Hereditary Stratigraphy: Genome Annotations to Enable Phylogenetic Inference over Distributed Populations

Matthew Andres Moreno, Emily Dolson, and Charles Ofria

Michigan State University, East Lansing, MI 48103
mmore500@msu.edu

Abstract

Phylogenies provide direct accounts of the evolutionary trajectories behind evolved artifacts in genetic algorithm and artificial life systems. Phylogenetic analyses can also enable insight into evolutionary and ecological dynamics such as selection pressure and frequency-dependent selection. Traditionally, digital evolution systems have recorded data for phylogenetic analyses through perfect tracking where each birth event is recorded in a centralized data structure. This approach, however, does not easily scale to distributed computing environments where evolutionary individuals may migrate between a large number of disjoint processing elements. To provide for phylogenetic analyses in these environments, we propose an approach to enable phylogenies to be inferred via heritable genetic annotations rather than directly tracked. We introduce a "hereditary stratigraphy" algorithm that enables efficient, accurate phylogenetic reconstruction with tunable, explicit trade-offs between annotation memory footprint and reconstruction accuracy. In particular, we demonstrate an approach that enables estimation of the most recent common ancestor (MRCA) between two individuals with fixed relative accuracy irrespective of lineage depth while only requiring logarithmic annotation space complexity with respect to lineage depth. This approach can estimate, for example, MRCA generation of two genomes within 10% relative error with 95% confidence up to a depth of a trillion generations with genome annotations smaller than a kilobyte. We also simulate inference over known lineages, recovering up to 85.70% of the information contained in the original tree using 64-bit annotations.

Introduction

In traditional serially-processed digital evolution experiments, phylogenetic trees can be tracked perfectly as they progress (Bohm et al., 2017; Wang et al., 2018; Lalejini et al., 2019) rather than reconstructed afterward, as must be done in most biological studies of evolution. Such direct phylogenetic tracking enables experimental possibilities unique to digital evolution, such as perfect reconstruction of the sequence of phylogenetic states that led to a particular evolutionary outcome (Lenski et al., 2003; Dolson et al., 2020). In a shared-memory context, it is not difficult to maintain a complete phylogeny by ensuring that offspring retain a permanent reference to their parent (or vice versa). As simulations progress, however, memory usage would balloon if all simulated organisms were stored permanently. Garbage collecting extinct lineages and saving older history to disk greatly ameliorates this issue (Bohm et al., 2017; Dolson et al., 2019).

If sufficient memory or disk space can be afforded to log all reproduction events, recording a perfect phylogeny in a distributed context is also not especially difficult. Processes could maintain records of each reproduction event, storing the parent organism (and its associated process) with all generated offspring (and their destination processes). As long organisms are uniquely identified globally, these "dangling ends" could be joined in postprocessing to weave a continuous global phylogeny. Of course, for the huge population sizes made possible by distributed systems, such stitching may become a demanding task in and of itself. Additionally, even small amounts of lost or corrupted data could fundamentaly degrade tracking by disjoining large tree subsections.

However, if memory and disk space are limited, distributed phylogeny tracking becomes a more burdonsome challenge. A naive approach might employ a server model to maintain a central store of phylogenetic data. Processes would dispatch notifications of birth and death events to the server, which would curate (and gabage collect) phylogenetic history much the same as current serial phylogenetic tracking implementations. Unfortunately, this server model approach would present scalability challenges: burden on the server process would worsen in direct proportion to processor count. This approach would also be similarly brittle to any lost or corrupted data.

A more scalable approach might record birth and death events only on the process(es) where they unfold. However, lineages that went extinct locally could not be safely garbage collected until the extinction of their offspring's lineages on other processors could be confirmed. Garbage collection would thus require extinction notifications to wind back across processes each lineage had traversed. Again, this approach would also be brittle to loss or corruption of data.

In a distributed context — especially, a distributed, best-effort context — phylogenetic reconstruction (as opposed to tracking) could prove simpler to implement, more efficient at runtime, and more robust to data loss while providing sufficient information to address experimental questions of interest. However, phylogenetic reconstruction from genomes with a traditional model of divergence under grandual accumulation of random mutations poses its own difficulties, including

- accounting for heterogeneity in evolutionary rates (i.e., the rate at which mutations accumulate due to divergent mutation rates or selection pressures) between lineages (Lack and Van Den Bussche, 2010),
- performing sequence alignment (Casci, 2008),

- mutational saturation (Hagstrom et al., 2004),
- appropriately selecting and applying complex reconstruction algorithms (Kapli et al., 2020), and
- computational intensity (Sarkar et al., 2010).

The computational flexibility of digital artificial life experiments provides a unique opportunity to ovecome these challenges: designing heritable genome annotations specifically to ensure simple, efficient, and effective phylogenetic reconstruction. For maximum applicability of such a solution, these annotations should be phenotypically neutral heritable instrumentation (Stanley and Miikkulainen, 2002) that can be applied to any digital genome.

In this paper, we present "hereditary stratigraphy," a novel heritable genome annotation system to facilitate post-hoc phylogenetic inference on asexual populations. This system allows explicit control over trade-offs between space complexity and accuracy of phylogenetic inference. Instead of modeling genome components diverging through a neutral mutational process, we keep a record of historical checkpoints that allow comparison of two lineages to identify the range of time in which they diverged. Careful management of these checkpoints allows for a variety of trade-off options, including:

- linear space complexity and fixed-magnitude inference error,
- constant space complexity and inference error linearly proportional to phylogenetic depth, and
- logarithmic space complexity and inference error linearly proportional to time elapsed since MRCA (which we suspect will be the most broadly useful trade-off).

In Methods we motivate and explain the hereditary stratigraphy approach. Then, in Results and Discussion we simulate post-hoc inference on known phylogenies to assess the quality of phylogenetic reconstruction enabled by the hereditary stratigraphy method.

