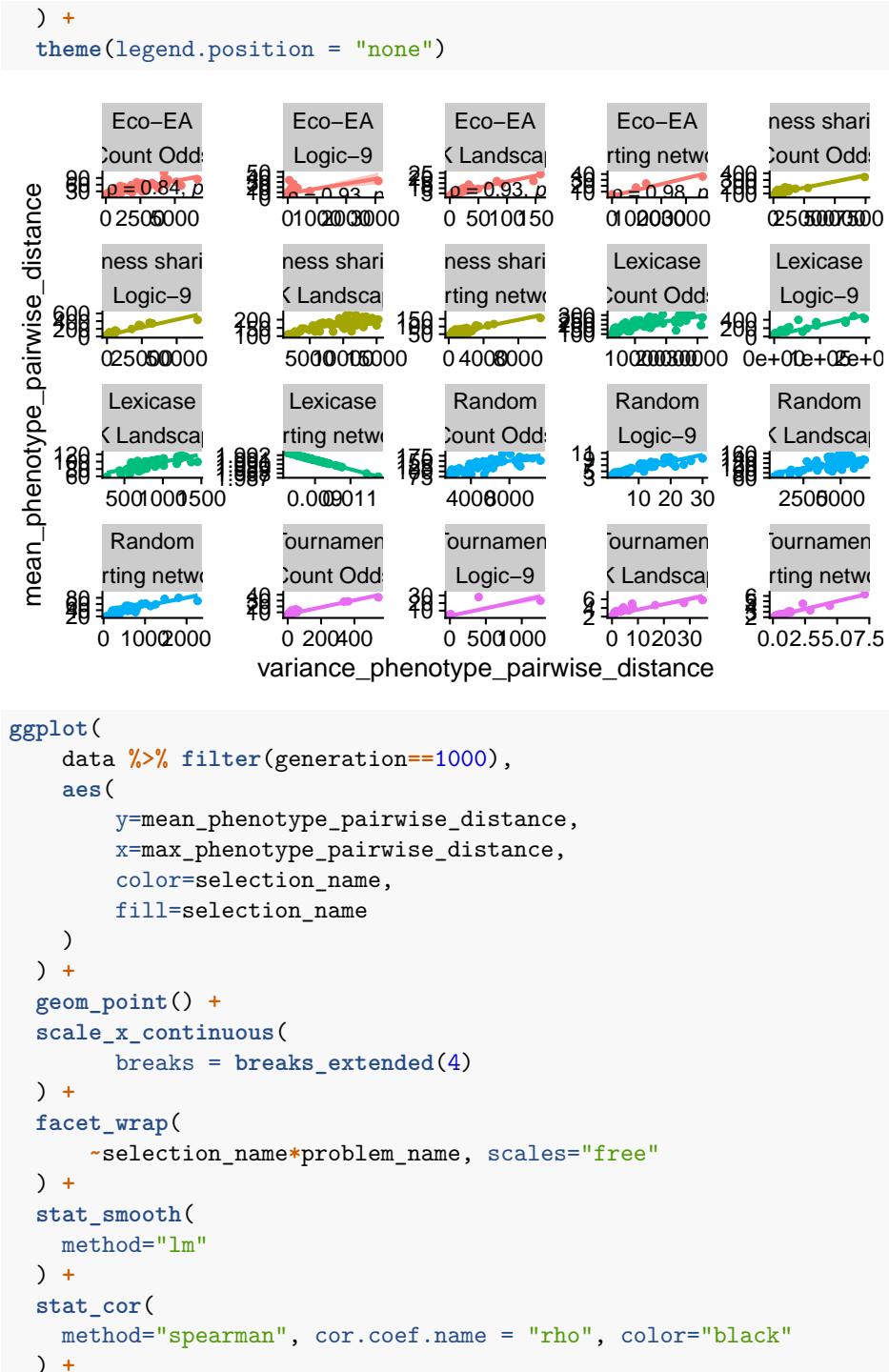
First, to get a big-picture overview, we make correlation matrices of all the different phylogenetic diversity metrics:

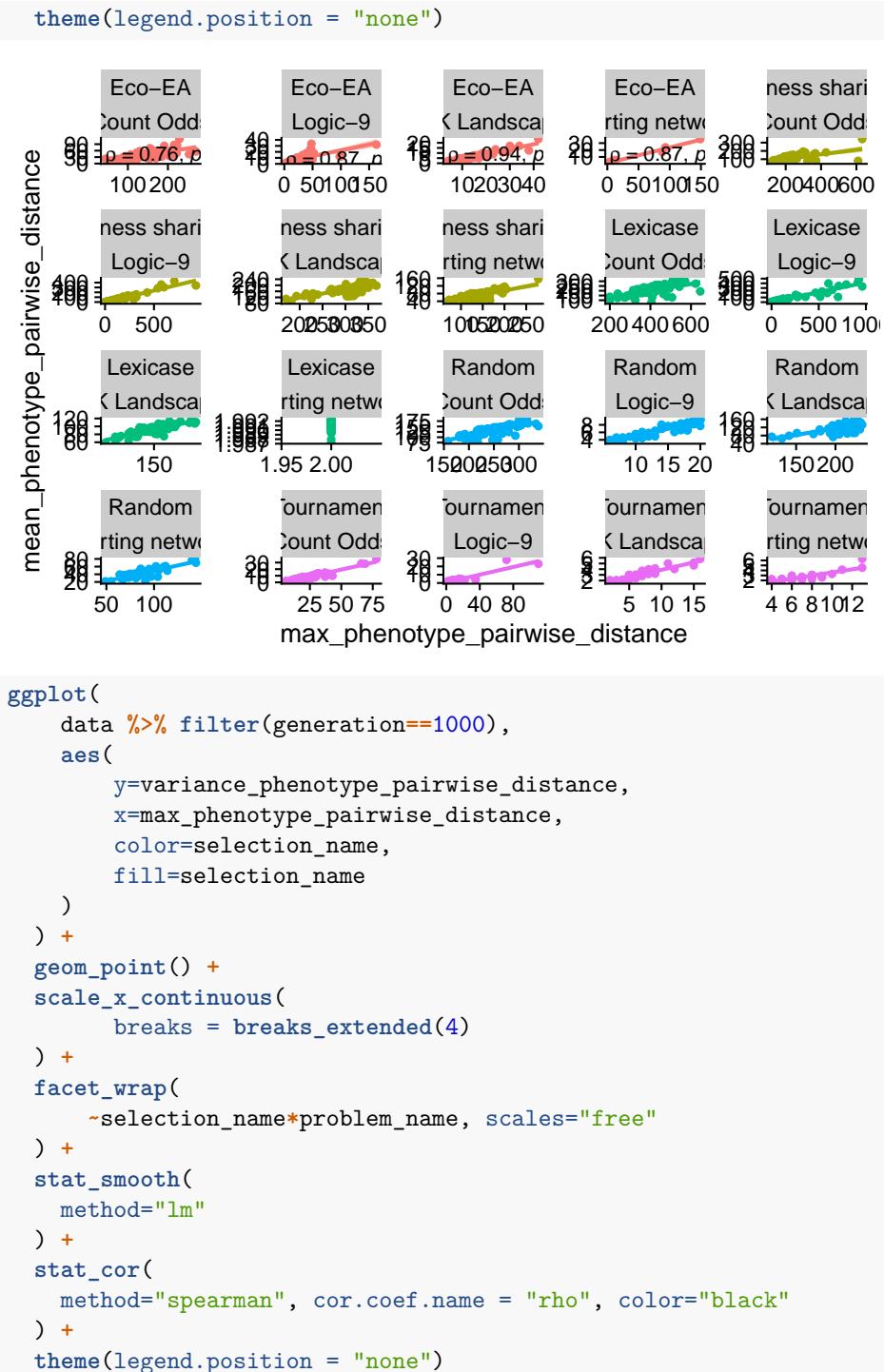
```
final_data %>%
  transmute(MinPD=min_phenotype_pairwise_distance,
            MeanPD=mean_phenotype_pairwise_distance,
            MaxPD=max_phenotype_pairwise_distance,
            VarPD=variance_phenotype_pairwise_distance,
            MinED = min_phenotype_evolutionary_distinctiveness,
            MeanED= mean_phenotype_evolutionary_distinctiveness,
            MaxED=max_phenotype_evolutionary_distinctiveness,
            VarED=variance_phenotype_evolutionary_distinctiveness,
            PD=phenotype_current_phylogenetic_diversity, # See Faith 1992
            MRCA=phenotype_mrca_depth, # Phylogenetic depth of most recent common ancestor
            N=phenotype_num_taxa      # Number of taxonomically-distinct phenotypes
  ) %>%
  cor_mat() %>%
  pull_lower_triangle() %>%
  cor_plot()
```

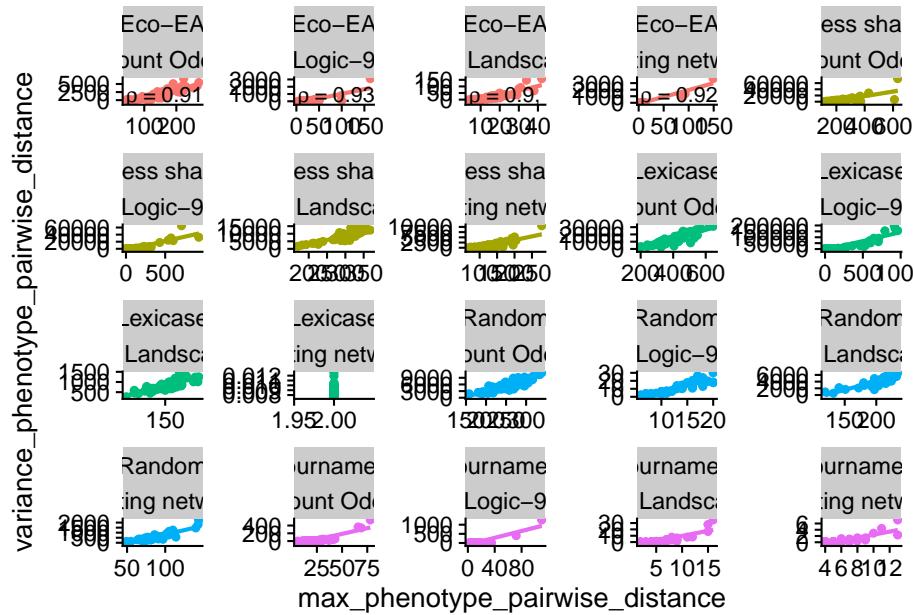


However, these correlations may well vary by selection scheme and by problem, and even over time within a selection scheme and problem. Let's take a look at some scatter plots.

