

Case Study of Novelty, Complexity, and Adaptation in a Multicellular System

Matthew Andres Moreno¹, Santiago Rodriguez Papa¹ and Charles Ofria¹

¹Michigan State University, East Lansing, MI 48103
mmore500@msu.edu

Abstract

Continuing generation of novelty, complexity, and adaptation are well-established as core aspects of open-ended evolution. However, the manner in which these phenomena relate remains an area of great theoretical interest. It is yet to be firmly established to what extent these phenomena are coupled and by what means they interact. In this work, we track the co-evolution of novelty, complexity, and adaptation in a case study from a simulation system designed to study the evolution of digital multicellularity. In this case study, we describe ten qualitatively distinct multicellular morphologies, several of which exhibit asymmetrical growth and distinct life stages. We contextualize the evolutionary history of these morphologies with measurements of complexity and adaptation. Our case study suggests a loose, sometimes divergent, relationship can exist among novelty, complexity, and adaptation.

Introduction

The challenge, and promise, of open-ended evolution has animated decades of inquiry and discussion within the artificial life community (Packard et al., 2019). The difficulty of devising models that produce characteristic outcomes of open-ended evolution suggests profound philosophical or scientific blind spots in our understanding of the natural processes that gave rise to contemporary organisms and ecosystems. Already, pursuit of open-ended evolution has yielded paradigm-shifting insights. For example, novelty search demonstrated how processes promoting non-adaptive diversification can ultimately yield adaptive outcomes that were previously unattainable (Lehman and Stanley, 2011). Such work lends insight to fundamental questions in evolutionary biology, such as the relevance — or irrelevance — of natural selection with respect to increases in complexity (Lehman, 2012; Lynch, 2007) and the origins of evolvability (Lehman and Stanley, 2013; Kirschner and Gerhart, 1998). Evolutionary algorithms devised in support of open-ended evolution models also promise to deliver tangible broader impacts for society. Possibilities include the generative design of engineering solutions, consumer products, art, video games, and AI systems (Nguyen et al., 2015; Stanley et al., 2017).

Preceding decades have witnessed advances toward defining — quantitatively and philosophically — the concept of open-ended evolution (Lehman and Stanley, 2012; Dolson et al., 2019; Bedau et al., 1998) as well as investigating causal phenomena that promote open-ended dynamics such as ecological dynamics, selection, and evolvability (Dolson, 2019; Soros and Stanley,

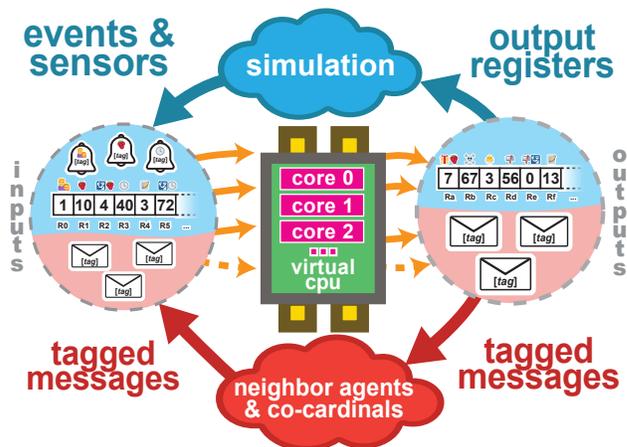


Figure 1: Overview of genome execution. Tagged events and messages (shown as bells and envelopes, respectively) activate module execution on virtual cores. Simulation state can also be read directly using sensor instructions to access input registers. Special instructions write to output registers, allowing interaction with the simulation, and generate tagged messages, allowing interaction with other virtual CPUs.

2014; Huizinga et al., 2018). The concept of open-endedness is fundamentally characterized by intertwined generation of novelty, functional complexity, and adaptation (Taylor et al., 2016). The mechanisms by which, and the extent to which these phenomena relate to one another remains an open question. Here, we aim to complement ongoing work to develop a firmer theoretical understanding of the relationship between novelty, complexity, and adaptation by exploring the evolution of these phenomena through a case study using the DISHTINY digital multicellularity framework (Moreno and Ofria, 2019). We apply existing methods and philosophy developed to describe novelty, complexity, and adaptation to assess how these qualities can change over evolutionary time and in relation to one another.

Methods

Simulation

The DISHTINY simulation environment tracks cells occupying tiles on a 120×120 toroidal grid. Cells collect a uniform inflow of continuous-valued resource. This resource can be spent in increments of 1.0 to attempt asexual reproduction into any of