Methods

This section will introduce intuition for the strategy of our hereditary stratigraph approach, define the vocabulary we developed to describe aspects of this approach, overview configurable aspects of the approach, present mathematical exposition of the properties of space complexity and inference quality under particular configurations, and then recap digital experiments that demonstrate this approach in an applied setting.

Hereditary Strata and the Hereditary Stratigraphic Column

Our algorithm, particularly the vocabulary we developed to describe it, draws loose inspiration from the concept of geological stratigraphy, inference of natural history through analysis of successive layers of geological material (Steno, 1916). As an introductory intuition, suppose a body of rock being built up through regular, discrete events depositing of geological material. Note that in such a scenario we could easily infer the age of the body of rock by counting up the number of layers present. Next, imagine making a copy of the rock body in its partially-formed state and then moving it far away. As time runs forward on these two rock bodies, independent layering processes will cause consistent disparity in the layers forming on each forwards from their point of separation.

To deduce the historical relationship of these rock bodies, we could simply align and compare their layers. Layers from their

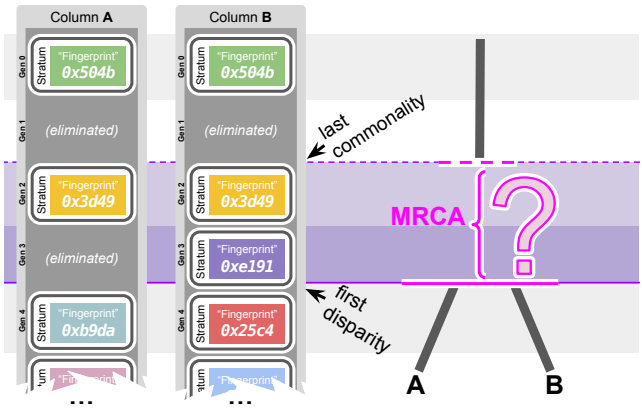


Figure 1: Inferring the generation of the most-recent common ancestor (MRCA) of two hereditary stratigraphic columns "A" and "B". Columns are aligned at corresponding generations. Then the first generation with disparate "fingerprints" is determined. This provides a hard upper bound on the generation of the MRCA: these strata *must* have been deposited along separate lines of descent. Searching backward for the first commonality preceding that disparity provides a soft lower bound on the generation of the MRCA: these strata evidence common ancestry but *might* collide by chance. Some strata mmay have been eliminated from the columns, as shown, in order to save space at the cost of increasing uncertainty of MRCA generation estimates.

base up through the first disparity would correspond to shared ancestry; further disparate layers would correspond to diverged ancestry. Figure 1 depicts the process of comparing columns for phylogenetic inference.

Shifting now from intuition to implementation, a fixed-length randomly-generated binary tag provides a suitable "fingerprint" mechanism mirroring our metaphorical "rock layers." We call this "fingerprint" tag a *differentia*. The width of this tag controls the probability of spurious collisions between independently generated instances. At 64 bits wide the tag effectively functions as a UID: collisions between randomly generated tags are so unlikely ($p < 5.42 \times 10^{-20}$) they can essentially be ignored. At the other end of the spectrum, collision probability would be $1/256$ for a single byte and $1/2$ for a single bit. In the case of narrow differentia, in order to set a lower bound for the MRCA generation, you would have to backtrack common strata from the last commonality until the probability of that many successive spurious collisions was enough to satisfy your desired confidence level (e.g., 95% confidence). Even then, there would be a possibility of the the true MRCA falling before the estimated lower bound. Note, however, that no matter the width of the differentia the generation of the first discrepancy provides a hard upper bound on the generation of the MRCA.

In accordance with our geological analogy, we refer to the packet of data accumulated each generation as a *stratum*. This packet contains the differentia and, although not employed in this work, could hold other arbitrary user-defined data (i.e., simulation timestamp, phenotype characteristics, etc.). Again in accordance with the geological analogy, we refer to the chronological stack of strata that accumulate over successive generations as a *hereditary*

stratigraphic column.

Stratum Retention Policy

As currently stated, strata in each column will accumulate proportionally to the length of evolutionary history simulated. In an evolutionary run with thousands or millions of generations, this approach would soon become intractable — particularly when columns are serialized and transmitted between distributed computing elements. To solve this problem, we can trade off precision for compactness by strategically deleting strata from columns as time progresses. Figure 2 overviews how stratum deposit and stratum elimination might progress over two generations under the hereditary stratigraphic column scheme.

Different patterns of deletion will lead to different trade-offs, both in terms of the scaling relationship of column size to generations elapsed and in terms of the arrangement of inference precision over evolutionary history (i.e., focusing precision on more recent evolutionary history versus spreading it evenly over the entire history).

We refer to the rule set used to selectively eliminate strata over time as the "stratum retention policy." We explore several different retention policy designs here, and implement our software to allows for free, modular interchange of retention policies.

Our software allows specification of a policy as either a "predicate" or a "generator." The predicate method requires a function that takes the generation of a stratum and the current number of strata deposited and returns whether that stratum should be retained at that point in time. The generator method requires a function that takes the current number of strata deposited and yields the set of generations that should be deleted at that point in time. Although the predicate form of a policy is useful for analyzing and proving properties of policies, the generator form is generally more efficient in practice. We provide equivalent predicate and generator implementations for each stratum retention policy discussed here.

Strata elimination causes a stratum's position within the column data structure to no longer correspond to the generation it was deposited. Therefore, it may seem necessary to store the generation of deposit within the stratum data structure. However, for all deterministic retention policies a perfect mapping exists backwards from column index to recover generation deposited without storing it. We provide this formula for each stratum retention policy surveyed here. Finally, for each policy we provide a formula to calculate the exact number of strata retained under any parameterization after n generations.