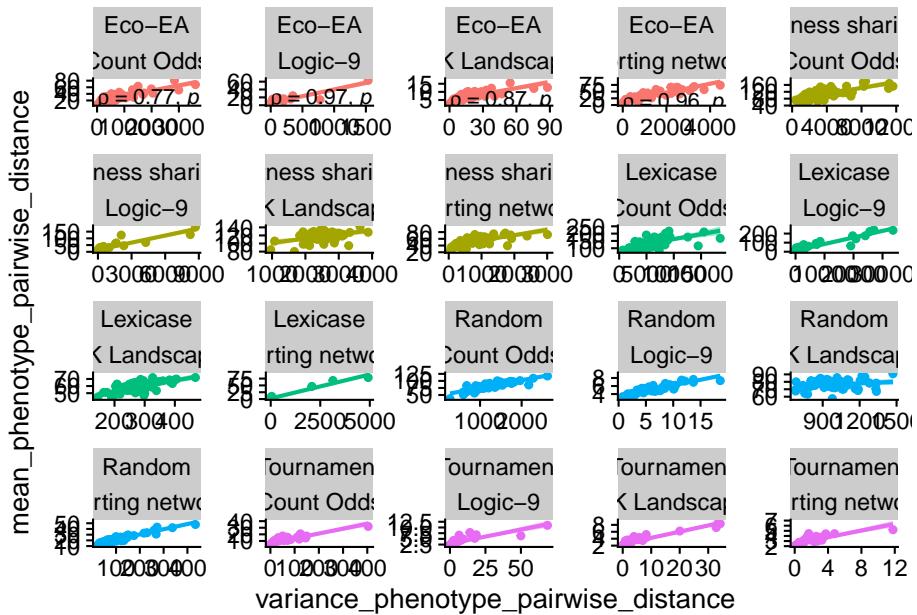
```
ggplot(
  data %>% filter(generation==1000),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=variance_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
```



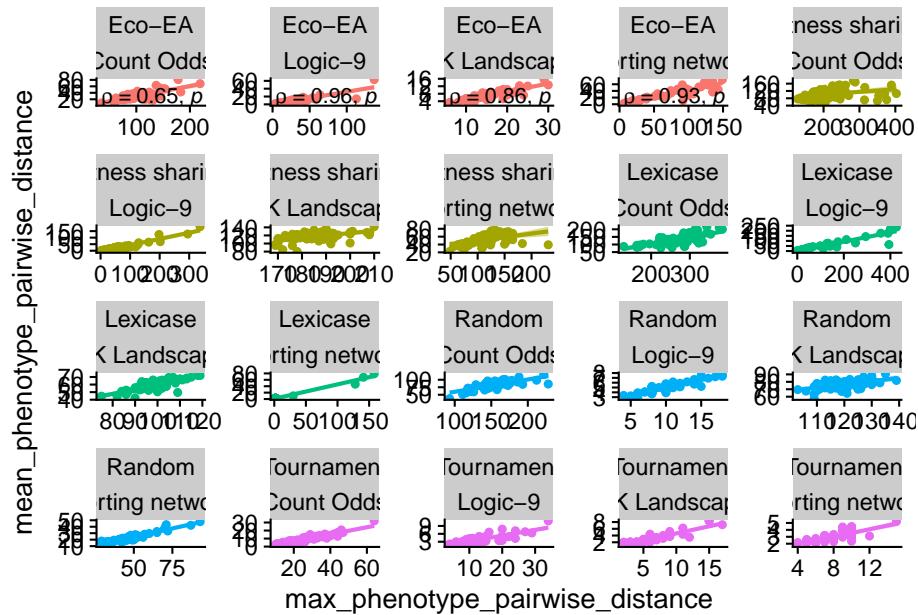




```
ggplot(
  data %>% filter(generation==500),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=variance_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



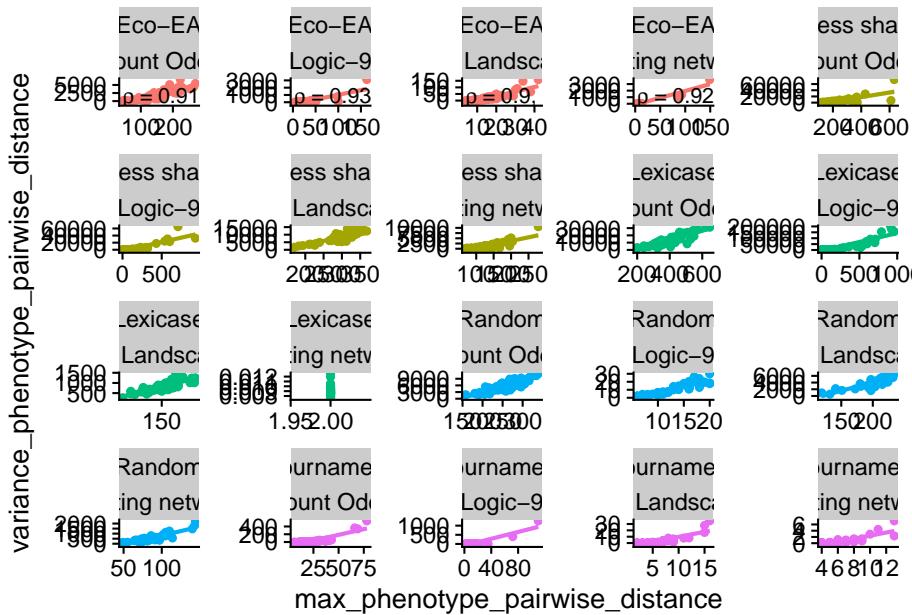
```
ggplot(
  data %>% filter(generation==500),
  aes(
    y=mean_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



```

ggplot(
  data %>% filter(generation==1000),
  aes(
    y=variance_phenotype_pairwise_distance,
    x=max_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")

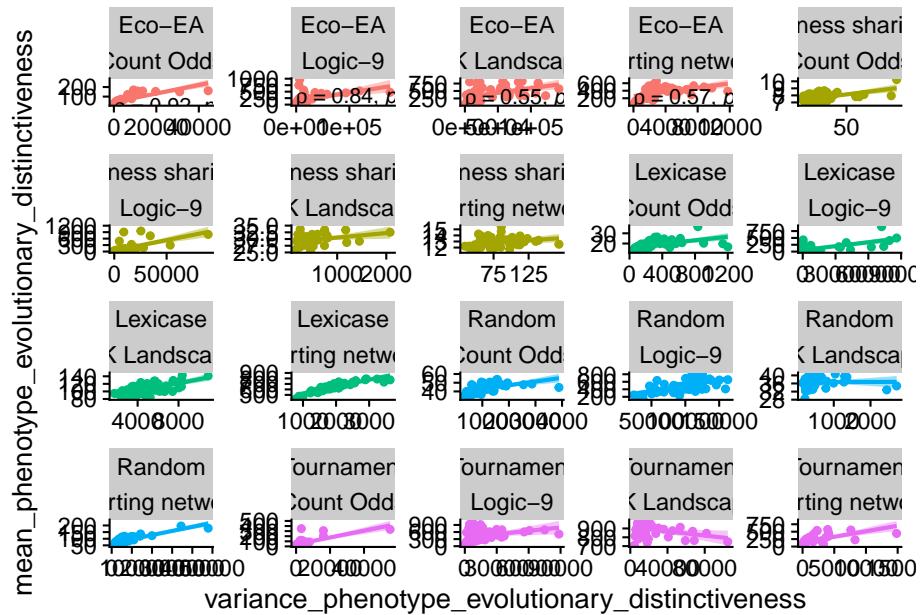
```



```

ggplot(
  data %>% filter(generation==1000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=variance_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")

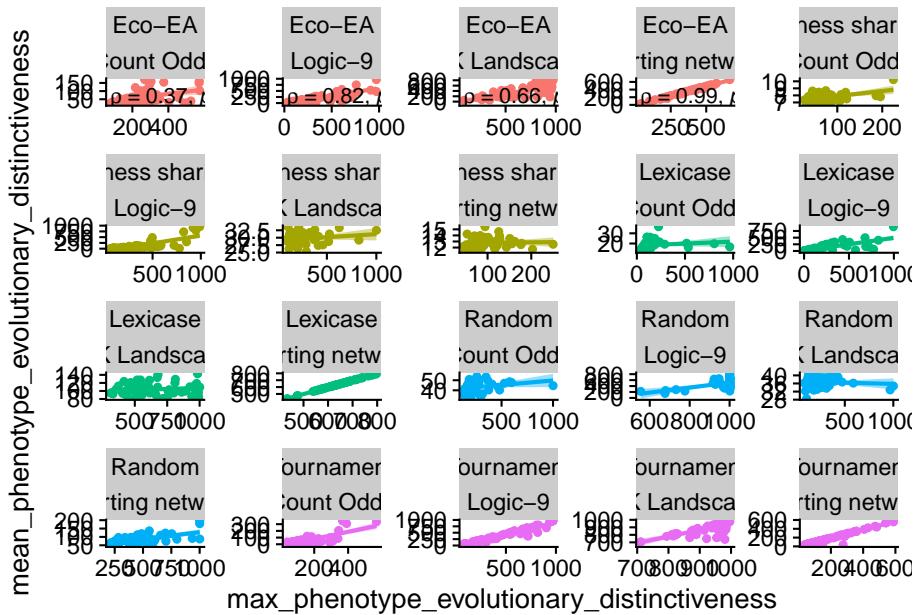
```



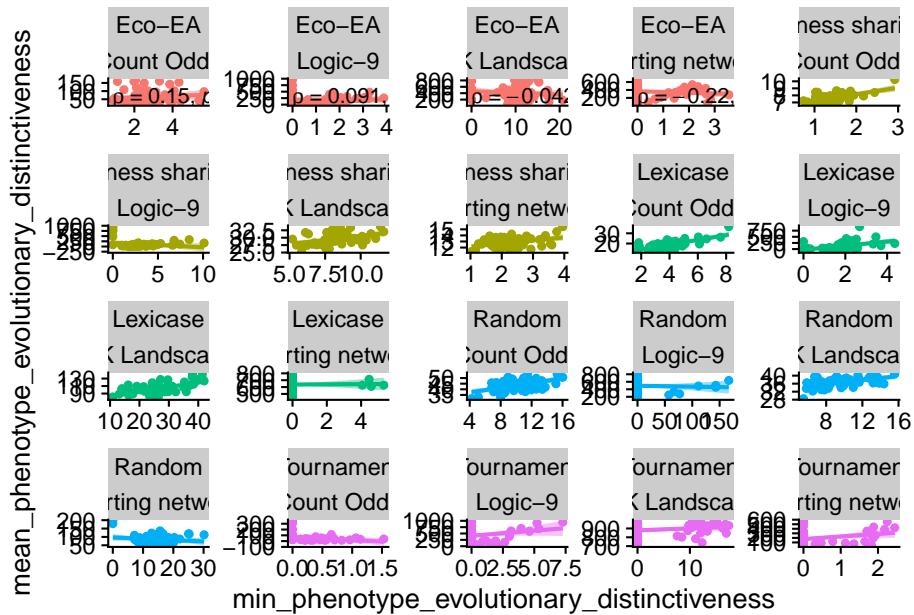
```

ggplot(
  data %>% filter(generation==1000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=max_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")

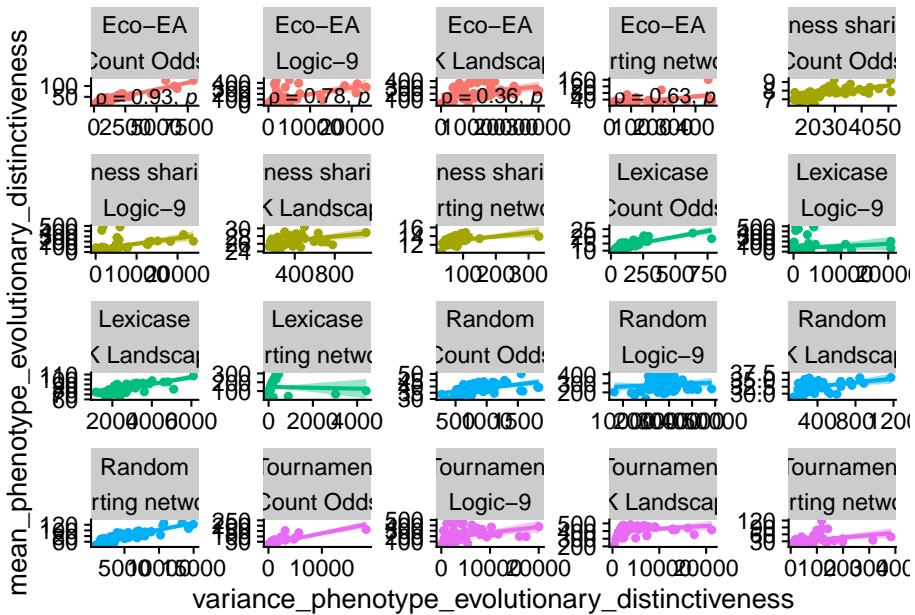
```