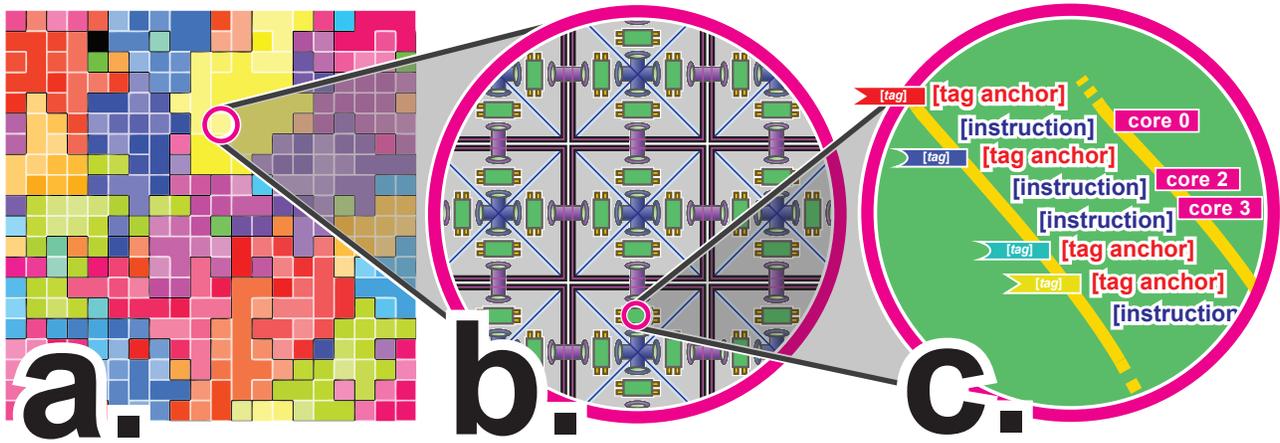


Figure 2: Overview of DISHTINY system. Cells occupy slots on a toridal grid (Subfigure *a*). As cells reproduce, they may grow their existing kin group (shown here by color) or splinter off to found new ones. Each cell, shown here bounded within black squares, is controlled by four virtual CPUs, referred to as “cardinals” and shown here within triangles (Subfigure *b*). Cardinals within a cell can interact via message passing (blue conduits). Cardinals can interact with the corresponding cardinal in their neighboring cell through message passing or simulation intrinsics (i.e., resource sharing, offspring spawning, etc.), represented here by purple conduits. These inter-cell interactions may span physical hardware threads or processes. All virtual CPUs within a cell independently execute the same linear genetic program (Subfigure *c*). Tagged subsections of this linear genetic program (“modules”) activate in response to stimuli.

a cell’s four adjacent cells. A cell can only be replaced if it commands less than 1.0 resource. If a cell rebuffs a reproduction attempt, its resource stockpile decrements by 1.0.

In order to facilitate the formation of coherent multicellular groups, the DISHTINY framework provides a mechanism for cells to form groups and detect group membership (Moreno and Ofria, 2019). Groups arise through cellular reproduction. When a cell proliferates, it may choose to initiate its offspring as a member of its kin group, thereby growing it, or induce the offspring to found a new kin group. This process is similar to the growth of biological multicellular tissues, where cell offspring can be retained as members of the tissue or permanently expelled.

We incentivize group formation by providing an additional resource inflow bonus based on group size. Groups that are too small do not receive this bonus. Group that are too large receive a penalty. In order to ensure group turnover, we force groups to fragment into unicells after 8,192 updates.

In previous work, we established that this framework can select for traits characteristic of multicellularity, such as cooperation, coordination, and reproductive division of labor Moreno and Ofria (2021). We also found more case studies of interest arose when two nested levels of group membership were tracked as opposed to a single, un-nested level of group membership Moreno and Ofria (2021). With nested group membership, group growth still occurs by cellular reproduction. However, cells are given the choice to expel offspring from the innermost group (but retain membership in the outermost group) in addition to choosing to retain offspring within both groups or to expel offspring from both groups. In this work, we allow for nested kin groups.

In addition to controlling reproduction behavior, evolving genomes can also share resources with adjacent cells, perform apoptosis

(recovering a small amount of resource that may be shared with neighboring cells), and pass arbitrary messages to neighboring cells. Cell behaviors are controlled by event-driven genetic programs in which linear GP modules are activated in response to cues from the environment or neighboring agents; signals are handled in quasi-parallel on up to 32 virtual cores (Figure 1) (Lalejini and Ofria, 2018). Each cell contains four independent virtual CPUs, all of which execute the same genetic program (Figure 2a). Each CPU manages interactions with a single neighboring cell. We refer to a CPU managing interactions with a particular neighbor as a “cardinal” (as in “cardinal direction”). These CPUs may communicate via intra-cellular message passing. Full details on the instruction set and event library used as well as simulation logic and parameter settings appear in supplementary material (Moreno, 2021).

Evolution

We performed evolution in three hour windows for compatibility with our compute cluster’s scheduling system. We refer to these windows as “stints.” One-hundred instruction genomes were randomly generated at the outset of the initial stint, stint 0. At the end of each three hour window, genomes were harvested and stored in a population file. Subsequent stints were then seeded with the previous stint’s population. No simulation state besides genome content was preserved between stints.

In order to ensure heterogeneity of biotic environmental factors experienced by evolving cells, we imposed a diversity maintenance scheme. In this scheme, descendants of a single progenitor cell from stint 0 that proliferated to constitute more than half of the population were penalized with resource loss. The severity of the penalty increased with increasing prevalence beyond half of the population. Thus, we ensured that descendants from at least two distinct stint 0 progenitors remained over the course of the simulation.

In our screen for case studies, we evolved 40 independent populations for 101 stints. We selected population 16005 from among these 40 to profile as a case study due to its distinct asymmetrical group morphology.

At the conclusion of each stint, we selected the most abundant genome within the population as a representative specimen. We performed a suite of follow-up analyses on each representative specimen to characterize aspects of complexity, detailed in the following subsections. To ensure that specimens were consistently sampled from descendants of the same stint 0 progenitor, we only considered genomes with the lowest available stint 0 progenitor ID.

Phenotype-neutral Nopout

After harvesting representative specimens from each stint, we began filtered out genome instructions that had no impact on the simulation.

To accomplish this, we performed sequential single-site “nopouts” where individual genome instructions were disabled by replaced with a `Nop` instruction.¹ We reverted nopouts that altered a strain’s phenotype and kept those that did not. To determine whether phenotypic alteration occurred, we seeded an independent, mutation-disabled simulation with the stain in question and ran it side-by-side with an independent, mutation-disabled simulation of the wildtype strain. If any divergence in resource concentration was detected between the two strains within a 2,048 update window, the single site nopout was reverted. We continued this process until no single-site nopouts were possible without altering the genome’s phenotype. To speed up evaluation, we performed step-by-step, side-by-side comparisons using a smaller toroidal grid size of just 100 tiles.