The next subsections introduce several stratum retention policies, explain the intuition behind their implementation, and elaborate their space complexity and resolution properties. For each policy, patterns of stratum retention are illustrated in Figure 3. The formulas for number of strata retained after n generations, the formulas to calculate stratum deposit generation from column index, and the retention predicate specifications of each policy are available in Supplementary Listing 5 (Moreno et al., 2022). The generator specification of each policy is available in Supplementary Listing 1 (Moreno et al., 2022). For tapered depth-proportional resolution and recency-proportional resolution, the accuracy of MRCA estimation can also be explored via an interactive in-browser web applet at `https://hopth.ru/bi`.

Fixed Resolution Stratum Retention Policy

The fixed resolution retention policy imposes a fixed absolute upper bound r on the spacing between retained strata. The strategy is simple: permanently retain a stratum every rth generation. (For arbitrary reasons of implementation convenience, we also require each stratum to be retained during at least the generation it is deposited). See top panel of Figure 3.

This retention policy suffers from linear growth in a column's memory footprint with respect to number of generations elapsed: every rth generation generation a new stratum is permanently retained. For this reason, it is likely not useful in practice except potentially in scenarios where the number of generations is small and fixed in advance. We include it here largely for illustrative purposes as a gentle introduction to retention policies.

Depth-Proportional Resolution Stratum Retention Policy

The depth-proportional resolution policy ensures spacing between retained strata will be less than or equal to a proportion $1/r$ of total number of strata deposited n. Achieving this limit on uncertainty requires retaining sufficient strata so that no more than n/r generations elapsed between any two strata. This policy accumulates retained strata at a fixed interval until twice as many as r are at hand. Then, every other retained stratum is purged and the cycle repeats with a new twice-as-wide interval between retained strata. See second from top panel of Figure 3.

When comparing stratigraphic columns from different generations, the resolution guarantee holds in terms of the number of generations experienced by the older of the two columns. Because this retention policy is deterministic, for two columns with the same policy, every stratum that is held by the older column is also guaranteed to be present in the younger column (unless it hasn't yet been deposited on the younger column). Therefore, the strata that would enable the desired resolution when comparing two columns of the same age are guaranteed to be available, even when one columnn has elapsed more generations.

Because the number of strata retained under this policy is bounded as $2r+1$, space complexity scales as $O(1)$ with respect to the number of strata deposited. It follows that the MRCA generation estimate uncertainty scales as $O(n)$ with respect to the number of strata deposited.

Tapered Depth-Proportional Resolution Stratum Retention Policy

This policy refines the depth-proportional resolution policy to provide a more stable column memory footprint over time. The naive depth-proportional resolution policy builds up strata until twice as many are present as needed then purges half of them all at once. The tapered depth-proportional resolution policy functions identically to the depth-proportional policy except that it removes unnecessary strata gradually from back to front as new strata are deposited, instead of eliminating them simultaneously. See third from top panel of Figure 3.

The column footprint stability of this variation makes it easier to parameterize our experiments to ensure comparable end-state column footprints for fair comparison between retention policies, in addition to making this policy likely better suited to most use cases.

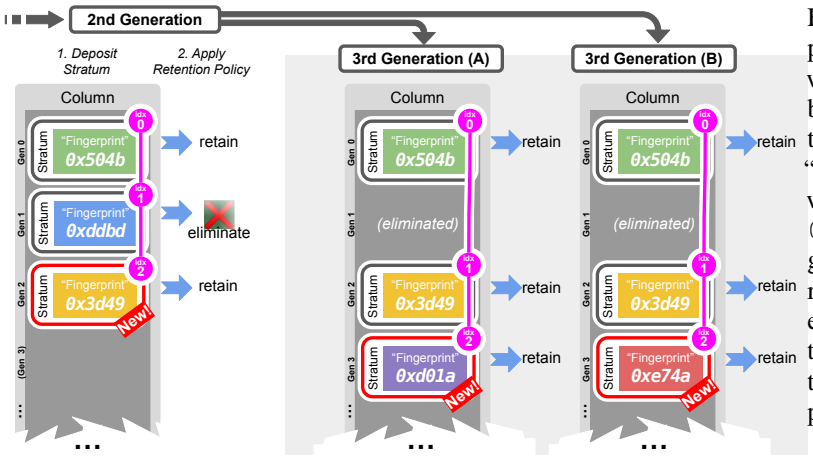


Figure 2: Cartoon illustration of stratum deposit process. This process marks the elapse of a generation when a hereditary stratigraphic column is inherited by an offspring. First, a new stratum is appended to the end of the column with a randomly-generated "fingerprint." This "fingerprint" distinguishes strata that were generated along disparate lines of descent (e.g., `0xd01a` for 3rd Generation A and `0xe74a` for 3rd generation B). Then, the column's configured stratum retention policy is applied to "prune" the column by eliminating strata from specific generations. Although this cartoon depicts an empty space for eliminated strata, the underlying data structure behind a column (i.e., the pink overlay) can condense to reduce space complexity.

Num Gens Elapsed	Guaranteed MRCA-Recency-Proportional Resolution			
	1	4	10	100
1.0×10^{3}	18	26	41	80
1.0×10^{6}	32	50	85	184
1.0×10^{9}	51	79	134	293
1.0×10^{12}	64	102	177	396

Table 1: Number strata retained after one thousand, one million, one billion, and one trillion generations under the recency-proportional resolution stratum retention policy. Four different policy parameterizations are shown, the first where MRCA generation can be determined between two extant columns with a guaranteed relative error of 100%, the second 25%, the third 10%, and the fourth 1%. A column's memory footprint will be a constant factor of these retained counts based on the fingerprint differentia width chosen. For example, if single byte differentia were used, the column's memory footprint in bits would be $8\times$ the number of strata retained.

