

Replaying Life's Tape: The Origin of Lineage-Dependent Adaptive Responses in a Long-Term Evolution Experiment with Digital Organisms

Jason N. Bundy, Michael Travisano, Charles Ofria, Richard E. Lenski

Table of contents

1. Abstract
2. Installation
3. Usage
4. Development

1. Abstract/p>

We “replayed life’s tape” virtually by evolving ten populations in parallel. Using the digital evolution software Avida enabled us to perform a long-term evolution experiment by tracking changes in populations for tens of thousands of generations much faster than would be possible with any biological organisms. All populations were derived from a common ancestor and evolved under identical conditions. We tracked the evolution of two traits in each population: size (genomic length) and fitness. We inferred the roles of natural selection and stochastic processes in the changes in each trait. Changes in size were dictated by chance, while changes in fitness were influenced by both selection and chance processes. We also examined changes in adaptation and diversification in each trait by comparing end-of-experiment results with those from an earlier timepoint. We saw a significant increase in the among-population diversity in size as well as substantial increases in fitness and its variance after the first few thousand generations, indicating that the evolutionary trajectories continued to diverge even after long periods that are rarely captured in evolution experiments. We also examined the correlation between size and fitness both across populations (cross-sectional) and within lineages (longitudinal) throughout the experiment. The cross-sectional regressions indicate a weak relationship between the traits, whereas the longitudinal regressions revealed strong, but idiosyncratic, associations. Contrary to predictions of consistent phenotypic trends under these conditions, our results demonstrate multiple adaptive responses that are unique to each lineage’s history.

2. Installation

The latest version of Avida can be downloaded at <https://avida.devosoft.org/>. If you are new to Avida, you can find helpful resources for new users on the Wiki: <https://github.com/devosoft/avida/wiki/Beginner-Documentation>. The instructions provided here assume familiarity with Avida and its basic configuration files. Further, you are free to modify and alter any of the included configuration files for your own projects which may utilize any directory structure that suits you. The instructions provided here will refer to the directory structure and configuration files we used in support of the current manuscript.

Directory Structure

In your primary project directory you will want to have the following sub-directories (i.e. folders):

- config: The directory contains Avida's setup and configuration files.
- logs: This directory is for storing the text files that record information about various scripts.
- output_phase1: This directory will contain the data output by Avida during the first phase of the experiment.
- output_phase2: This directory will contain the data output by Avida during the second phase of the experiment.
- output_analysis: This directory will contain the files and subfolders that result from analyzing the data.
- python_scripts: This directory contains the python scripts necessary for analyzing the data. It contains 'analysis' and 'operations' sub-directories.
- resources: This directory contains a subfolder for storing fonts and can be used as a container for additional project resources.

config

This folder contains the Avida executable (avida), the Avida configuration file (avida.cfg), the analyze mode configuration file (analyze.cfg), and a subdirectory containing folders for various types of setup files. The "setup" subdirectory contains folders for environment files, events lists, instruction sets, saved organisms, and saved populations.

logs

This folder contains simple text files that record the operation and output of various python scripts used in the analysis. These are similar to a "lab notebook" generated automatically by the scripts. They can be useful in keeping track of the project as well as having clear textual representations of what the various scripts have done. They can also be useful in detecting errors.

output_analysis

This will contain dataframes and two subdirectories. One, "raw", for copies of unprocessed dataframes used during the analysis and the other, "masked", for filtered datasets, plots, and figures that result from running various analysis scripts.

output_phase1

This will contain the individual output directories that correspond to each individual run. Each individual output directory will contain a "data" subfolder with the individual run's data, a "setup" subfolder for storing initial setup files (i.e. environments, events, instruction sets, saved organisms, and saved populations), as well as the avida executable file, the analyze mode configuration file (i.e. 'analyze.cfg'), and the Avida configuration file (i.e. 'avida.cfg').

output_phase2

This will contain the individual output directories that correspond to each individual run. Each individual output directory will contain a "data" subfolder with the individual run's data, a "setup" subfolder for storing initial setup files (i.e. environments, events, instruction sets, saved organisms, and saved populations), as

well as the avida executable file, the analyze mode configuration file (i.e. 'analyze.cfg'), and the Avida configuration file (i.e. 'avida.cfg').

python_scripts

This contains the Python scripts for the project in two sub-folders. The first, "analysis" contains scripts used to analyze data, while the other, "operations", contains scripts used in various procedures throughout the project including the creation and sorting of data dictionaries and related .csv files, label formatting, and miscellaneous operations.

resources

This folder can be used to store any other files or assets you'd like to keep organized with the project. By default, it contains one subfolder, "fonts", for storing fonts that can be used for plotting.

The full directory structure used for this project is shown below. If you'd like to recreate our experiment without modifying the original code you will want to replicate this structure. If you'd like to use a different directory structure, you can use the structure below as a reference and modify the code to suit your needs. It's important to remember that subdirectories labeled "...individual run directories" will be output by Avida and therefore you do not need to manually create the subdirectories (i.e. data and setup).

Project directory:

- config
 - setup
 - environments
 - events
 - instructions
 - organisms
 - populations
- logs
 - created_events
- output_analysis
 - masked
 - figures
 - ACH_images
 - paired_ttest
 - phase1_plot
 - raw
- output_phase1
 - ...individual run directories
 - data
 - setup
- python_scripts
 - analysis
 - operations
 - axis_labels
- resources
 - fonts

3. Usage

We performed our experiment using a custom run system developed by the Digital Evolution Lab at Michigan State University using the High Performance Computing Center (HPCC) maintained by Michigan State University's Institute for Cyber-Enabled Research (ICER). We have included the run_list files we used to perform the experiment. However, below we describe the parameters necessary for recreating our experiment in the absence of these resources. Using the same initial seed should generate the same results. We have also included the configuration files necessary to reproduce our experimental design.

Phase 1

In the first phase of the experiment we evolved ten isolated, replicate populations from a single ancestral genotype in a common environment. You can reproduce our results using the following configuration details with the files included in this repository.

Seed: 101-110

avida configuration file: avida.cfg

environment: env_38tasks_noequ_even.cfg

events list: events_500k.cfg

instruction set: instset-heads.cfg

default organism: default_heads.org

Analysis

The primary data for each run following the second phase will be found in the "average.dat" file within the "data" directory for each run. The data we used in our analysis are labeled "fitness" and "copied size".

4. Recreating plots and tables

There is a file called "recreating_visuals.md" in the scripts folder that describes how to create each figure and table in the manuscript.

5. Development

Creative Commons CC-BY_NC: Anyone can share, reuse, remix, or adapt this material, providing this is not done for commercial purposes and the original authors are credited and cited. At present, inquiries and support will only be addressed in support of the present manuscript. To inquire, please email author Jason Bundy (bundyjay86@gmail.com).