This process left us with a “Phenoytpe-neutral Nopout” variant of the wildtype genome where all remaining instructions contributed to the phenotype.

However, in further analyses we discovered that 21 phenotype-neutral nopouts from our case study were *not* actually neutral — competition experiments revealed they were significantly less fit than the wildtype strain. This might be due to insufficient spatial or temporal scope to observe expression of particular genome sites in our test for phenotypic divergence.

Estimating Critical Fitness Complexity

Next, we sought to detect genome instructions that contributed to a strain’s fitness.

For each remaining op instruction in the Phenotype-neutral Nopout variant, we took the wildtype strain and applied a nopout at the corresponding site. We then competed this variant against the wildtype strain. Only evaluating remaining op instructions in the Phenotype-neutral Nopout variant allowed us to decrease the number of fitness competitions we had to perform.

¹This `Nop` instruction was chosen to perform the same number of random number generator touches as the original instruction to control for arbitrary effects of advancing the generator.

Fitness competitions began by seeding a population half-and-half with two strains. These competitions ran for 10 minutes (about 4,200 updates) on a 60×60 toridal grid after which the simulation was ended and the relative abundances of descendants of both seeded strains were assessed.

To determine whether fitness differed significantly between a wildtype and variant strain, we compared the relative abundance of the strains observed at the end of competitions against outcomes from 20 control wildtype-vs-wildtype competitions. We fit a T -distribution to the abundance outcomes observed under the control wildtype-vs-wildtype competitions and deemed outcomes that fell outside the central 98% probability density of that distribution a significant difference in fitness. This allowed us to screen for fitness effects of single-site nopouts while only performing a single competition per site.

This process left us with a “Fitness-noncritical Nopout” variant of the wildtype genome where all remaining instructions contributed to the phenotype. We called the number of remaining instructions its “critical fitness complexity.” We adjusted this figure downwards for the expected 1% rate of false-positive fitness differences among tested genome sites. This metric mirrors the MODES complexity metric described in (Dolson et al., 2019) and the approximation of sequence complexity advanced in (Adami et al., 2000).

Estimating State Interface Complexity

In addition to estimating the number of genome sites that contribute to fitness, we were interested in measuring the number of different environmental cues and the number of different output mechanisms that cells adaptively incorporated into behavior.

One possible way to take this measure would be to disable event cues, sensor instructions, and output registers one by one and test for changes in fitness. However, this approach would fail to distinguish context-dependent input/output from merely contingent input/output. For example, a cell might happen to depend on a sensor being set at a certain frequency but not on the actual underlying simulation information the sensor represents.

To isolate context-dependent input/output state interactions, we tested the fitness effect of swapping particular input/output states between CPUs rather than completely disabling them. That is, for example, CPU b would be forced to perform the output generated by CPU a or CPU b would be shown the input meant for CPU a . We performed this manipulation on half the population in a fitness competition for each individual component of the simulation’s introspective state (44 sensor states relating to the status of a CPU’s own cell), extrospective state (61 sensor states relating to the status of a neighboring cell), and writable state (18 output states, 10 of which control cell behavior and 8 of which act as global memory for the CPU).² We deemed a state as fitness-critical if this manipulation resulted in decreased fitness at significance $p < 0.01$ using a T -test parameterized by 20 control wild-type vs wild-type competitions.

²A full description of each piece of introspective, extrospective, and writable state is listed in supplementary material (Moreno, 2021).

We describe the number of states that cells interact with to contribute to fitness as “State Interface Complexity.”

Estimating Messaging Interface Complexity

In addition to estimating the number of input/output states cells use to interact with the environment, we were interested in estimating the number of distinct intra-cellular messages cardinals within a cell use to coordinate and inter-cellular messages that cells use to coordinate. As with state interface complexity, distinguishing context-dependent behavior from contingent behavior is critical to attaining a meaningful measurement. For example, a cardinal might happen to depend on always receiving a inter-cellular message from a neighbor or an intra-cellular message from another cardinal. Although meaningless, if that message were blocked fitness would decrease. So, instead of simply discarding messages to test for a fitness effect, we re-route messages back to the sending cardinal instead of their intended recipient. We deemed a messages as fitness-critical if this manipulation resulted in decreased fitness at significance $p < 0.01$ using a T -test parameterized by 20 control wild-type vs wild-type competitions.

We refer to the number of distinct messages that cells send to contribute to fitness as “Messaging Interface Complexity.”

We refer to the sum of State Interface Complexity, Intra-messaging Interface Complexity, and Inter-messaging Interface Complexity as “Cardinal Interface Complexity.”

Implementation

Multithreading was employed to speed up execution. We broke the simulation into four 60×60 subgrids. Each subgrid executed asynchronously, using the Conduit C++ Library to orchestrate best-effort, real-time interactions between simulation elements on different threads. This approach is inspired by Ackley’s notion of indefinite scalability (Ackley and Small, 2014). In other work benchmarking the system, we have demonstrated that this approach improves scalability. The simulation scales to 4 threads with 80% efficiency, up to 64 threads with 40% efficiency and up to 64 nodes with 80% efficiency (Moreno et al., 2021).