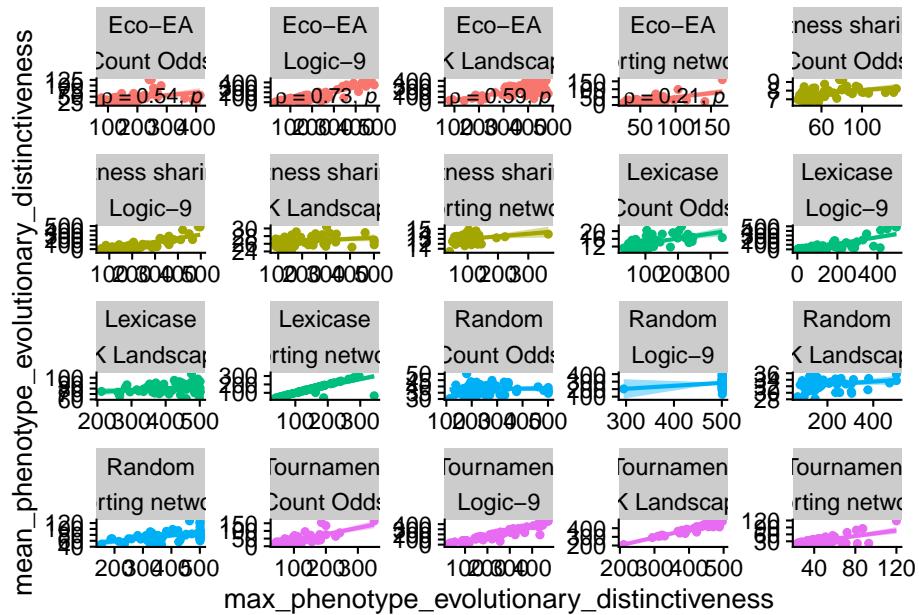
```
ggplot(
  data %>% filter(generation==1000),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=min_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



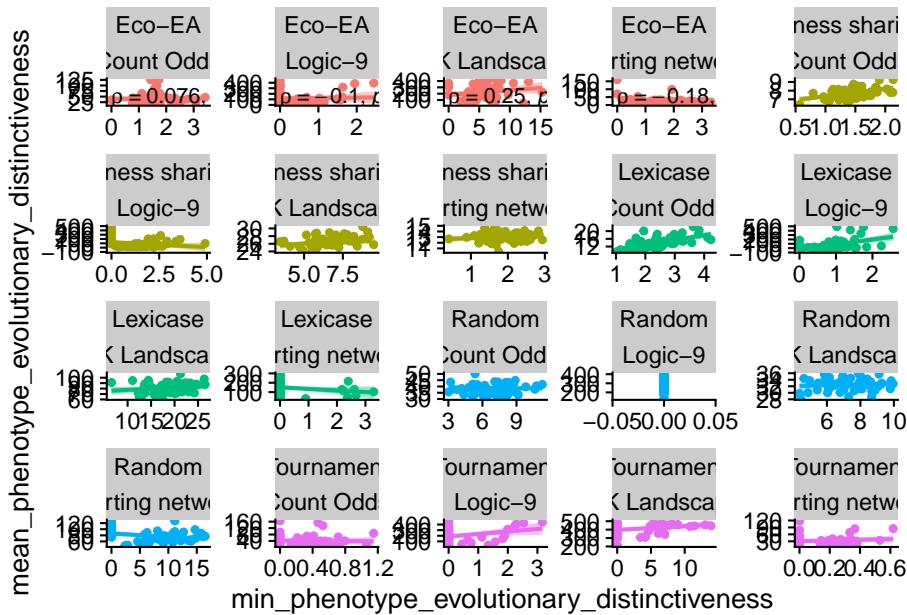
```
ggplot(
  data %>% filter(generation==500),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=variance_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



```
ggplot(
  data %>% filter(generation==500),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=max_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")
```



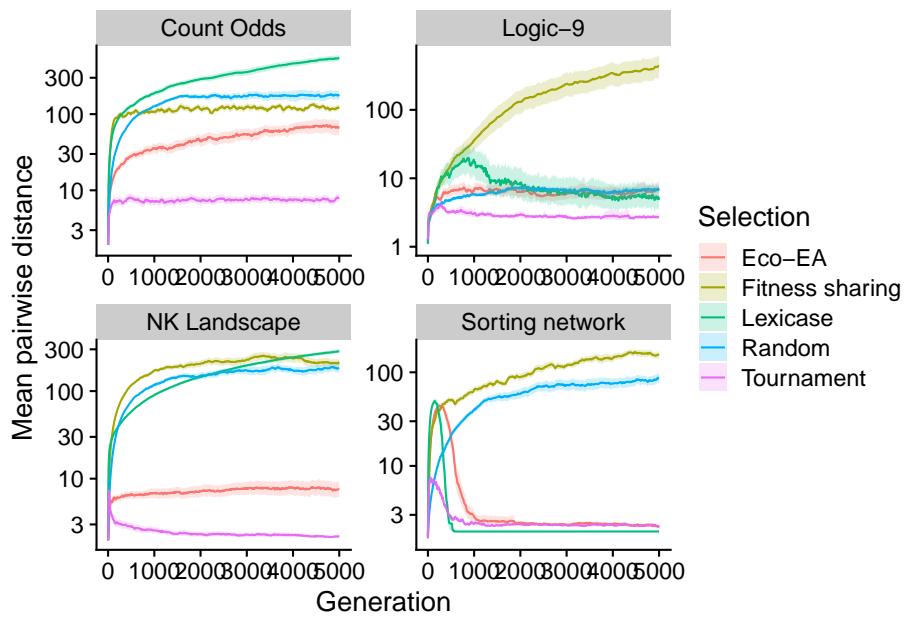
```
ggplot(
  data %>% filter(generation==500),
  aes(
    y=mean_phenotype_evolutionary_distinctiveness,
    x=min_phenotype_evolutionary_distinctiveness,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_x_continuous(
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
  theme(legend.position = "none")
```



3.3.2 Over time

```
ggplot(
  data,
  aes(
    x=generation,
    y=mean_phenotype_pairwise_distance,
    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_log10(
    name="Mean pairwise distance"
  ) +
  scale_x_continuous(
    name="Generation"
  )
```

```
) +
  scale_color_discrete("Selection") +
  scale_fill_discrete("Selection") +
  facet_wrap(~problem_name, scales = "free")
```



3.3.3 Final

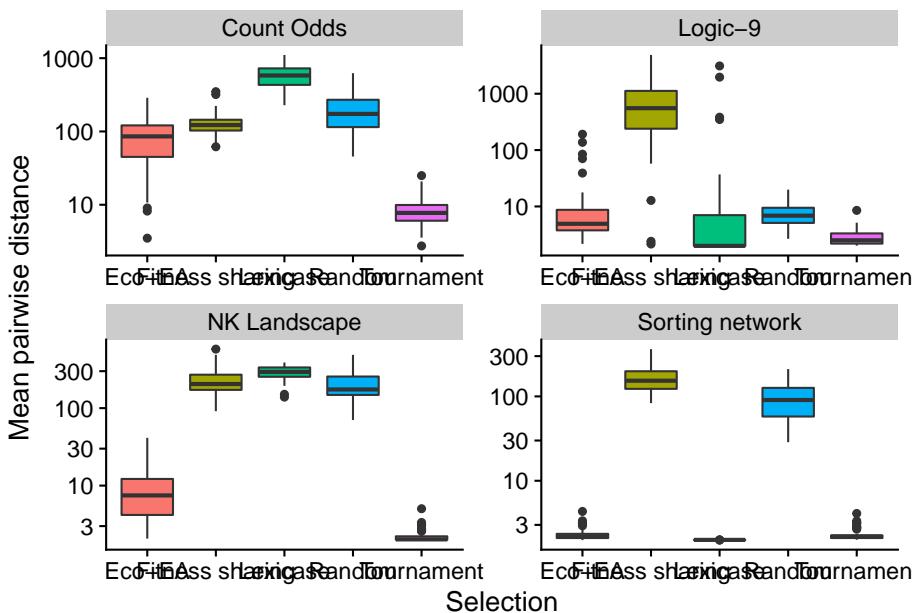
```
# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(mean_phenotype_pairwise_distance ~ selection_name) %>%
  adjust_pvalue(method = "bonferroni") %>%
  add_significance() %>%
  add_xy_position(x="selection_name", step.increase=1)
#stat.test$manual_position <- stat.test$y.position * .5
#stat.test$manual_position <- c(110, 150, 170, 170, 130, 110)
stat.test$label <- mapply(p_label, stat.test$p.adj)

ggplot(
  final_data,
  aes(
    x=selection_name,
    y=mean_phenotype_pairwise_distance,
    fill=selection_name
  )
```

```

) +
  geom_boxplot() +
  scale_y_log10(
    name="Mean pairwise distance"
) +
  scale_x_discrete(
    name="Selection"
) +
  scale_fill_discrete(
    name="Selection"
) +
  scale_color_discrete(
    name="Selection"
) +
  theme(legend.position = "none") +
  facet_wrap(~problem_name, scales = "free")

```



```

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"

```

.y.	group1	group2	n1	n2	statistic	p.value
mean_phenotype_pairwise_distance	Eco-EA	Fitness sharing	240	240	2831.0	0.000
mean_phenotype_pairwise_distance	Eco-EA	Lexicase	240	240	25159.0	1.7e-10
mean_phenotype_pairwise_distance	Eco-EA	Random	240	240	9552.0	0.000
mean_phenotype_pairwise_distance	Eco-EA	Tournament	240	240	43052.5	0.000
mean_phenotype_pairwise_distance	Fitness sharing	Lexicase	240	240	33683.0	1.0e-10
mean_phenotype_pairwise_distance	Fitness sharing	Random	240	240	40738.0	0.000
mean_phenotype_pairwise_distance	Fitness sharing	Tournament	240	240	57314.0	0.000
mean_phenotype_pairwise_distance	Lexicase	Random	240	240	28272.0	8.4e-10
mean_phenotype_pairwise_distance	Lexicase	Tournament	240	240	34801.0	7.3e-10
mean_phenotype_pairwise_distance	Random	Tournament	240	240	54927.0	0.000

```
)
) %>%
scroll_box(width="600px")
```

3.4 Phenotypic diversity

3.4.1 Relationship between different types of phenotypic diversity

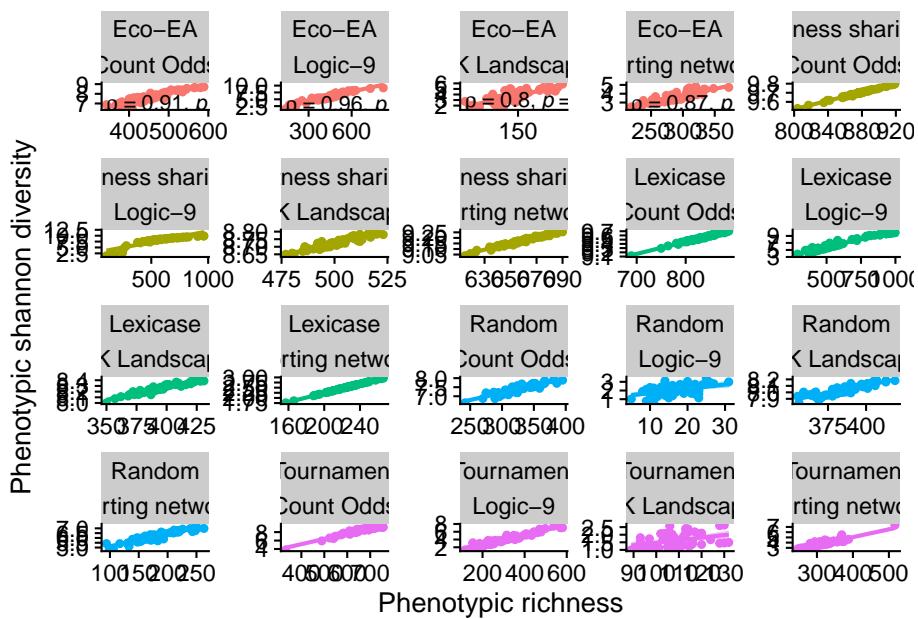
First, we should assess the extent to which different metrics of phenotypic diversity are capturing different information.

```
ggplot(
  data %>% filter(generation==1000),
  aes(
    y=phenotype_diversity,
    x=phenotype_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Phenotypic shannon diversity"
  ) +
  scale_x_continuous(
    name="Phenotypic richness",
    breaks = breaks_extended(4)
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
```

```

stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



3.4.2 Over time