By design, this policy has the same space complexity and MRCA estimation uncertainty scaling relationships with number generations elapsed as the naive depth-proporitonal resolution policy.

MRCA-Recency-Proportional Resolution Stratum Retention Policy

The MRCA-recency-proportional resolution policy ensures distance between the retained strata surrounding any generation point will be less than or equal to a user-specified proportion $1/r$ of the number of generations elapsed since that generation.

This policy can be constructed recursively. So, to begin, let's consider setting up just the *first* generation g of the stratum after the root ancestor we will retain when n generations have elapsed. A simple geometric analysis reveals that providing the guaranteed resolution for the worst-case generation within the window between generation 0 and generation g (i.e., generation $g-1$) requires

$$g \leq \lfloor n/(r+1) \rfloor.$$

We now have an upper bound for the generation of the first stratum generation we must retain. However, we must guarantee that strata at these generations are actually available for us to retain (i.e., haven't been purged out of the column at a previous time point). We will do this by picking the generation that is the highest power of 2 less than or equal to our bound. If we repeat this procedure as we recurse, we are guaranteed that this generation's stratum will have been preserved across all previous timepoints.

Why does this work? Consider a sequence where all elements are spaced out by strictly nonincreasing powers of 2. Consider the first element of the list. All multiples this first element will be included in the list. So, when we ratchet up g to $2g$ as n increases, we are guaranteed that $2g$ has been retained. This principle generalizes recursively down the list. This is a similar principle to the approach of strictly-doubling interval sizes used in the Depth-Proportional Resolution stratum retention policies described above.

This step of truncating to the nearest less than or equal to power of 2 affects our recursive step size is at most halved. So, because step size is a constant fraction of remaining generations n (at worst $\frac{n}{2(r+1)}$), the number of steps made (and number of strata retained) scales as $O(\log(n))$ with respect to the number of strata deposited. Table 1 provides exact figures for the number of strata retained under different parameterizations of the recency-proportional retention policy between one thousand and one trillion generations.

As for MRCA generation estimate uncertainty, in the worst case it scales as $O(n)$ with respect to the greater number of strata deposited. However, with respect to estimating the generation of the MRCA for lineages diverged any fixed number of generations ago, uncertainty scales as $O(1)$.

How does space complexity scale with respect to the policy's specified resolution r? Through extrapolation from OEIS sequences A063787 and A056791 via guess and check (OEIS, 2021b,a), we posited the exact number of strata retained after n generations as

$$\text{HammingWeight}(n) + \sum_{1}^{r} \lfloor \log_2(\lfloor n/r \rfloor) \rfloor + 1.$$

This expression has been unit tested extensively to ensure perfect reliability. Approximating and applying logarithmic properties, this policy's space complexity can be calculated within a constant factor as