Over the 101 three-hour evolutionary stints performed to evolve the case study, 7,565,309 simulation updates elapsed. (This translates to 74,904 updates elapsed per stint or about 6.9 updates per second.) However, the update processing rate was not uniform across stints: the simulation slowed about 77% as stints progressed. Supplementary Figure 14 shows elapsed updates for each stint (Moreno, 2021). During stint 0, 176,816 updates elapsed (about 16.3 updates per second). During stint 100, only 41,920 updates elapsed (about 3.8 updates per second).

Although working asynchronously, threads processed similar number of updates during each stint. The mean standard deviation of update-processing rate between threads was 2%. The mean difference of the update-processing rate between the fastest and slowest threads was 5%. The maximum value of these statistics observed during a stint was 9% and 20%, respectively, at stint 44. Supplementary Figure 14b shows the distribution of elapsed updates across threads for each stint evolved during the case study (Moreno, 2021).

Software is available under a MIT License at <https://github.com/mmoro500/dishtiny>. All data is available via the Open Science Framework at <https://osf.io/prq49>. Supplementary material is available via the Open Science Framework at <https://osf.io/gekc8> (Moreno, 2021).

Results

Evolutionary History

Due to the distributed nature of the experimental framework, we did not perform perfect phylogeny tracking. However, we did track the total number of ancestors seeded into stint 0 with extant descendants. At the end of stints 0 and 1, three distinct original phylogenetic roots were present in the population. From stint 2 onward, only two distinct original phylogenetic roots were present.

We performed follow-up analyses on specimens sampled from the lowest original phylogenetic root ID present in the population. For the first two stints, this was root ID 2,378. During stint 2, original phylogenetic root 2,378 went extinct. So, all further follow-up analyses were sampled from descendants of ancestor 12,634.

We also tracked the number of genomes reconstituted at the outset of each stint with extant descendants at the end of that stint. This count grows from approximately 10 around stint 15 to upwards of 30 around stint 40 (Supplementary Figure 10a (Moreno, 2021)). Among descendants of the lowest original phylogenetic root, the number of independent lineages spanning a stint also increases from around 5 to around 15 (Supplementary Figure 10b (Moreno, 2021)). This decrease in phylogenetic consolidation on a stint-by-stint basis correlates with the waning number of simulation updates performed per stint (Supplementary Figures 10c and 10d (Moreno, 2021)). More complete phylogenetic data will be necessary in future experiments to address questions about the possibility of long-term stable coexistence beyond the two strains supported under the explicit diversity maintenance scheme.

On the specimen from stint 100 used in the final case study, an evolutionary history of 20,212 cell generations had elapsed. Of these cellular reproductions, 11,713 (58%) had full kin group commonality, 7,174 had partial kin group commonality (35%), and 1,325 had no kin group commonality (7%). On this specimen, 1,672 mutation events had elapsed. During these events, 7,240 insertion-deletion alterations had occurred and 26,153 point mutations had occurred. This strain experienced a selection pressure of 18% over its evolutionary history, meaning that only 82% of the mutations that would be expected given the number of cellular reproductions that had elapsed were present.

Qualitative Morphological Categorizations

We performed a qualitative survey of the evolved life histories along the evolutionary timeline by analyzing video recordings of monocultures of each stint’s representative specimen.

Table 1 summarizes the ten morphological categories we grouped specimens into. In brief, specimens from early stints largely grew as unicellular or small multicellular groups (morphs a , b). Then, the specimen from stint 14 grew as larger, symmetrical groups

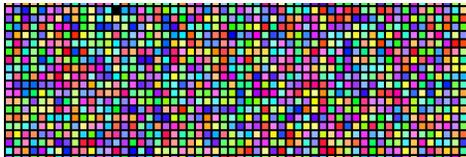
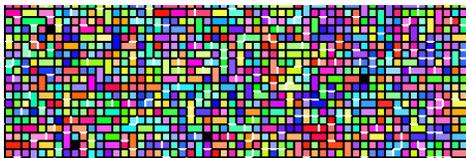
Morph	Description	Snapshot	Video
a	This morphology consists of individual cells, no multi-cellular kin groups. Resource use is low—most cells simply hoard resource until their stockpile is beyond sufficient to reproduce. Only a handful of cells intermittently expend resource.		https://hophth.ru/21/b=prq49+s=16005+t=0+v=video+w=specimen
b	This morphology consists of mostly individual cells, with some two-, three-, and four-cell groups evenly spread out. Resource usage occurs in short spurts in one or two adjacent cells.		https://hophth.ru/21/b=prq49+s=16005+t=1+v=video+w=specimen
c	Large multi-cellular groups dominate, consisting of hundreds of cells. Group growth is unchecked and continues until cells' resource stockpiles are entirely depleted by the excess group size penalty.		https://hophth.ru/21/b=prq49+s=16005+t=2+v=video+w=specimen
d	Clear groups of 10 to 15 cells in size form. Cell proliferation appears somewhat more active at the periphery of groups compared to the interior.		https://hophth.ru/21/b=prq49+s=16005+t=14+v=video+w=specimen
e	Groups are visibly elongated along the horizontal axis. After initial development, some gradual, irregular occurs along the vertical axis.		https://hophth.ru/21/b=prq49+s=16005+t=15+v=video+w=specimen
f	Groups are horizontally elongated similarly to morphology <i>e</i> , but have a greater consistent vertical thickness of three or four cells.		https://hophth.ru/21/b=prq49+s=16005+t=39+v=video+w=specimen
g	Initial group growth is almost entirely horizontal, with groups usually taking up only one row of cells. However, after an apparent timing cue groups switch to aggressive vertical growth.		https://hophth.ru/21/b=prq49+s=16005+t=45+v=video+w=specimen
h	Initial group growth is almost entirely horizontal, with groups usually taking up only one row of cells. However, after an apparent timing cue unrestrained vertical and horizontal cell proliferation occurs.		https://hophth.ru/21/b=prq49+s=16005+t=59+v=video+w=specimen
i	Irregular groups of mostly less than ten cells arise. Incessant proliferation with almost no resource retention leads to rapid group turnover.		https://hophth.ru/21/b=prq49+s=16005+t=74+v=video+w=specimen
j	Groups grow horizontally and then proliferate vertically on a timing cue like morph <i>e</i> . However, several viable horizontal-bar offspring groups form before force-fragmentation.		https://hophth.ru/21/b=prq49+s=16005+t=100+v=video+w=specimen