```

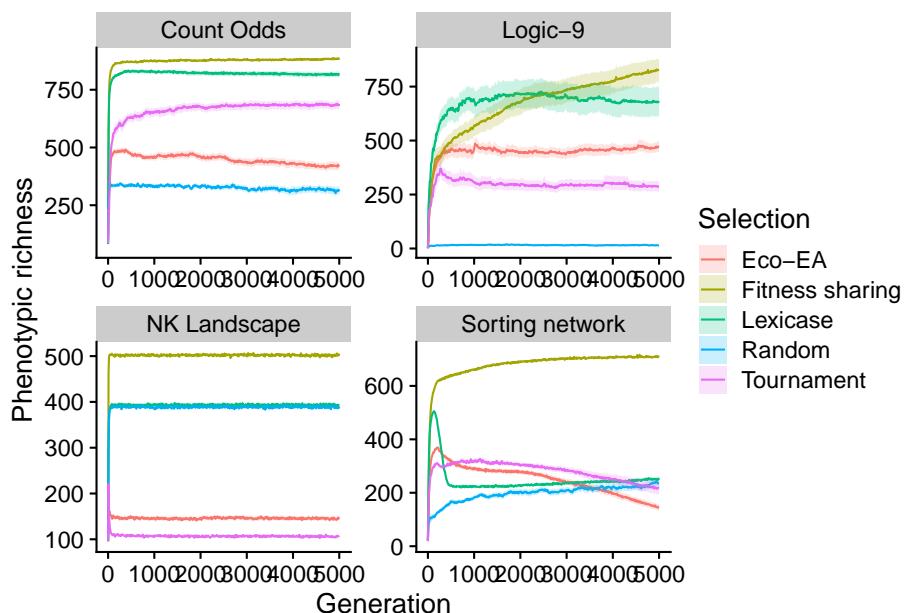
ggplot(
  data,
  aes(
    x=generation,
    y=phenotype_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  stat_summary(geom="line", fun=mean) +
  stat_summary(
    geom="ribbon",
    fun.data="mean_cl_boot",
    fun.args=list(conf.int=0.95),

```

```

    alpha=0.2,
    linetype=0
) +
scale_y_continuous(
  name="Phenotypic richness"
) +
scale_x_continuous(
  name="Generation"
) +
scale_color_discrete("Selection") +
scale_fill_discrete("Selection") +
facet_wrap(~problem_name, scales = "free")

```



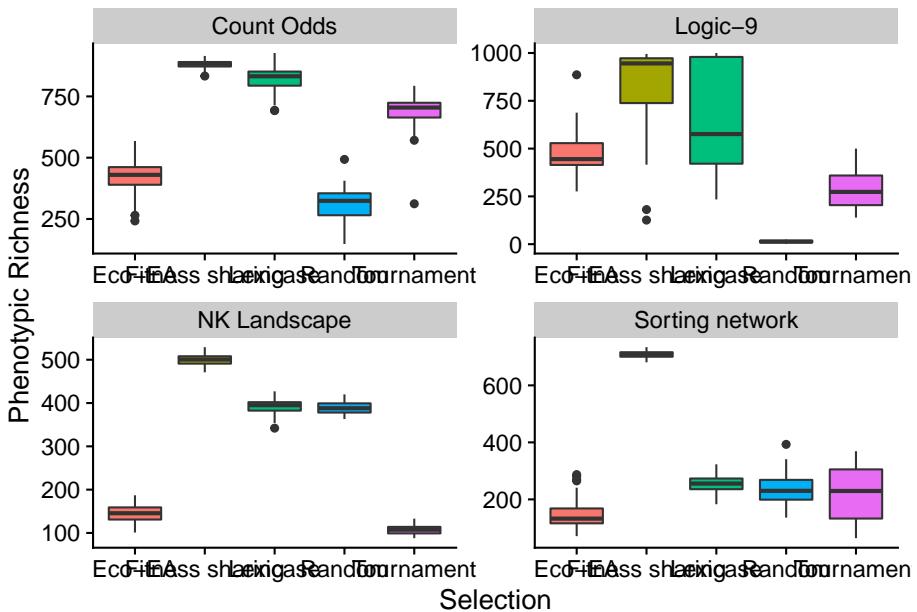
3.4.3 Final

```

# Compute manual labels for geom_signif
stat.test <- final_data %>%
  wilcox_test(phenotype_num_taxa ~ 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 * .5
#stat.test$manual_position <- c(110, 150, 170, 170, 130, 110)
stat.test$label <- mapply(p_label, stat.test$p.adj)

```

```
ggplot(
  final_data,
  aes(
    x=selection_name,
    y=phenotype_num_taxa,
    fill=selection_name
  )
) +
  geom_boxplot() +
  scale_y_continuous(
    name="Phenotypic Richness"
  ) +
  scale_x_discrete(
    name="Selection"
  ) +
  scale_fill_discrete(
    name="Selection"
  ) +
  scale_color_discrete(
    name="Selection"
  ) +
  theme(legend.position = "none") +
  facet_wrap(~problem_name, scales = "free")
```



.y.	group1	group2	n1	n2	statistic	p	p.adj
phenotype_num_taxa	Eco-EA	Fitness sharing	240	240	2406.0	0.000000	0.000000
phenotype_num_taxa	Eco-EA	Lexicase	240	240	14833.5	0.000000	0.000000
phenotype_num_taxa	Eco-EA	Random	240	240	34261.5	0.000326	0.003260
phenotype_num_taxa	Eco-EA	Tournament	240	240	29587.5	0.604000	1.000000
phenotype_num_taxa	Fitness sharing	Lexicase	240	240	42195.5	0.000000	0.000000
phenotype_num_taxa	Fitness sharing	Random	240	240	57231.0	0.000000	0.000000
phenotype_num_taxa	Fitness sharing	Tournament	240	240	51342.0	0.000000	0.000000
phenotype_num_taxa	Lexicase	Random	240	240	47218.0	0.000000	0.000000
phenotype_num_taxa	Lexicase	Tournament	240	240	43850.0	0.000000	0.000000
phenotype_num_taxa	Random	Tournament	240	240	25692.5	0.041000	0.410000

```
stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box(width="600px")
```

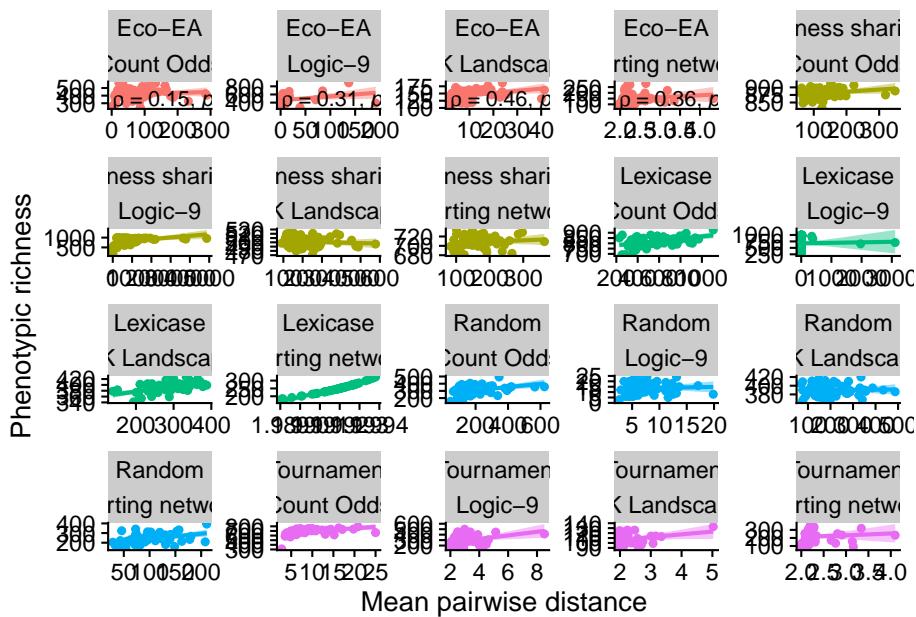
3.5 Relationship between phenotypic and phylogenetic diversity

```
ggplot(
  final_data,
  aes(
    y=phenotype_num_taxa,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Phenotypic richness"
  ) +
  scale_x_continuous(
    name="Mean pairwise distance"
```

```

) +
  facet_wrap(
    ~selection_name*problem_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



3.6 Relationship between diversity and success

3.6.1 Earlier in run

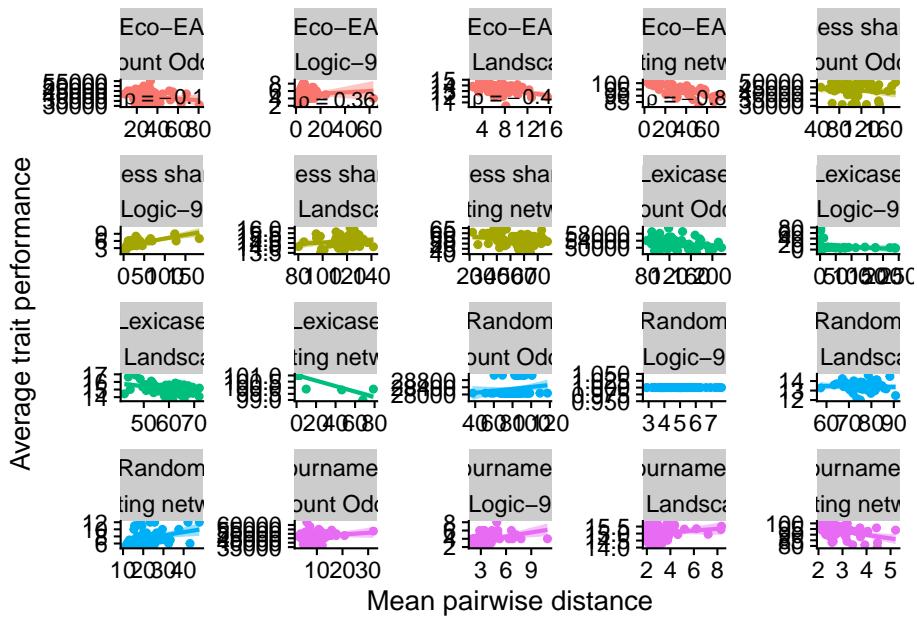
```

ggplot(
  data %>% filter(generation==500),
  aes(
    y=max_performance,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
)
```

```

)
) +
geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
scale_x_continuous(
  name="Mean pairwise distance"
) +
facet_wrap(
  ~selection_name*problem_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



```

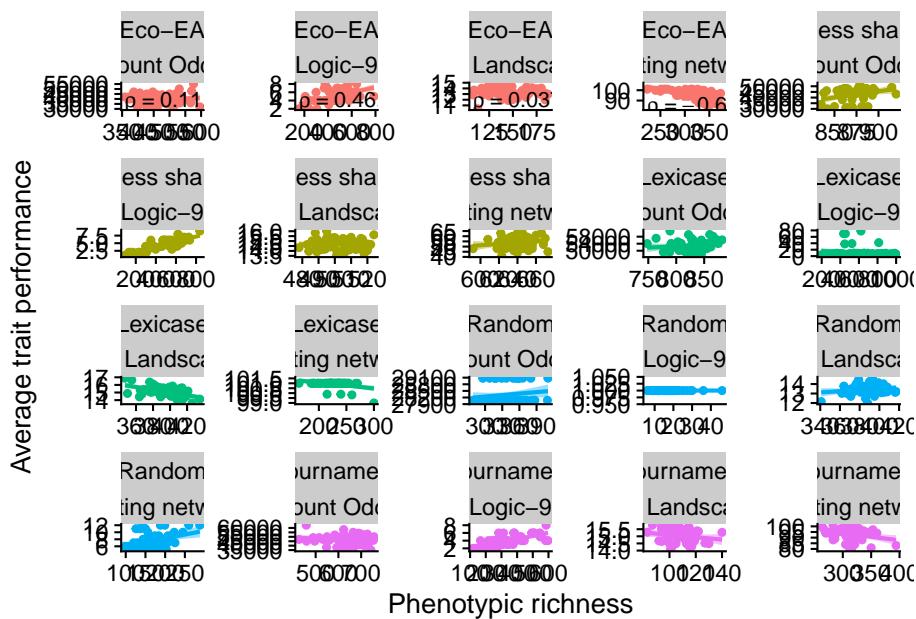
ggplot(
  data %>% filter(generation==500),
  aes(
    y=max_performance,
    x=phenotype_num_taxa,

```