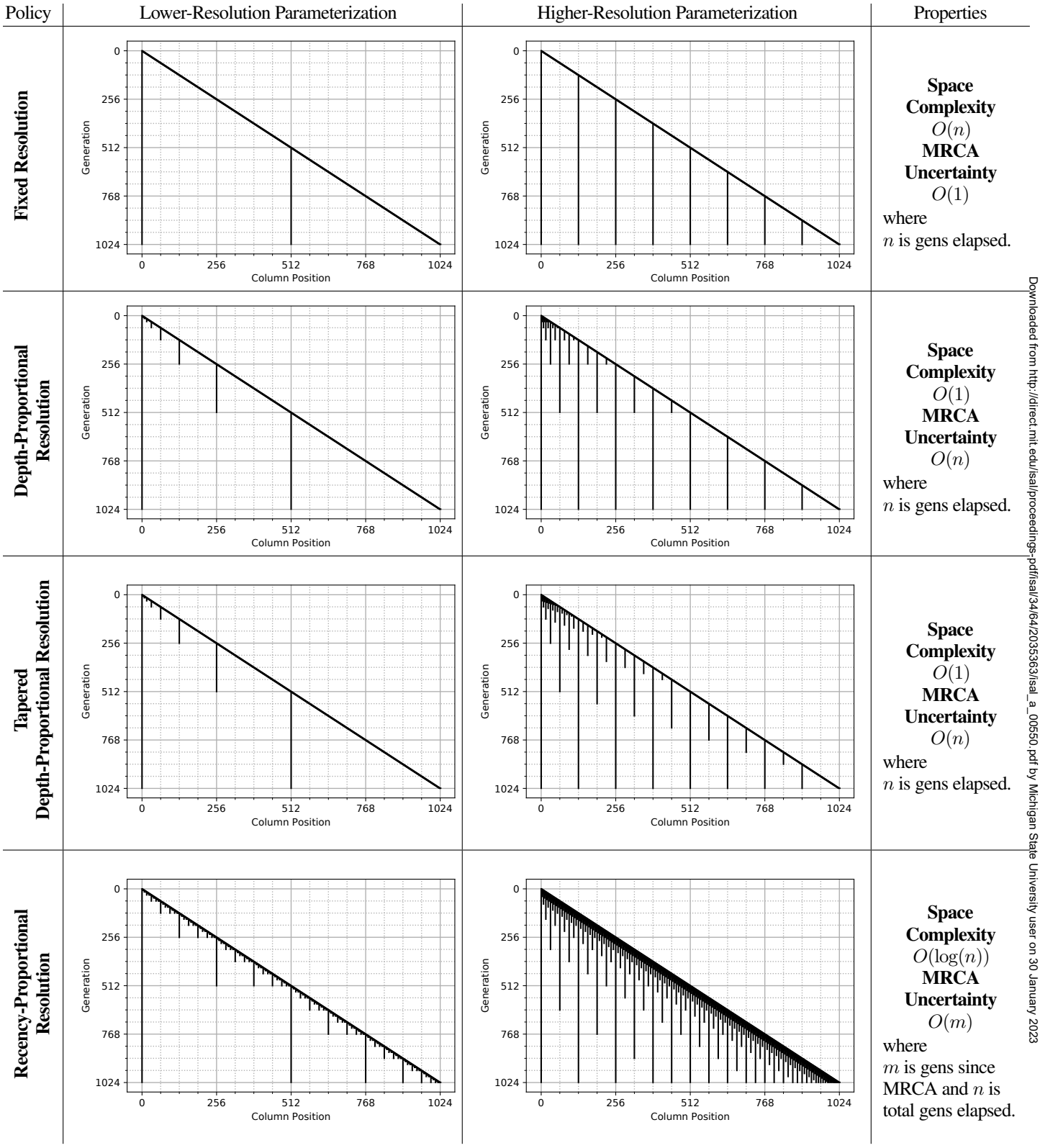


Figure 3: Comparison of stratum retention policies. Policy visualizations show retained strata in black. Time progresses along the y-axis from top to bottom. New strata are introduced along the diagonal and then "drip" downward as a vertical line until eliminated. The set of retained strata present within a column at a particular generation g can be read as intersections of retained vertical lines with a horizontal line with intercept g. Policy visualizations are provided for two parameterizations for each policy: the first where the maximum uncertainty of MRCA generation estimates would be 512 generations and the second where the maximum uncertainty of MRCA generation estimates would be 128 generations.

$$\log(n)+\log\left(\frac{n^r}{r!}\right).$$

To analyze the relationship between space complexity and resolution r, we will examine the ratio of space complexities induced when scaling resolution r up by a constant factor $f>1$. Evaluating this ratio as $r\to\infty$, we find that space complexity scales directly proportional to f,

$$\lim_{r\to\infty}\frac{\log(n)+\log\left(\frac{n^{fr}}{(fr)!}\right)}{\log(n)+\log\left(\frac{n^r}{r!}\right)}=f.$$

Evaluating this ratio as $n\to\infty$, we find that this scaling relationship is never worse than directly proportional for any r,

$$\begin{aligned}\lim_{n\to\infty}\frac{\log(n)+\log\left(\frac{n^{fr}}{(fr)!}\right)}{\log(n)+\log\left(\frac{n^r}{r!}\right)}&=\frac{fr+1}{r+1}\\&=f\frac{r+1/f}{r+1}\\&\leq f.\end{aligned}$$

Computational Experiments

In order to assess the practical performance of the hereditary stratigraph approach in an applied setting, we simulated the process of stratigraph propagation over known "ground truth" phylogenies extracted from pre-existing digital evolution simulations (Hernandez et al., 2022). These simulations propagated populations of between 100 and 165 bitstrings between 500 and 5,000 synchronous generations under the NK fitness landscape model (Kauffman and Weinberger, 1989). In order to ensure coverage of a variety of phylogenetic conditions, we sampled a variety of selection schemes that impose profoundly different ecological regimens (Dolson and Ofria, 2018),

- EcoEA Selection (Goings et al., 2012),
- Lexicase Selection (Helmuth et al., 2014),
- Random Selection, and
- Sharing Selection (Goldberg et al., 1987).

Supplementary Table 7 provides full details on the conditions each ground truth phylogeny was drawn from. The phylogenies themselves are available with our supplementary material (Moreno et al., 2022).

For each ground truth phylogeny, we tested combinations of three configuration parameters:

- target end-state memory footprints for extant columns (64, 512, and 4096 bits),
- differentia width (1, 8, and 64 bits), and
- stratum retention policy (tapered depth-proportional resolution and recency-proportional resolution).

Stratum retention policies were parameterized so that the maximum number of strata possible were present at the end of the experiment without exceeding the target memory footprint. If the target memory footprint is exceeded by the sparsest possible parameterization of a retention policy, then that sparsest possible parameterization was used. Supplementary Tables tables 2 to 6 provide the calculated paramaterizations and memory footprints of extant columns (Moreno et al., 2022).

In order to assess the viability of phylogenetic inference using hereditary stratigraphic columns from extant organisms, we used the end-state stratigraphs to reconstruct an estimate of the actual ground truth phylogenetic histories. The first step to reconstructing a phylogenetic tree for the history of an extant population at the end of an experiment is to construct a distance matrix by calculating all pairwise phylogenetic distances between extant columns. We defined phylogenetic distance between two extant columns as the sum of each extant organism's generational distance back to the generation of their MRCA, estimated as the mean of the upper and lower 95% confidence bounds. Supplementary Figure 6 provides a cartoon summary of the process of calculating phylogenetic distance between two extant columns (Moreno et al., 2022).

We then used the unweighted pair group method with arithmetic mean (UPGMA) reconstruction tool provided by the BioPython package to generate estimate phylogenetic trees (Cock et al., 2009; Sokal, 1958). After generating the reconstructed tree topology, we performed a second pass to adjust branch lengths so that each internal tree node sat at the mean of its estimated 95% confidence generation bounds.