Table 1: Qualitative morph phenotype categorizations. Color coding of morph IDs has no significance beyond guiding the eye in scatter plots where points are labeled by morph. Snapshot visualizes spatial layout of kin groups on toroidal grid at a fixed point in time. Each cell corresponds to a small square tile. Color hue denotes and black borders divide outermost kin groups while color saturation denotes and white borders divide innermost kin groups.

(morph *d*). At stint 15, a distinct, asymmetrical horizontal bar morphology evolved (morph *e*). At stint 45, a delayed secondary spurt of group growth in the vertical direction arose (morph *g*). This morphology was sampled frequently until stint 60 when morph *e* began to be sampled primarily again. However, morph *g* was observed as late as stint 90.

Table 1 provides more detailed descriptions of each qualitative morph category as well as video and a still image example of each. Supplementary Table 2 provides morph categorization for each stint as well as links to view the stint’s specimen in a video or in-browser web simulation (Moreno, 2021).

Fitness

In order to assess ongoing changes in fitness, we performed 20 replicate fitness competitions between the population seeded into each stint and the immediately population preceding it one stint prior. We determined that a significant change in fitness had occurred between populations if one population won more than 15 of those competitions, corresponding to a significance level of $p < 0.05$ under the binomial null hypothesis. Figure 3a shows the fraction of competitions each stint won against its competitor. Significant increases in fitness occur throughout the evolutionary history of the case study, but not at every stint. In fact, some 20 stints exhibit significantly *worse* fitness compared to their predecessor. These episodes of population-wide fitness decline merit further inquiry, but seem likely to be related to the implicit, contextual nature of fitness in this system.

Figure 3b shows the magnitude the median fitness differential observed for each predecessor competition. Although the emergence of morphologies *d*, *e*, and *g* were associated with significant increases in fitness, the magnitude of these fitness differentials is very similar to those of other stints (Figure 3b).

We also measured growth rate of specimen strains by tracking doubling time (in updates) when seeded into quarter-full toroidal grids (Figure 3c). Morph *b* exhibited a fast growth rate early on that was never matched by later morphs. This measure appears to be a poor overall proxy for fitness, highlighting the importance of biotic aspects of the simulation environment (which are not present in the empty space the assayed cells double into).

Fitness Complexity

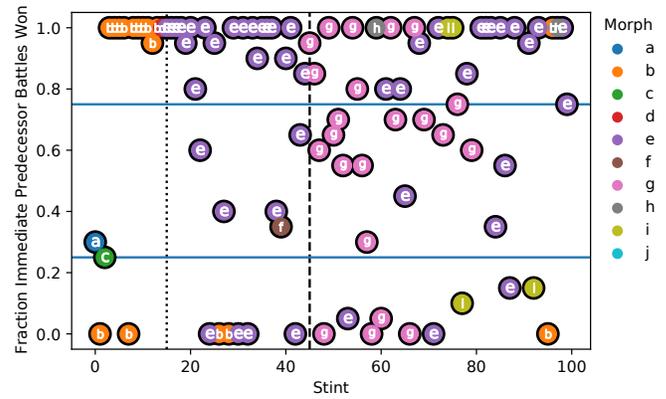
Figure 4a plots critical fitness complexity of specimens drawn from across the case study’s evolutionary history.

Critical fitness complexity reaches more than 20 under morph *b*, jumps to more than 40 under morph *d*, drops to slightly more than 30 for morph *e*. Critical fitness complexity reaches a peak of 48 sites around stint 39 then levels out and decreases.

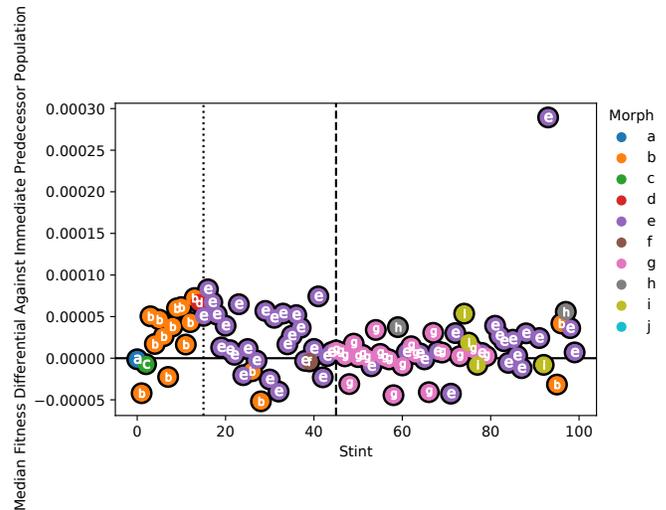
Interface Complexity

Figure 5 summarizes cardinal interface complexity, as well as its constituent components, for specimens drawn from across the case study’s evolutionary history.