```

        color=selection_name,
        fill=selection_name
    )
) +
geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Phenotypic richness"
) +
  facet_wrap(
    ~selection_name*problem_name, scales="free"
) +
  stat_smooth(
    method="lm"
) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
) +
  theme(legend.position = "none")

```



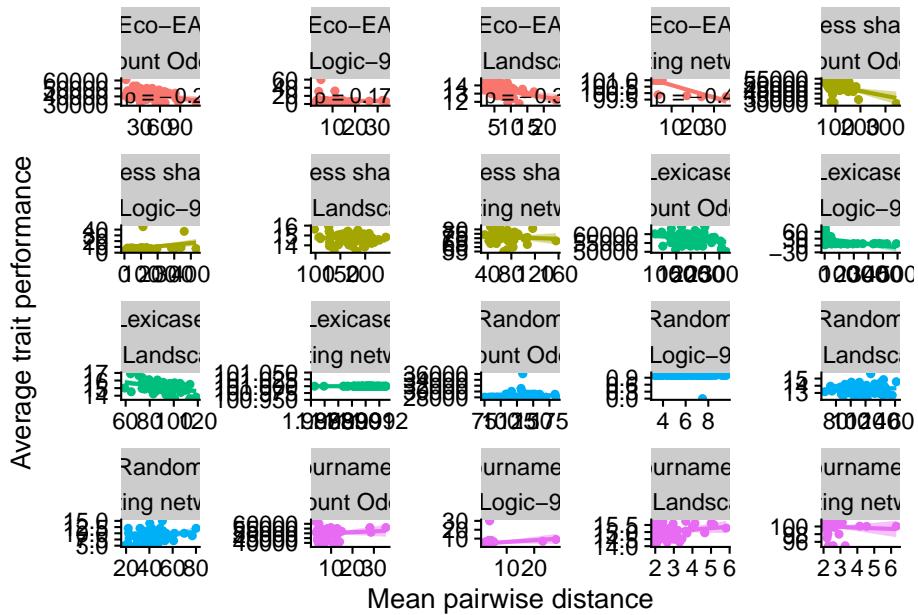
```

phylogney_vs_performance <- ggplot(
  data %>% filter(generation==1000),
  aes(

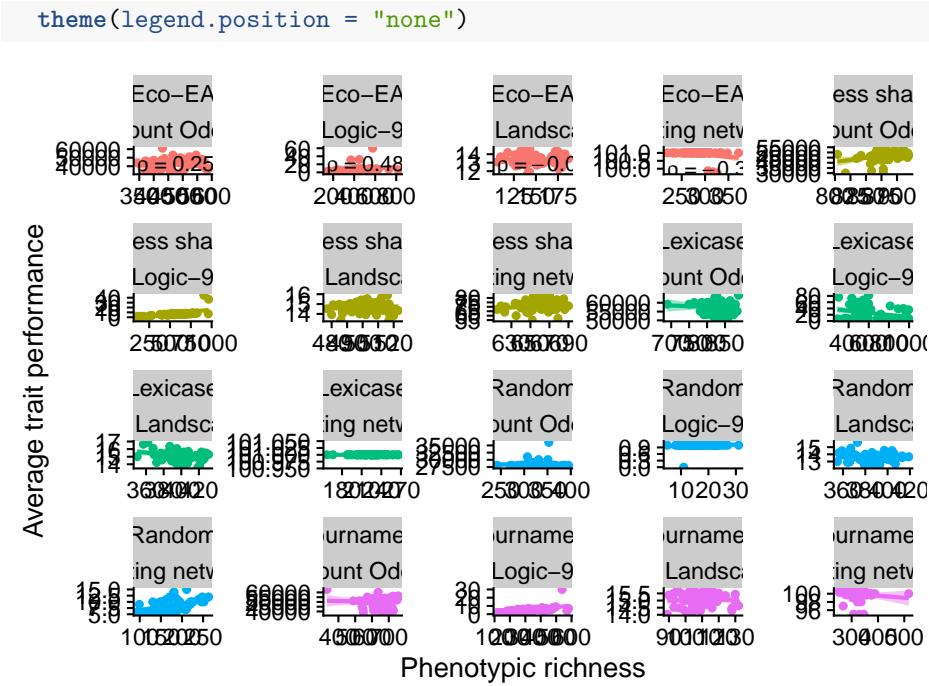
```

```
    y=max_performance,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
geom_point() +
scale_y_continuous(
  name="Average trait performance"
) +
scale_x_continuous(
  name="Mean pairwise distance"
) +
facet_wrap(
  ~selection_name*problem_name, scales="free"
) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

phylogney_vs_performance
```



```
ggplot(
  data %>% filter(generation==1000),
  aes(
    y=max_performance,
    x=phenotype_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
  ) +
  scale_x_continuous(
    name="Phenotypic richness"
  ) +
  facet_wrap(~selection_name*problem_name, scales="free")
) +
  stat_smooth(
    method="lm"
  ) +
  stat_cor(
    method="spearman", cor.coef.name = "rho", color="black"
  ) +
```



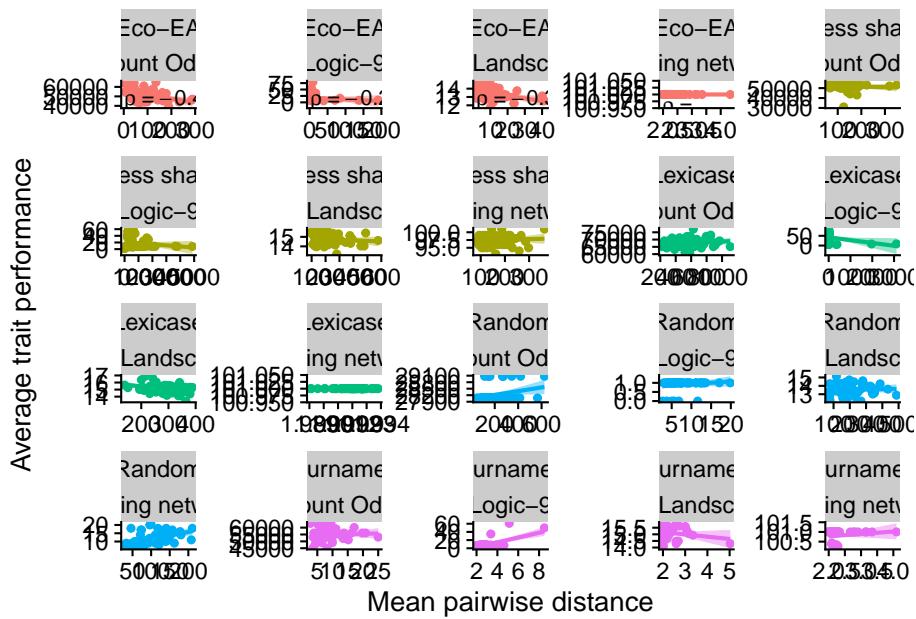
3.6.2 End of run

```
ggplot(
  final_data,
  aes(
    y=max_performance,
    x=mean_phenotype_pairwise_distance,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Mean pairwise distance"
) +
  facet_wrap(
    ~selection_name*problem_name, scales="free"
) +
  stat_smooth()
```

```

    method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



```

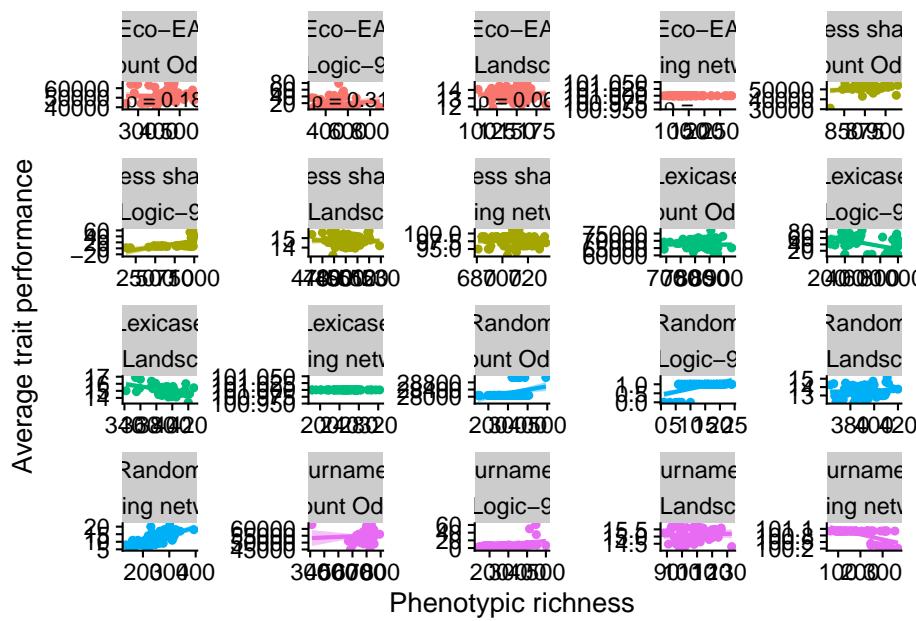
ggplot(
  final_data,
  aes(
    y=max_performance,
    x=phenotype_num_taxa,
    color=selection_name,
    fill=selection_name
  )
) +
  geom_point() +
  scale_y_continuous(
    name="Average trait performance"
) +
  scale_x_continuous(
    name="Phenotypic richness"
) +
  facet_wrap(
    ~selection_name*problem_name, scales="free"
)

```

```

) +
stat_smooth(
  method="lm"
) +
stat_cor(
  method="spearman", cor.coef.name = "rho", color="black"
) +
theme(legend.position = "none")

```



3.7 Causality analysis

3.7.1 Setup

First let's define a function we'll use to calculate and output significance and effect size for these results:

```

transfer_entropy_stats <- function(res) {
  stat.test <- res %>%
    group_by(selection_name, offset) %>%
    wilcox_test(value ~ Type) %>%
    adjust_pvalue(method = "bonferroni") %>%
    add_significance()
  stat.test$label <- mapply(p_label, stat.test$p.adj)
}

```

```

# Calculate effect sizes for these differences
effect_sizes <- res %>%
  group_by(selection_name, offset) %>%
  wilcox_effsize(value ~ Type)

stat.test$effsize <- effect_sizes$effsize
stat.test$magnitude <- effect_sizes$magnitude

stat.test %>%
  kbl() %>%
  kable_styling(
    bootstrap_options = c(
      "striped",
      "hover",
      "condensed",
      "responsive"
    )
  ) %>%
  scroll_box()
}

```

3.7.2 Transfer entropy from diversity to fitness

3.7.2.1 Max pairwise distance vs. phenotypic richness

```

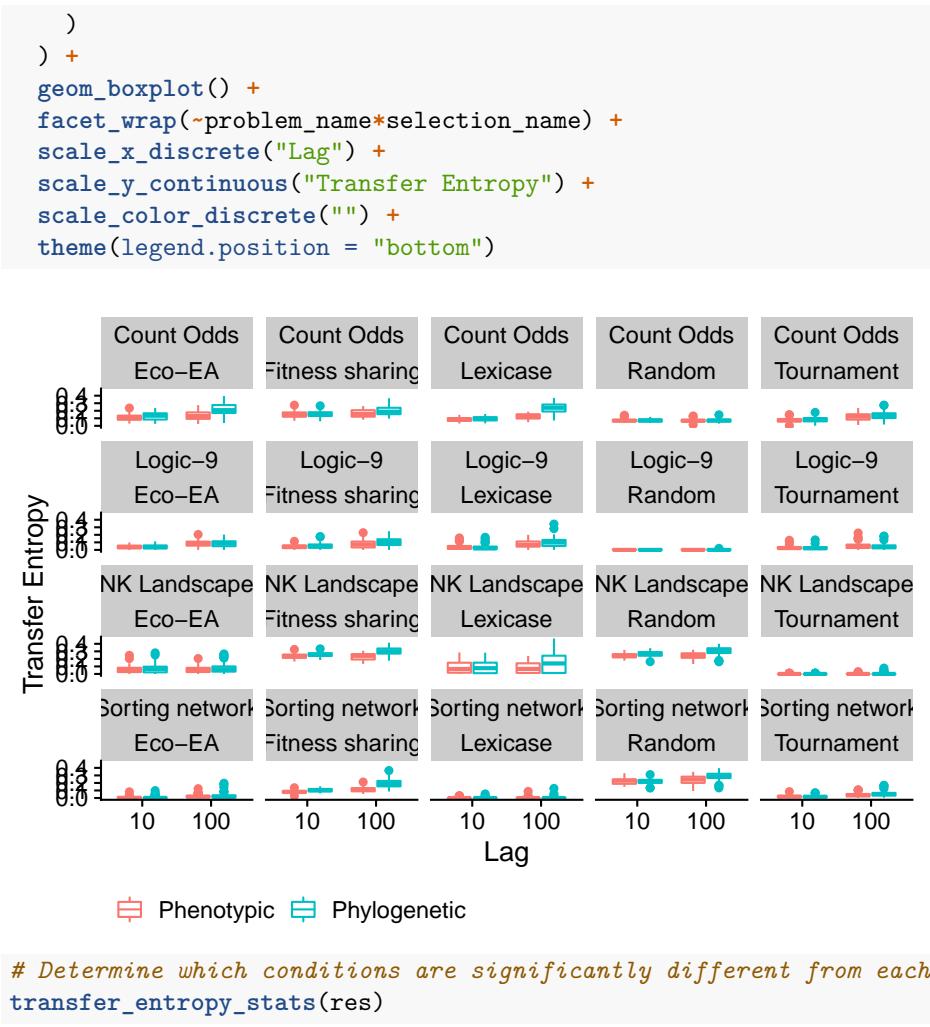
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(max_phenotype_pairwise)))
  fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(max_phenotype_pairwise)))
  fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(max_phenotype_pairwise)))
  fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
  fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
  fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
)

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*$")
res>Type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
)

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	27721.5	0.478000	1.00
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	25260.0	0.019800	0.19
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	25411.0	0.025700	0.29
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	18873.0	0.000000	0.00
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	28721.5	0.959000	1.00
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	22943.0	0.000103	0.00
Random	10	value	Phenotypic	Phylogenetic	240	240	27780.5	0.502000	1.00
Random	100	value	Phenotypic	Phylogenetic	240	240	24742.0	0.007480	0.07
Tournament	10	value	Phenotypic	Phylogenetic	240	240	29185.0	0.799000	1.00
Tournament	100	value	Phenotypic	Phylogenetic	240	240	27464.0	0.377000	1.00



3.7.2.2 Mean pairwise distance vs. phenotypic richness