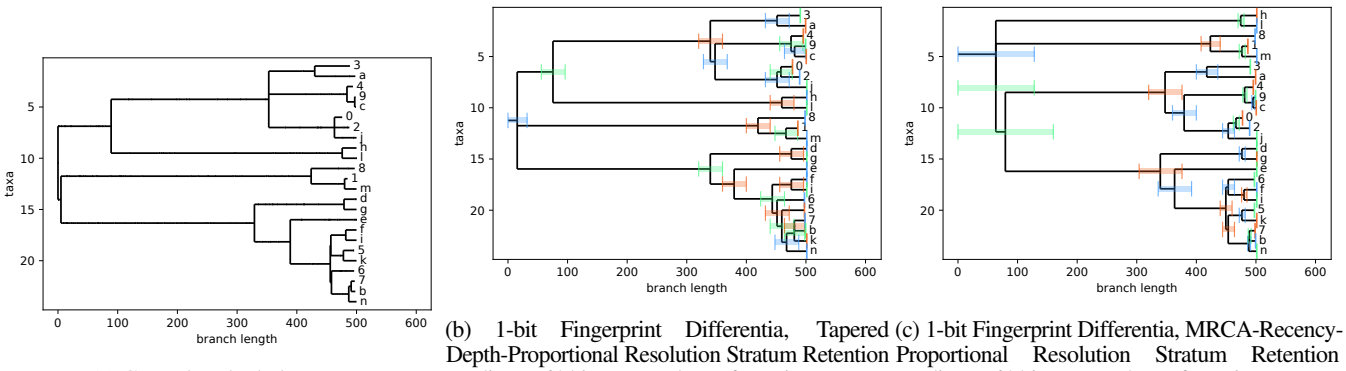
Software and Data

As part of this work, we published the `hstrat` Python library with a stable public-facing API intended to enable incorporation in other projects with extensive documentation and unit testing on GitHub at `https://github.com/mmore500/hstrat` and on PyPI.In the near future, we intend to complete and publish a corresponding C++ library.

Supporting software materials can be found on GitHub at `https://github.com/mmore500/hereditary-stratigraph-concept` Supporting computational notebooks are available for in-browser use via BinderHub at `https://hopth.ru/bk` (Ragan-Kelley and Willing, 2018). Our work benefited from many pieces of open source scientific software (Sukumaran and Holder, 2010; Virtanen et al., 2020; Hunter, 2007; Virtanen et al., 2020; Waskom, 2021; Bostock et al., 2011; Meurer et al., 2017; Smith, 2020b; Paradis et al., 2004; Ushey et al., 2022; Wickham et al., 2022). The ground truth phylogenies used in this work as well as supplementary figures, tables, and text are available via the Open Science Framework at `https://osf.io/4sm72/` (Foster and Deardorff, 2017; Moreno et al., 2022). Phylogenetic data associated with this project is stored in the Alife Community Data Standards format (Lalejini et al., 2019).

Results and Discussion

In this section, we analyze the quality of reconstructions of known phylogenetic trees using hereditary stratigraphy. Figure 4 compares an example reconstruction from columns using tapered depth-proportional stratum retention, an example reconstruction using recency-proportional stratum retention, and the underlying ground truth phylogeny. Interactive in-browser visualizations comparing all reconstructed phylogenies to their corresponding ground truth are available at `https://hopth.ru/bi`.



(a) Ground truth phylogeny.

(b) 1-bit Fingerprint Differentia, Tapered Depth-Proportional Resolution Stratum Retention Predicate, 64 bit target column footprint.

(c) 1-bit Fingerprint Differentia, MRCA-Recency-Proportional Resolution Stratum Retention Predicate, 64 bit target column footprint.

Figure 4: Example phylogeny reconstructions of ground-truth lexicase selection phylogeny from inference on extant hereditary stratigraphic columns. Shaded error bars on reconstructions indicate 95% confidence intervals for the true generation of tree nodes. Arbitrary color is added to enhance distinguishability.

Reconstruction Accuracy

Measuring tree similarity is a challenging problem, with many conflicting approaches that all provide different information (Smith, 2020a). Ideally, we would use a metric of reconstruction accuracy that 1) is commonly used so that there exists sufficient context to understand what constitutes a good value, 2) behaves consistently across different types of trees, and 3) behaves reasonably for the types of trees common in artificial life data. Unfortunately, these objectives are somewhat in conflict. The primary source of this problem is multifurcations, nodes from which more than two lineages branch at once. In reconstructed phylogenies in biology, multifurcations are generally assumed to be the result of insufficient information. It is thought that the real phylogeny had multiple bifurcations that occurred so close together that the reconstruction algorithm is unable to separate them. In artificial life phylogenies, however, we have the opposite problem. When we perfectly track a phylogeny, it is common for us to know that a multifurcation did in fact occur. However, it is challenging for our reconstructions to properly identify multifurcations, because it requires perfectly lining up multiple divergence times. Many of the most popular tree distance metrics interpret the difference between a multifurcation and a set of bifurcations as a dramatic change in topology. For some use cases, this change in topology may indeed be meaningful, although research on the extent of this problem is limited. Nevertheless, we suspect that for the majority of use cases, the tiny branch lengths between the internal nodes will make this source of error relatively minor.

To overcome this obstacle, we have measured our reconstruction accuracy using multiple metrics. We will primarily focus on Mutual Clustering Information (as implemented in the R TreeDist package) (Smith, 2020a), which is a direct measure of the quantity of information in the ground truth phylogeny that was successfully captured in the reconstruction. It is relatively unaffected by the failure to perfectly reproduce multifurcations. For the purposes of easy comparison to the literature, we also measured the Clustering Information Distance (Smith, 2020a).

Across ground truth phylogenies, we were able to reconstruct the phylogenetic topology with between 47.75% and 85.70% of the information contained in the original tree using a 64-bit column memory footprint, between 47.75% and 80.36% using a 512-bit column memory footprint, and between 51.13% and 83.53% using a 4096-bit column memory footprint. While the Clustering Information Distance reached its maximum possible score (1.0) for the heavily-multifurcated EcoEA phylogeny, it agreed with the Mutual Clustering Information score for less multifurcated phylogenies, such as fitness sharing. Using the Recency Proportional Resolution retention policy and a 4096-bit column memory footprint, we were able to reconstruct a fitness sharing phylogeny with a Clustering Information Distance of only 0.2923471 from the ground truth. For context, that result is comparable to the distance between phylogenies reconstructed from two closely-related proteins in H3N2 flu (0.25) (Jones et al., 2021). To build further intution, we strongly encourage readers to refer to our interactive web reconstruction. Figure 5 summarizes error reconstructing the fitness sharing selection phylogeny in terms of the mutual clustering information metric (Smith, 2022). The phylogenies reconstructed from the EcoEA condition performed comparably, with lexicase and random selection faring somewhat worse (Moreno et al., 2022). In the case of random selection, we suspect that this reduced