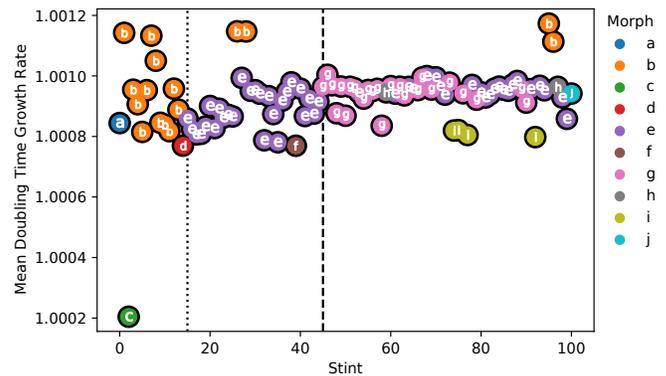
Notably, cardinal interface complexity more than doubles from 6 interactions to 17 interactions coincident with the emergence



(a) Fraction of 20 independent competitions that were won against immediate predecessor population. Blue horizontal lines represent significance level $p < 0.05$ for binomial null hypothesis. Neutral outcomes fall inside the blue bars, significant fitness increases fall above them, and significant fitness decreases fall below them.

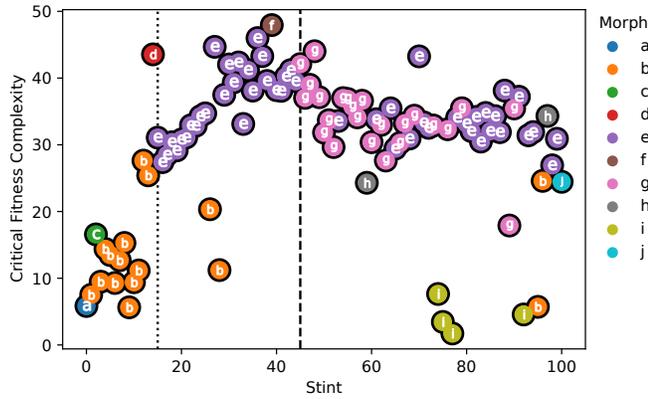


(b) Magnitude of fitness differential against immediately-preceding stint population. Positive fitness differential indicates greater fitness compared to predecessor. Solid horizontal line indicates neutral fitness differential.



(c) Growth rate estimated from doubling time experiments, measuring time for a monoculture to grow from 0.25 maximum population size to 0.5 maximum population size.

Figure 3: Fitness assays. Color coding and letters correspond to qualitative morph codes described in Table 1. Dotted vertical line denotes emergence of morph *e*. Dashed vertical line denotes emergence of morph *g*.



(a) Critical fitness complexity. Number of single-site nopouts that significantly decrease fitness, adjusted for expected false positives.

Figure 4: Fitness complexity estimates. Color coding and letters correspond to qualitative morph codes described in Table 1. Dotted vertical line denotes emergence of morph *e*. Dashed vertical line denotes emergence of morph *g*.

of morph *e* (Figure 5a). This is due to simultaneous increases in extrospective state sensing (2 to 9 states; Figure 5e), introspective state sensing (1 to 4 states; Figure 5d), and writable state usage (1 to 2 states; Figure 5f).

The emergence of morph *g* coincided with an increase in writable state interface complexity from 1 to 3 as shown in Figure 5f. However, morph *g* was not associated with other changes in other aspects of cardinal interface complexity. The greatest observed cardinal interface complexity was 22 interactions at stints 54 and 67.

Discussion

Throughout the case study lineage, we describe ten qualitatively distinct multicellular morphologies (Table 1). The emergence of some, but not all, of these morphologies coincided with an increase in fitness compared to the preceding population. For example, morphologies *c* and *f* do not significantly outcompete their predecessors while morphologies *d*, *e*, and *g* do (Figure 3a). This latter set of novelties might be described as “innovations,” which Hochberg et al. define as qualitative novelty associated with an increase in fitness (Hochberg et al., 2017). Interestingly, the magnitude of the fitness differentials associated with the emergence of morphologies *d*, *e*, and *g* do not appear to fall outside the bounds of other stint-to-stint fitness differentials (Figure 3b).

The relationship between innovation and complexity also appears to be loosely coupled. The emergence of morphology *d* was accompanied by a spike in critical fitness complexity (from 25 sites at stint 13 to 43 sites at stint 14). However, the emergence of morphology *e* may have coincided with a loss of critical fitness complexity (from 43 sites to 31 sites). Due to limitations in our phylogenetic tracking, it is unclear whether morphology *e* was direct descendant of morphology *d*. If, instead, morphology *e* evolved from remnants of morphology *b*, the emergence of morphology *e* coincided

with a more modest increase in fitness complexity from 25 sites to 31 sites. Similarly, the emergence of morphology *g* with 42 critical sites at stint 45 coincided with a relatively modest increase in fitness complexity from 39 critical sites at stint 44.

We also see evidence that increases in complexity do not imply qualitative novelty in morphology (though may involve behavioral novelty). In Figure 4a, we can also observe notable increases in critical fitness complexity that did not coincide with apparent morphological innovation. For example, fitness complexity spiked from 11 sites at stint 11 to 27 sites at stint 12 while morphology *b* was retained. In addition, a more gradual increase in fitness complexity was observed from 27 sites at stint 16 to 46 sites at stint 36 all with consistent morphology *e*.