```

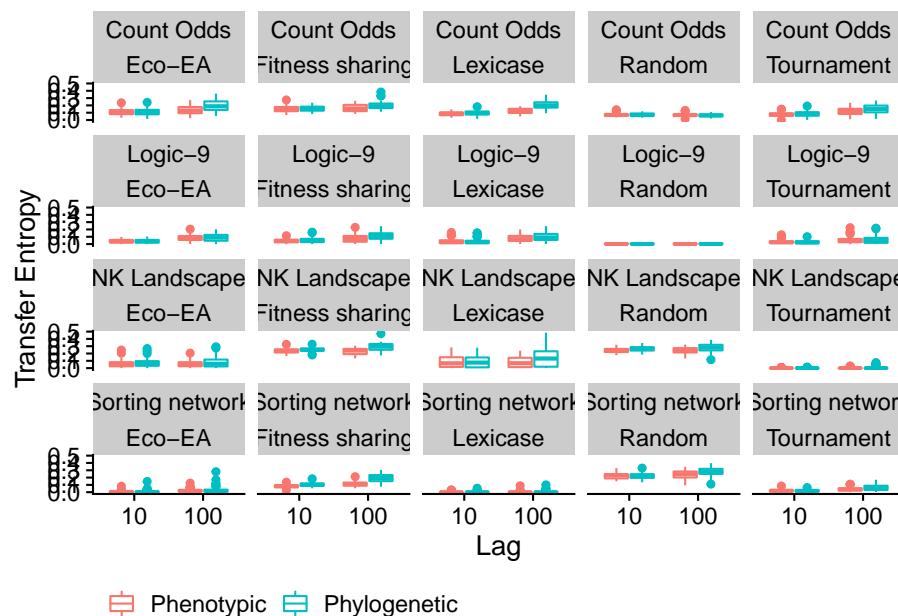
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_pairwise), 1)),
  fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_pairwise), 10)),
  fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_pairwise), 50)),
  fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa), 1)),
  fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa), 10)),
  fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa), 50))
)

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*\$")
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~problem_name*selection_name) +
  scale_x_discrete("Lag") +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("") +
  theme(legend.position = "bottom")

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	27824.0	0.521000	1.00
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	25774.5	0.046500	0.46
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	25570.0	0.033600	0.33
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	18806.0	0.000000	0.00
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	28316.5	0.749000	1.00
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	23267.5	0.000244	0.00
Random	10	value	Phenotypic	Phylogenetic	240	240	27609.5	0.433000	1.00
Random	100	value	Phenotypic	Phylogenetic	240	240	26008.5	0.065800	0.63
Tournament	10	value	Phenotypic	Phylogenetic	240	240	28652.5	0.922000	1.00
Tournament	100	value	Phenotypic	Phylogenetic	240	240	26073.0	0.071100	0.77



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

3.7.2.3 Mean pairwise distance vs. phenotypic Shannon diversity

```
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)),
    fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)),
    fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)),
    fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(phenotypic)))
```

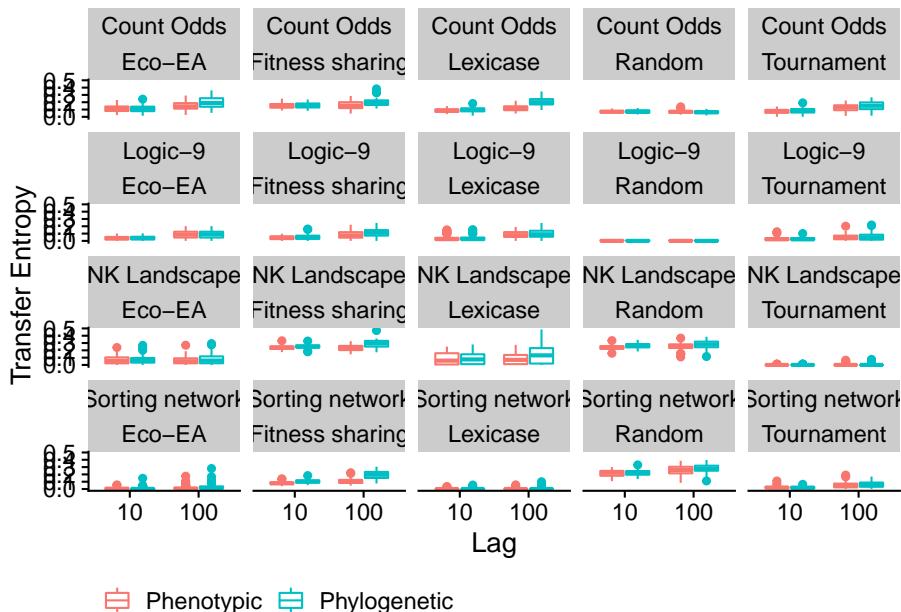
```

fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(phenotype_diver)
fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(phenotype_diver
)

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*$")
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~problem_name*selection_name) +
  scale_x_discrete("Lag") +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("") +
  theme(legend.position = "bottom")

```



```

# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	28432.0	0.809000	1.00
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	25094.5	0.014800	0.14
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	25665.0	0.039100	0.38
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	18866.0	0.000000	0.00
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	27887.0	0.545000	1.00
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	23061.5	0.000142	0.00
Random	10	value	Phenotypic	Phylogenetic	240	240	27546.0	0.409000	1.00
Random	100	value	Phenotypic	Phylogenetic	240	240	27250.5	0.307000	1.00
Tournament	10	value	Phenotypic	Phylogenetic	240	240	28662.0	0.927000	1.00
Tournament	100	value	Phenotypic	Phylogenetic	240	240	27121.5	0.267000	1.00

```

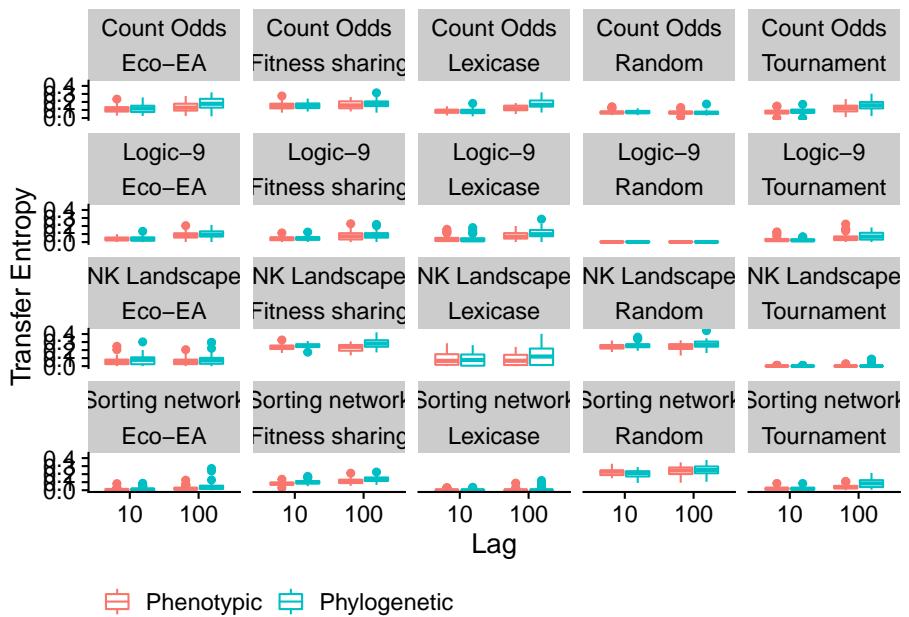
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)))
  fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)))
  fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(mean_pheno)))
  fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(pheno)))
  fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(pheno)))
  fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(pheno)))
)

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*$")
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~problem_name*selection_name) +
  scale_x_discrete("Lag") +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("") +
  theme(legend.position = "bottom")

```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	27071.5	2.55e-01	1.000000	ns
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	24236.0	2.67e-03	0.026700	*
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	26368.0	1.10e-01	1.000000	ns
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	24482.0	4.49e-03	0.044900	*
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	29278.0	7.52e-01	1.000000	ns
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	23573.5	5.30e-04	0.005300	**
Random	10	value	Phenotypic	Phylogenetic	240	240	28256.0	7.20e-01	1.000000	ns
Random	100	value	Phenotypic	Phylogenetic	240	240	27327.0	3.32e-01	1.000000	ns
Tournament	10	value	Phenotypic	Phylogenetic	240	240	28964.5	9.13e-01	1.000000	ns
Tournament	100	value	Phenotypic	Phylogenetic	240	240	22758.0	6.36e-05	0.000636	***



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

3.7.2.5 Mean evolutionary distinctiveness vs. phenotypic Shannon diversity

```
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_evolution)),
    fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_evolution)),
    fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(mean_phenotype_evolution))
```

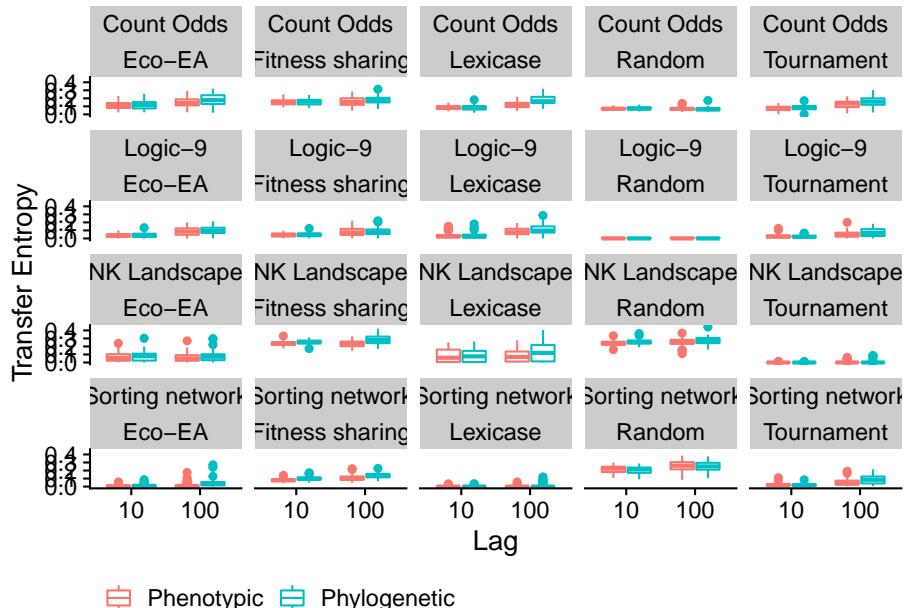
```

fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(phenotype)))
fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(phenotype)))
fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(phenotype)))