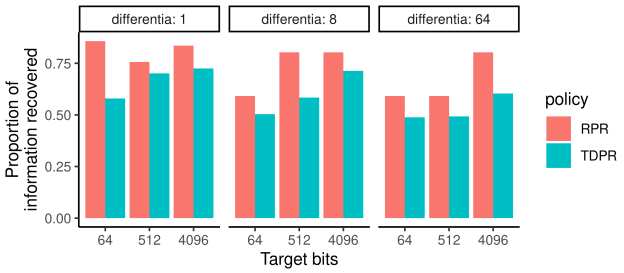


Figure 5: Proportion of information present in the ground-truth ftness sharing phylogeny that was captureed by our reconstruction, across various retention policies. High is better (1 is perfect). RPR is recency-proportional resolution policy and TDPR is tapered depth-proportional resolution policy.

performance is the result of having many nodes that originated very close together at the end of the experiment. As expected, we did observe overall more accurate reconstructions from columns that were allowed to occupy larger memory footprints.

Differentia Size

Among the surveyed ground truth phylogenies and target column footprints, we consistently found that smaller differentia were able to yield more or as accurate phylogenetic reconstructions. The stronger performance of narrow differentia was particularly apparent in low-memory-footprint scenarios where overall phylogenetic inference power was weaker. Overall, single-bit differentia outperformed 64-bit differentia under 20 condtions, and were indistinguishable under 7 conditions, and were worse under 3 conditions. Full results are available in Supplementary Section . Although narrower differentia have less distinguishing power on their own, their smaller size allows more to be packed into the memory footprint to cover more generations, which seems to help reconstruction power. We must note that narrower differentia can pack more thoroughly into the footprint caps we imposed on column size, so their extant columns tended to have slightly more overall bits. However, this was a small enough imbalance (in most cases $< 10\%$) that we believe it is unlikely to fully account for the stronger performance of narrow-differentia configurations.

Retention Policy

Across the surveyed ground truth phylogenies and target column memory footprints, we found that the recency-proportional resolution stratum retention policy generally yielded better phylogenetic reconstructions. Phylogenetic reconstruction quality was better in 28 conditions, equivalent in 14 conditions, and worse in 3 conditions. Again, this effect was most apparent in the small-stratum-count scenarios where overal inference power was weaker. Full results are available in Supplementary Section . The stronger performance of recency-proportional resolution is likely due to the denser retention of recent strata under the recency-proportional metric, which help to resolve the more numerous (and therefore typically more tightly spaced) phylogenetic events in the near past (Zhaxybayeva and Gogarten, 2004). Recency-proportional resolution tended to be able to fit fewer strata within the prescribed memory footprints (except in cases where it could not fit within the footprint) so its stronger performance cannot be attributed to more retained bits in the end-state extant columns.

Conclusion

To our knowledge, this work provides a novel design for digital genome components that enable phylogenetic inference on asexual populations. This provides a viable alternative to perfect phylogenetic tracking, which is complex and possibly cumbersome in distributed computing scenarios, especially with fallible nodes. Our approach enables flexible, explicit trade-offs between space complexity and inference accuracy. Hereditary stratigraphic columns are efficient: our approach can estimate, for example, the MRCA generation of two genomes within 10% error with 95% confidence up to a depth of a trillion generations with genome annotations smaller than a kilobyte. However, they are also powerful: we were able to achieve tree reconstructions recovering up to 85.70% of the information contained in the original tree with only a 64-bit memory footprint.

This and other methodology to enable decentralized observation and analysis of evolving systems will be essential for artificial life experiments that use distributed and best-effort computing approaches. Such systems will be crucial to enabling advances in the field of artificial life, particularly with respect to the question of open-ended evolution (Ackley and Cannon, 2011; Moreno et al., 2021b,a) Mork work is called for to further enable experimental analyses in distributed, best-effort systems while preserving those systems' efficiency and scalability. As parallel and distributed computing becomes increasingly ubiquitous and begins to more widely pervade artificial life systems, hereditary stratigraphy should serve as a useful technique in this toolbox.

Important work extending and analyzing hereditary stratigraphy remains to be done. Analyses should be performed to expound MRCA resolution guarantees of stratum retention policies when using narrow (i.e., single-bit) differentia. Constant-size-complexity stratum retention policies that preferentially retain a denser sampling of more-recent strata should be developed and analyzed. Extensions to sexual populations should be explored, including the possibility of annotating and tracking individual genome components instead of whole-genome individuals. An alternate approach might be to define a preferential inheritance rule so that at each generation slot within a column, a single differentia sweeps over an entire interbreeding population. Optimization of tree reconstruction from extant hereditary stratigraphs remains an open question, too, particularly with regard to properly handling multifurcations. It would be particularly valuable to develop methodology to annotate inner nodes of trees reconstructed from hereditary stratigraphs with confidence levels.

The problem of designing genomes to maximize phylogenetic reconstructability raises unique questions about phylogenetic estimation. Such a backward problem — optimizing genomes to make analyses trivial as opposed to the usual process of optimizing analyses to genomes — puts questions about the genetic information analyses operate on in a new light. In particular, it would be interesting to derive upper bounds on phylogenetic inference accuracy given genome size and generations elapsed.

Acknowledgment

This research was supported in part by NSF grants DEB-1655715 and DBI-0939454 as well as by Michigan State University through the computational resources provided by the Institute for Cyber-Enabled Research. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1424871. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Thank you to Santiago Rodriguez Papa for manuscript comments.

References

Ackley, D. H. and Cannon, D. C. (2011). Pursue robust indefinite scalability. In *13th Workshop on Hot Topics in Operating Systems (HotOS XIII)*.