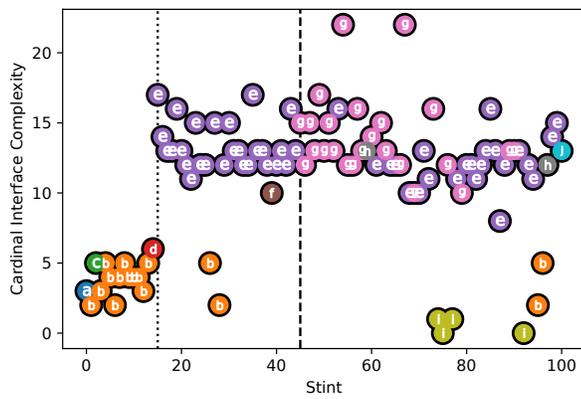
Finally, we also observed surprising contradictions between alternate measures of functional complexity. Notably, cardinal interface complexity more than doubled from 6 interactions under morph *d* to 17 interactions with the emergence of morph *e*. However, critical fitness complexity of morph *d* was 12 sites greater than morph *e* at stint 15. In addition, the gradual increase in critical fitness complexity between stint 15 and 36 under morphology *e* is not accompanied by a clear change in interface complexity. These apparent inconsistencies between metrics for functional complexity evidence the multidimensionality of this idea and underscore well-known difficulties in attempts to describe and quantify it (Böttcher, 2018).

Conclusion

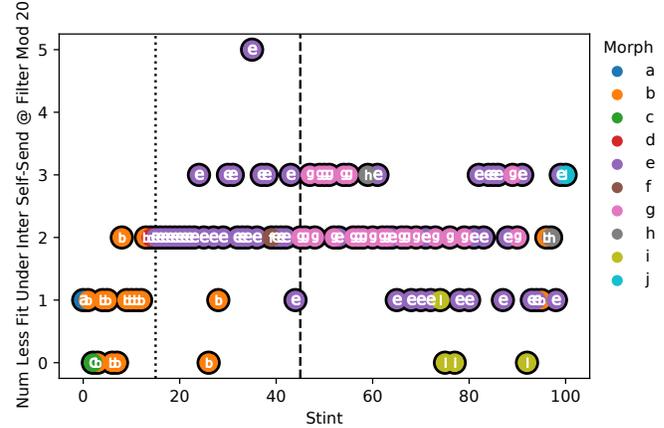
This case study sheds anecdotal light on a loose coupling between novelty, complexity, and adaptation.

We observe instances where novelty coincides with adaptation and instances where it does not. We observe instances where increases in complexity coincide with adaptation and where decreases in complexity coincide with adaptation. We observe instances where innovation coincides with spikes in complexity and instances where it does not. We even observe contradiction between metrics that measure different aspects functional complexity, with a near tripling of interface complexity coinciding with a drop in critical fitness complexity.

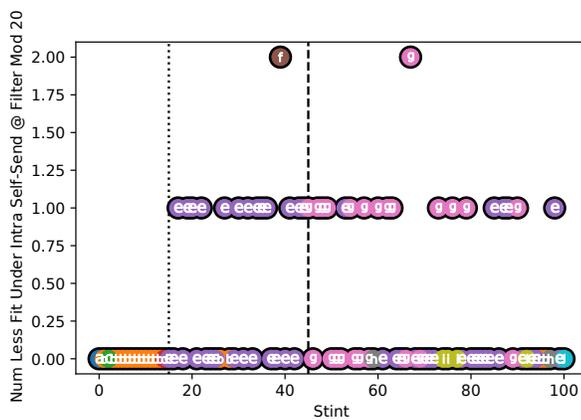
The anecdotal account of loose coupling between the conceptual threads of novelty, complexity, and adaptation provided by this case study highlights the importance of considering these factors independently when developing open-ended evolution theory — any coupling among them is by no means for granted. This case study highlights the potential usefulness of toolbox-based approaches to analyzing open-ended evolution systems in which an array of analyses are performed to distinguish disparate dimensions of open-endedness (Dolson et al., 2019). In future work, we are interested in further extending this toolbox. In particular, we are interested in developing methodology for systems where fitness is implicit and expensive to measure that will allow estimation epistatic contributions to fitness without resorting to all-pairs knockouts or other even more extensive assays.



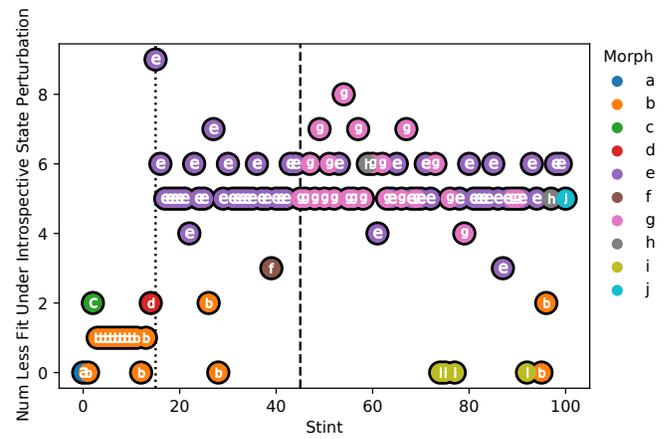
(a) Cardinal interface complexity, the total number of distinct interactions between a virtual CPU controlling cell behavior and its surroundings that contribute to fitness. (Sum of Figures 5e, 5d, 5f, 5b, and 5c.)