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*$")
res>Type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~problem_name*selection_name) +
  scale_x_discrete("Lag") +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("") +
  theme(legend.position = "bottom")

```



selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.adj	p.adj
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	27878.0	0.544000	1.00000	ns
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	24139.5	0.002160	0.02160	*
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	26469.0	0.125000	1.00000	ns
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	24377.0	0.003610	0.03610	*
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	28847.5	0.975000	1.00000	ns
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	23292.5	0.000261	0.00261	**
Random	10	value	Phenotypic	Phylogenetic	240	240	28179.0	0.682000	1.00000	ns
Random	100	value	Phenotypic	Phylogenetic	240	240	28542.5	0.865000	1.00000	ns
Tournament	10	value	Phenotypic	Phylogenetic	240	240	28976.5	0.907000	1.00000	ns
Tournament	100	value	Phenotypic	Phylogenetic	240	240	23925.0	0.001250	0.01250	*

```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

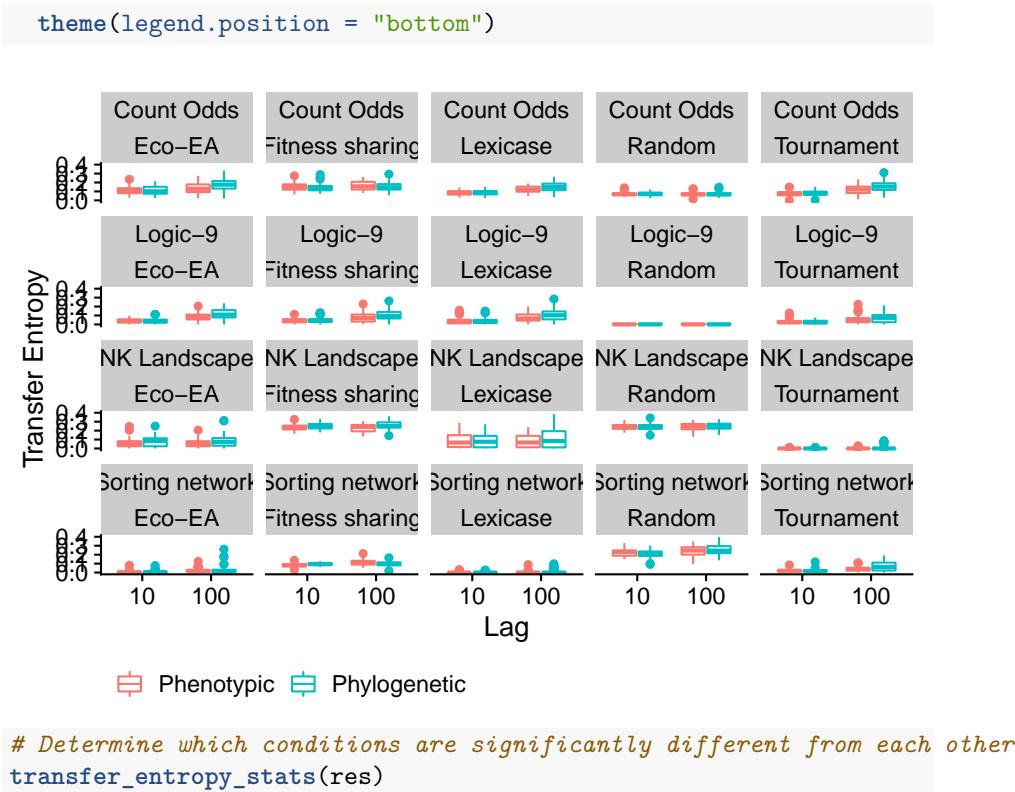
3.7.2.6 Variance evolutionary distinctiveness vs. phenotypic richness

```
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
  summarise(
    fit_phylo_10 = condinformation(discretize(max_performance), discretize(lag(variance_phenotype_etc)))
    fit_phylo_100 = condinformation(discretize(max_performance), discretize(lag(variance_phenotype_etc)))
    fit_phylo_500 = condinformation(discretize(max_performance), discretize(lag(variance_phenotype_etc)))
    fit_pheno_10 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
    fit_pheno_100 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
    fit_pheno_500 = condinformation(discretize(max_performance), discretize(lag(phenotype_num_taxa)))
  )

res <- res %>% pivot_longer(cols=contains("o_10"))
res$offset <- str_extract(res$name, "[[:digit:]]*$")
res$type <- case_when(str_detect(res$name, "phylo") ~ "Phylogenetic", TRUE ~ "Phenotypic")

ggplot(
  res %>% filter(str_detect(name, "fit_ph*")),
  aes(
    x=as.factor(offset),
    y=value,
    color=Type
  )
) +
  geom_boxplot() +
  facet_wrap(~problem_name*selection_name) +
  scale_x_discrete("Lag") +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("") +
```

selection_name	offset	.y.	group1	group2	n1	n2	statistic	p	p.a
Eco-EA	10	value	Phenotypic	Phylogenetic	240	240	27618.0	0.43700	1.000
Eco-EA	100	value	Phenotypic	Phylogenetic	240	240	24904.5	0.01040	0.104
Fitness sharing	10	value	Phenotypic	Phylogenetic	240	240	27307.0	0.32600	1.000
Fitness sharing	100	value	Phenotypic	Phylogenetic	240	240	28643.0	0.91800	1.000
Lexicase	10	value	Phenotypic	Phylogenetic	240	240	28782.5	0.99100	1.000
Lexicase	100	value	Phenotypic	Phylogenetic	240	240	25088.0	0.01390	0.139
Random	10	value	Phenotypic	Phylogenetic	240	240	29327.0	0.72900	1.000
Random	100	value	Phenotypic	Phylogenetic	240	240	28087.5	0.63900	1.000
Tournament	10	value	Phenotypic	Phylogenetic	240	240	28982.5	0.90400	1.000
Tournament	100	value	Phenotypic	Phylogenetic	240	240	24187.0	0.00227	0.0227



3.7.3 Transfer entropy between types of diversity

While we're calculating transfer entropy, we might as well also calculate it between phenotypic diversity and phylogenetic diversity, as these could potentially also be in a feedback loop.

3.7.3.1 Max pairwise distance and phenotypic richness

```

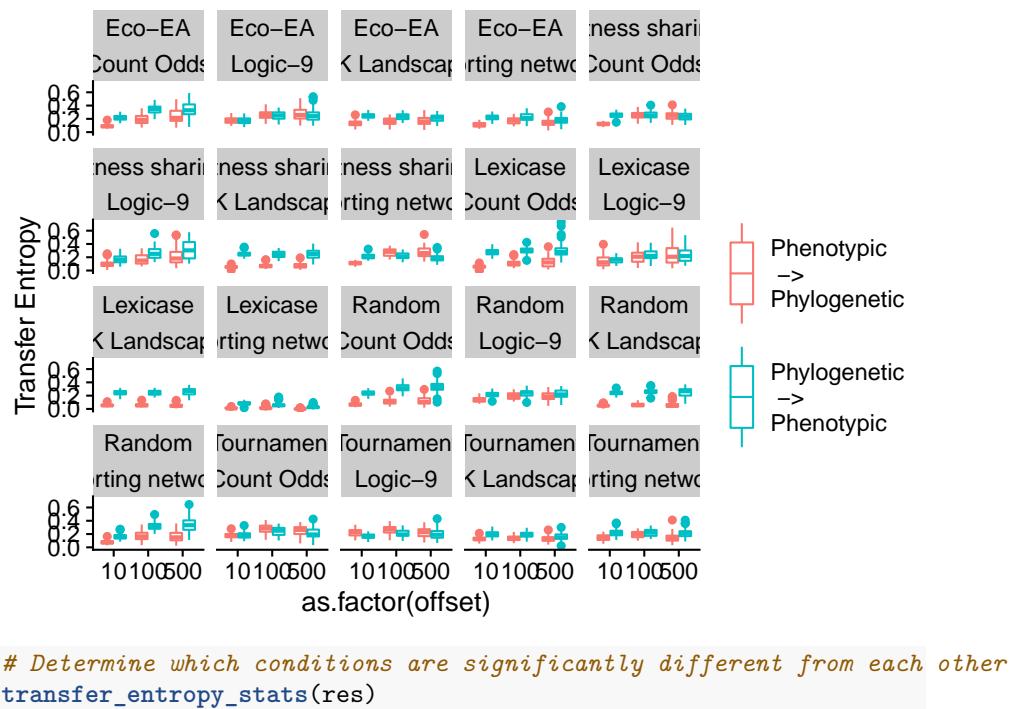
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  phen_phylo_10 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(max_phenotype_pairwise_distance, 1)),
                                    discretize(lag(phenotype_num_taxa, 1))),
  phen_phylo_100 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(max_phenotype_pairwise_distance, 10)),
                                    discretize(lag(phenotype_num_taxa, 10))),
  pheno_phylo_500 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(max_phenotype_pairwise_distance, 50)),
                                    discretize(lag(phenotype_num_taxa, 50))),
  phylo_pheno_10 = condinformation(discretize(max_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 1)),
                                    discretize(lag(max_phenotype_pairwise_distance, 1))),
  phylo_pheno_100 = condinformation(discretize(max_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 10)),
                                    discretize(lag(max_phenotype_pairwise_distance, 10))),
  phylo_pheno_500 = condinformation(discretize(max_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 50)),
                                    discretize(lag(max_phenotype_pairwise_distance, 50)))))

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*\$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n",
                       TRUE ~ "Phylogenetic\n\t->\nPhenotypic\n")

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name * problem_name) +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("")

```

selection_name	offset	.y.	group1	group2	n1
Eco-EA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240



3.7.3.2 Mean pairwise distance and phenotypic richness

```

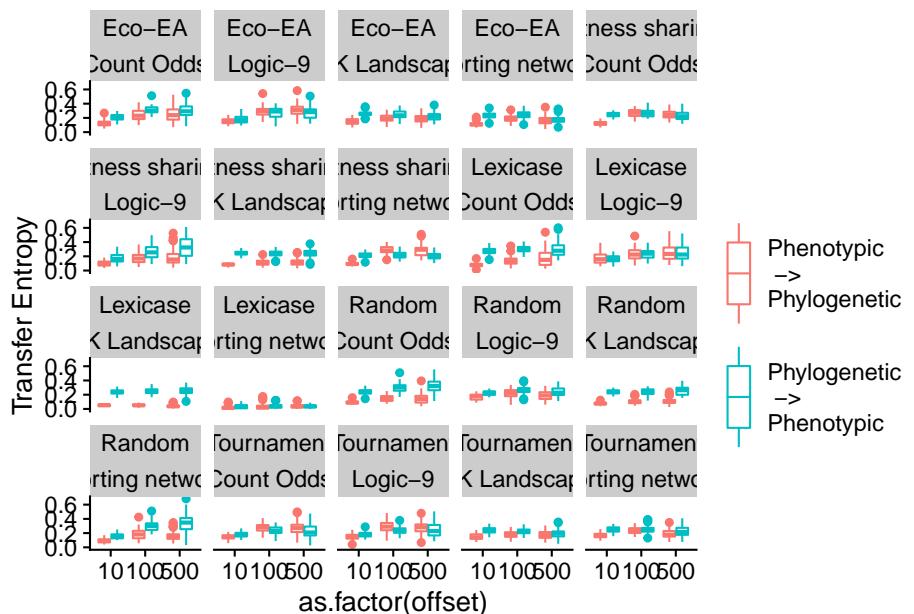
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  phen_phylo_10 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(mean_phenotype_pairwise_distance, 1)),
                                    discretize(lag(phenotype_num_taxa, 1))),
  phen_phylo_100 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(mean_phenotype_pairwise_distance, 10)),
                                    discretize(lag(phenotype_num_taxa, 10))),
  pheno_phylo_500 = condinformation(discretize(phenotype_num_taxa),
                                    discretize(lag(mean_phenotype_pairwise_distance, 50)),
                                    discretize(lag(phenotype_num_taxa, 50))),
  phylo_pheno_10 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 1)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 1))),
  phylo_pheno_100 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 10)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 10))),
  phylo_pheno_500 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_num_taxa, 50)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 50)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n",
                       TRUE ~ "Transfer Entropy")

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name*problem_name) +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("")