Bohm, C., Hintze, A., et al. (2017). Mabe (modular agent based evolver): A framework for digital evolution research. In *ECAL 2017, the Fourteenth European Conference on Artificial Life*, pages 76–83. MIT Press.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.

Casci, T. (2008). Lining up is hard to do. *Nature Reviews Genetics*, 9(8):573–573.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

Dolson, E., Lalejini, A., Jorgensen, S., and Ofria, C. (2020). Interpreting the tape of life: Ancestry-based analyses provide insights and intuition about evolutionary dynamics. *Artificial Life*, 26(1):58–79.

Dolson, E. and Ofria, C. (2018). Ecological theory provides insights about evolutionary computation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 105–106.

Dolson, E. L., Vostinar, A. E., Wiser, M. J., and Ofria, C. (2019). The modes toolbox: Measurements of open-ended dynamics in evolving systems. *Artificial Life*, 25(1):50–73.

Foster, E. D. and Deardorff, A. (2017). Open science framework (osf). *Journal of the Medical Library Association: JMLA*, 105(2):203.

Goings, S., Goldsby, H., Cheng, B. H., and Ofria, C. (2012). An ecology-based evolutionary algorithm to evolve solutions to complex problems. In *ALIFE 2012: The Thirteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 171–177. MIT Press.

Goldberg, D. E., Richardson, J., et al. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, volume 4149. Hillsdale, NJ: Lawrence Erlbaum.

Hagstrom, G. I., Hang, D. H., Ofria, C., and Torng, E. (2004). Using avida to test the effects of natural selection on phylogenetic reconstruction methods. *Artificial life*, 10(2):157–166.

Helmuth, T., Spector, L., and Matheson, J. (2014). Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*, 19(5):630–643.

Hernandez, J. G., Lalejini, A., and Dolson, E. (2022). What Can Phylogenetic Metrics Tell us About Useful Diversity in Evolutionary Algorithms? In Banzhaf, W., Trujillo, L., Winkler, S., and Worzel, B., editors, *Genetic Programming Theory and Practice XVIII*, pages 63–82. Springer Singapore, Singapore.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jones, J. E., Le Sage, V., Padovani, G. H., Calderon, M., Wright, E. S., and Lakdawala, S. S. (2021). Parallel evolution between genomic segments of seasonal human influenza viruses reveals rna-rna relationships. *Elife*, 10:e66525.

Kapli, P., Yang, Z., and Telford, M. J. (2020). Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, 21(7):428–444.

Kauffman, S. A. and Weinberger, E. D. (1989). The nk model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology*, 141(2):211–245.

Lack, J. B. and Van Den Bussche, R. A. (2010). Identifying the confounding factors in resolving phylogenetic relationships in vespertilionidae. *Journal of Mammalogy*, 91(6):1435–1448.

Lalejini, A., Dolson, E., Bohm, C., Ferguson, A. J., Parsons, D. P., Rainford, P. F., Richmond, P., and Ofria, C. (2019). Data standards for artificial life software. In *ALIFE 2019: The 2019 Conference on Artificial Life*, pages 507–514. MIT Press.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Moreno, M. A., Dolson, E., and Ofria, C. (2022). Hereditary stratigraph concept supplement. Available at https://doi.org/10.17605/osf.io/4sm72.

Moreno, M. A., Papa, S. R., and Ofria, C. (2021a). Case study of novelty, complexity, and adaptation in a multicellular system. In *OEE4: The Fourth Workshop on Open-Ended Evolution*.

Moreno, M. A., Papa, S. R., and Ofria, C. (2021b). Conduit: a c++ library for best-effort high performance computing. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1795–1800.

OEIS (2021a). Sequence a056791. The on-line encyclopedia of integer sequences. Available at https://oeis.org/A056791.

OEIS (2021b). Sequence a063787. The on-line encyclopedia of integer sequences. Available at https://oeis.org/A063787.

Paradis, E., Claude, J., and Strimmer, K. (2004). Ape: analyses of phylogenetics and evolution in r language. *Bioinformatics*, 20(2):289–290.

Ragan-Kelley, B. and Willing, C. (2018). Binder 2.0-reproducible, interactive, sharable environments for science at scale. In *Proceedings of the 17th Python in Science Conference (F. Akici, D. Lippa, D. Niederhut, and M. Pacer, eds.)*, pages 113–120.

Sarkar, S., Majumder, T., Kalyanaraman, A., and Pande, P. P. (2010). Hardware accelerators for biocomputing: A survey. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 3789–3792. Ieee.

Smith, M. R. (2020a). Information theoretic generalized robinson-foulds metrics for comparing phylogenetic trees. *Bioinformatics*, 36(20):5007–5013.

Smith, M. R. (2020b). ms609/treedistdata: v1.0.0.

Smith, M. R. (2022). Robust analysis of phylogenetic tree space. *Systematic Biology*.

Sokal, R. R. (1958). A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.

Steno, N. (1916). *The prodromus of Nicolaus Steno's dissertation concerning a solid body enclosed by process of nature within a solid*, volume 11. University of Michigan Press.

Sukumaran, J. and Holder, M. T. (2010). Dendropy: a python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571.

Ushey, K., Allaire, J., and Tang, Y. (2022). *reticulate: Interface to 'Python'*. https://rstudio.github.io/reticulate/, https://github.com/rstudio/reticulate.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wang, R., Clune, J., and Stanley, K. O. (2018). Vine: an open source interactive data visualization tool for neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1562–1564.

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wickham, H., François, R., Henry, L., and Müller, K. (2022). *dplyr: A Grammar of Data Manipulation*. https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

Zhaxybayeva, O. and Gogarten, J. P. (2004). Cladogenesis, coalescence and the evolution of the three domains of life. *TRENDS in Genetics*, 20(4):182–187.