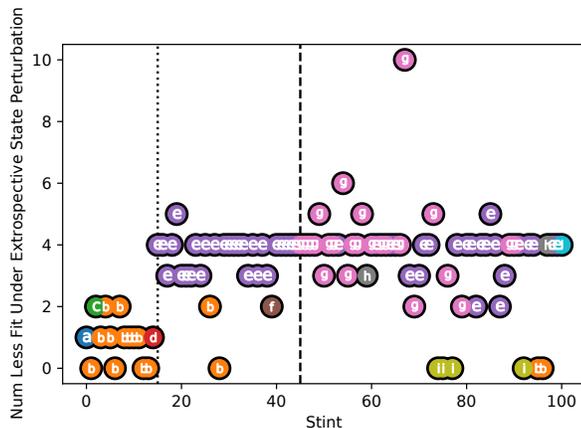
(b) Intermessage interface complexity, the number of distinct inter-cell messages that contribute to fitness.



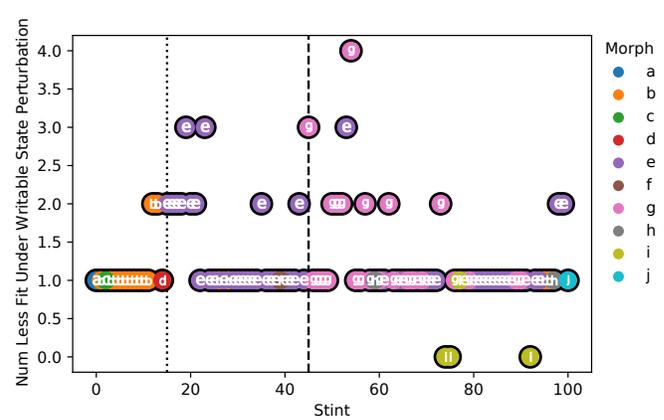
(c) Intramessage interface complexity, the number of distinct inter-cell messages that contribute to fitness.



(d) Introspective interface complexity, the number of states viewed in the own cell that contribute to fitness. See Supplementary Figure 7 for detail on the introspective states that contribute to fitness.



(e) Extrospective interface complexity, the number of states viewed in neighboring cells that contribute to fitness. See Supplementary Figure 6 for detail on the extrospective states that contribute to fitness.



(f) Writable state interface complexity, the number of output states that contribute to fitness. See Supplementary Figure 8 for detail on the writable states that contribute to fitness.

Figure 5: Interface complexity estimates. Color coding and letters correspond to qualitative morph codes described in Table 1. Dotted vertical line denotes emergence of morph *e*. Dashed vertical line denotes emergence of morph *g*.

Acknowledgements

Thanks to members of the DEVOLAB, in particular Katherine Perry for help implementing DISHTINY visualizations. 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.

References

- Ackley, D. and Small, T. (2014). Indefinitely scalable computing=artificial life engineering. In *Artificial Life Conference Proceedings 14*, pages 606–613. MIT Press.
- Adami, C., Ofria, C., and Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463–4468.
- Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In *Artificial life VI*, pages 228–237.
- Böttcher, T. (2018). From molecules to life: quantifying the complexity of chemical and biological systems in the universe. *Journal of molecular evolution*, 86(1):1–10.
- Dolson, E. (2019). *On the Constructive Power of Ecology in Open-ended Evolving Systems*. PhD thesis, Michigan State University.
- 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.
- Downing, K. L. (2015). *Intelligence emerging: adaptivity and search in evolving neural systems*. MIT Press.
- Hochberg, M. E., Marquet, P. A., Boyd, R., and Wagner, A. (2017). Innovation: an emerging focus from cells to societies.
- Huizinga, J., Stanley, K. O., and Clune, J. (2018). The emergence of canalization and evolvability in an open-ended, interactive evolutionary system. *Artificial life*, 24(3):157–181.
- Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences*, 95(15):8420–8427.
- Lalejini, A. and Ofria, C. (2018). Evolving event-driven programs with signalgp. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1135–1142.
- Lehman, J. (2012). *Evolution through the Search for Novelty*. PhD thesis, University of Central Florida.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.
- Lehman, J. and Stanley, K. O. (2012). Beyond open-endedness: Quantifying impressiveness. In *Artificial Life Conference Proceedings 12*, pages 75–82. MIT Press.
- Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one*, 8(4):e62186.
- Lynch, M. (2007). The frailty of adaptive hypotheses for the origins of organismal complexity. *Proceedings of the National Academy of Sciences*, 104(suppl 1):8597–8604.
- Moreno, M. A. (2021). Oee4 supplementary material. <https://osf.io/gekc8>.
- Moreno, M. A. and Ofria, C. (2019). Toward open-ended fraternal transitions in individuality. *Artificial life*, 25(2):117–133.
- Moreno, M. A. and Ofria, C. (2021). Exploring evolved multicellular life histories in an open-ended digital evolution system. *arXiv preprint arXiv:2104.10081*.
- Moreno, M. A., Papa, S. R., and Ofria, C. (2021). Conduit: A c++ library for best-effort high performance computing. *arXiv preprint arXiv:2105.10486*.
- Nguyen, A. M., Yosinski, J., and Clune, J. (2015). Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, page 959–966, New York, NY, USA. Association for Computing Machinery.
- Packard, N., Bedau, M. A., Channon, A., Ikegami, T., Rasmussen, S., Stanley, K. O., and Taylor, T. (2019). An overview of open-ended evolution: Editorial introduction to the open-ended evolution ii special issue. *Artificial life*, 25(2):93–103.
- Soros, L. and Stanley, K. (2014). Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In *Artificial Life Conference Proceedings 14*, pages 793–800. MIT Press.
- Stanley, K. O., Lehman, J., and Soros, L. (2017). Open-endedness: The last grand challenge you’ve never heard of. *O’Reilly Online*.
- Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., et al. (2016). Open-ended evolution: Perspectives from the oee workshop in york. *Artificial life*, 22(3):408–423.