```

selection_name	offset	.y.	group1	group2	n1
Eco-EA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

3.7.3.3 Mean pairwise distance and shannon diversity

```

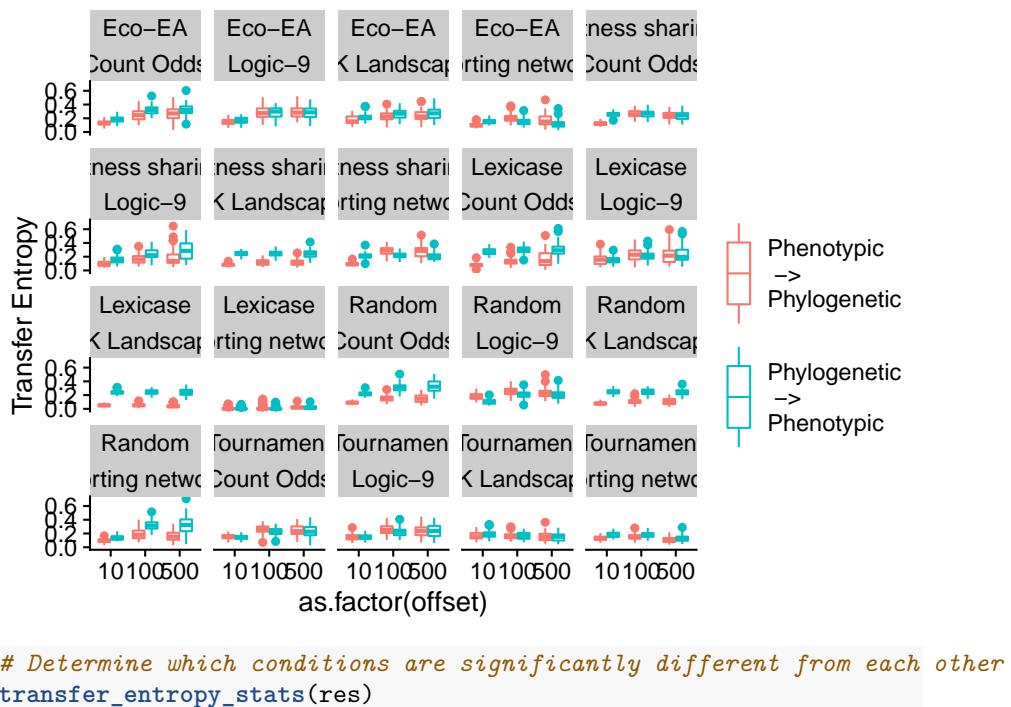
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  phen_phylo_10 = condinformation(discretize(phenotype_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance, 1)),
                                    discretize(lag(phenotype_diversity, 1))),
  phen_phylo_100 = condinformation(discretize(phenotype_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance, 10)),
                                    discretize(lag(phenotype_diversity, 10))),
  pheno_phylo_500 = condinformation(discretize(phenotype_diversity),
                                    discretize(lag(mean_phenotype_pairwise_distance, 50)),
                                    discretize(lag(phenotype_diversity, 50))),
  phylo_pheno_10 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_diversity, 1)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 1))),
  phylo_pheno_100 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_diversity, 10)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 10))),
  phylo_pheno_500 = condinformation(discretize(mean_phenotype_pairwise_distance),
                                    discretize(lag(phenotype_diversity, 50)),
                                    discretize(lag(mean_phenotype_pairwise_distance, 50)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t-\nPhylogenetic\n",
                       TRUE ~ "Phenotypic\n\t-\nPhylogenetic\n")

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
) +
  geom_boxplot() +
  facet_wrap(~selection_name*problem_name) +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("")

```

selection_name	offset	.y.	group1	group2	n1
Eco-EA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240



3.7.3.4 Mean evolutionary distinctiveness and phenotypic richness

```

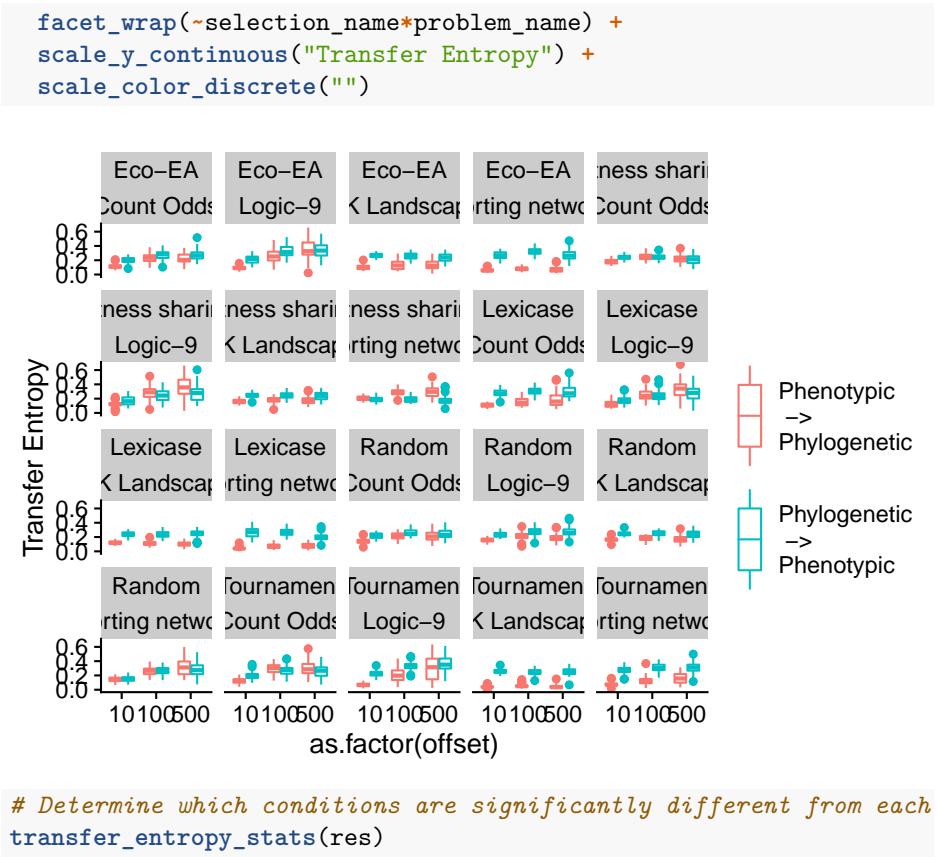
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  phen_phylo_10 = condinformation(
    discretize(phenotype_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1)),
    discretize(lag(phenotype_num_taxa, 1))),
  phen_phylo_100 = condinformation(
    discretize(phenotype_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10)),
    discretize(lag(phenotype_num_taxa, 10))),
  pheno_phylo_500 = condinformation(
    discretize(phenotype_num_taxa),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 50)),
    discretize(lag(phenotype_num_taxa, 50))),
  phylo_pheno_10 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(phenotype_num_taxa, 1)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1))),
  phylo_pheno_100 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(phenotype_num_taxa, 10)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10))),
  phylo_pheno_500 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(phenotype_num_taxa, 50)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 50)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n", TRUE ~ "Phenotypic\n\t->\nPhylogenetic\n", FALSE ~ "Phenotypic\n\t->\nPhenotypic\n")

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
) +
  geom_boxplot() +

```

selection_name	offset	.y.	group1	group2	n1
Eco-EA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Eco-EA	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Fitness sharing	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Lexicase	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Random	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240
Tournament	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240



3.7.3.5 Mean evolutionary distinctiveness and shannon diversity

```

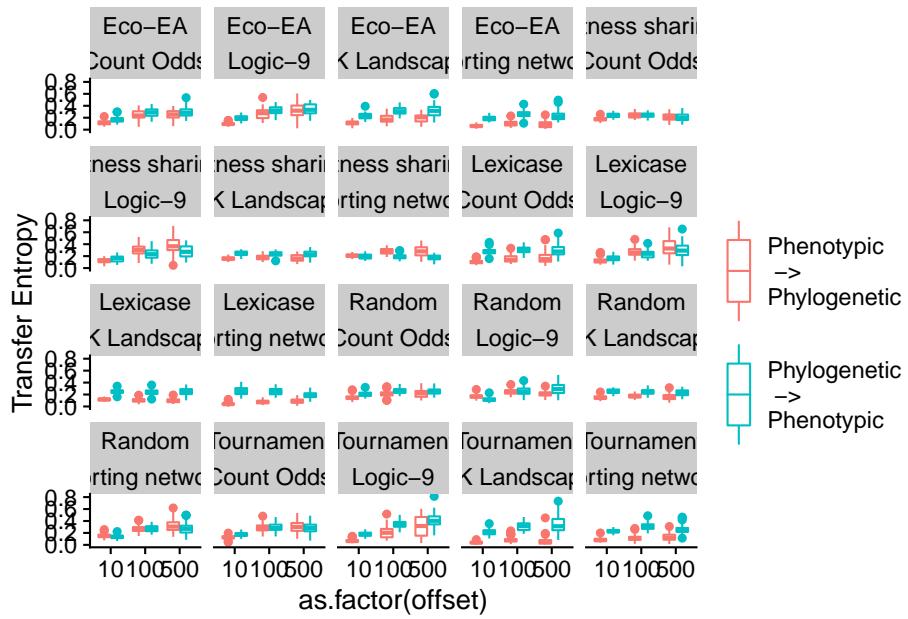
res <- data %>% group_by(SEED, selection_name, problem_name) %>%
summarise(
  phen_phylo_10 = condinformation(
    discretize(pheno_type_diversity),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1)),
    discretize(lag(pheno_type_diversity, 1))),
  phen_phylo_100 = condinformation(
    discretize(pheno_type_diversity),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10)),
    discretize(lag(pheno_type_diversity, 10))),
  pheno_phylo_500 = condinformation(
    discretize(pheno_type_diversity),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 50)),
    discretize(lag(pheno_type_diversity, 50))),
  phylo_pheno_10 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_type_diversity, 1)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 1))),
  phylo_pheno_100 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_type_diversity, 10)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 10))),
  phylo_pheno_500 = condinformation(
    discretize(mean_phenotype_evolutionary_distinctiveness),
    discretize(lag(pheno_type_diversity, 50)),
    discretize(lag(mean_phenotype_evolutionary_distinctiveness, 50)))
)

# Turn Transfer Entropy columns into rows
res <- res %>% pivot_longer(cols=contains("phylo"))
# Pull lag into a column
res$offset <- str_extract(res$name, "[[:digit:]]*$")
# Make column indicating direction of transfer entropy
res$type <- case_when(str_detect(res$name, "phylo_pheno") ~ "\nPhenotypic\n\t->\nPhylogenetic\n", TRUE)

ggplot(
  res,
  aes(
    x=as.factor(offset),
    y=value,
    color=type
  )
)

```

```
) +
  geom_boxplot() +
  facet_wrap(~selection_name*problem_name) +
  scale_y_continuous("Transfer Entropy") +
  scale_color_discrete("")
```



```
# Determine which conditions are significantly different from each other
transfer_entropy_stats(res)
```

selection_name	offset	.y.	group1	group2	n1	n2	statist
Eco-EA	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	273
Eco-EA	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	1213
Eco-EA	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	1663
Fitness sharing	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	1551
Fitness sharing	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	3413
Fitness sharing	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	3329
Lexicase	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	293
Lexicase	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	99
Lexicase	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	1513
Random	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	237
Random	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	191
Random	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	201
Tournament	10	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	19
Tournament	100	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	86
Tournament	500	value	Phenotypic -> Phylogenetic	Phylogenetic -> Phenotypic	240	240	148

Bibliography

- Allen, M., Poggiali, D., Whitaker, K., Marshall, T. R., van Langen, J., and Kievit, R. A. (2021). Raincloud plots: a multi-platform tool for robust data visualization. *Wellcome Open Research*, 4:63.
- Hernandez, J. G., Lalejini, A., and Ofria, C. (2021). An Exploration of Exploration: Measuring the ability of lexicase selection to find obscure pathways to optimality. *arXiv:2107.09760 [cs]*. arXiv: 2107.09760.