EVEVis: A Multi-Scale Visualization System for Dense Evolutionary Data

Robert Miller* Vadim Mozhayskiy† Ilias Tagkopoulos‡ Kwan-Liu Ma§

University of California, Davis

ABSTRACT

Evolutionary simulations can produce datasets consisting of thousands or millions of separate entities, complete with their genealogical relationships. Biologists must examine this data to determine when and where these entities have changed, both on an individual basis and on a population-wide basis. Therefore, desirable features of a visualization system for evolutionary data are the capability of showing the status of the population at any given moment in time, good scalability, and smooth transition between high-level and low-level views. We propose a multi-scale visualization method, including a novel tree layout that both shows population status over time and can easily scale to very large populations. From this layout, the user can navigate to visualizations for moments in time or for individual entities. We demonstrate the effectiveness of the visualization on an existing evolutionary simulation called EVE: Evolution in Variable Environments.

Index Terms: H.5.m [Information Interfaces and Presentation]: Miscellaneous—Multi-scale visualization; J.3 [Computer Applications]: Life and Medical Sciences—Biology and Genetics

1 INTRODUCTION

Genetic algorithms are used for applications ranging from economic modeling of the stock market to machines learning to play better chess, but tools for analysis of their behavior are currently insufficient. Such algorithms make use of a rudimentary evolutionary simulation to evolve behaviors that optimize some performance metric. Most users of genetic algorithms are concerned only with the best behavior evolved, so earlier iterations are discarded as the algorithm continues. However, some users of genetic algorithms are concerned with the actual process of evolution and how it occurs. In such cases data must be stored for a large sample of the evolving population so that observed changes in behavior can be analyzed later. One example is that of biologists who have developed a cellular evolution simulator named EVE [31] [22][23]. EVE provides an in-silico representation of cells, which researchers can analyze to understand how biological cells evolve and function. One specific capability that these researchers would like to have is the ability to use the same visualization tool to see both the large-scale population dynamics in a simulation, as well as the small-scale characteristics of individual cells. We have designed a new visualization tool we call EVEVis specifically to permit such multi-scale analysis of data.

2 RELATED WORK

There exist several visualization tools for phylogenies, but these tools are designed for analysis of biological phylogenies, such as the tree of life [5] [20]. EVE stores complete information about the parent-child relationship between every cell generated in a simulation run. This produces a phylogenetic forest with detailed characteristics of each of the millions of simulated cells. Complete data such as this is not normally available for large phylogenies, so existing visualization tools are not designed to handle such data.

A number of visualization tools also exist for the representation of cellular metabolic pathways such as those simulated by EVE [12][13]. The most popular of these is Cytoscape [19] [28], and there exist additional plugins for Cytoscape that extend its capabilities [3][6]. All of these tools are designed to assist scientists in determining and constructing accurate representations of measured metabolic pathways.

EVE data contains a representation of the entire cellular network used for simulating cells, so there is no need for assistance in determining the pathway. EVEVis therefore has no need to provide the user with tools to edit the cellular network and instead focuses on providing a good visualization of the network for analysis.

Some tools for metabolic pathway visualization provide the capability of plotting the expression profiles (behavior signals) of cellular components [6]. Cells generated during EVE simulations are several orders of magnitude simpler than the biological cells that other visualizations are intended for. EVEVis takes advantage of this simplicity by representing the expression profiles of every node in the cellular network in a compact manner.

Dendrograms are a commonly used representation for small phylogenies [30]. They possess the property that the distance between two elements can be used as an approximate metric for how distantly related the two elements are [26]. EVEVis also possesses this property in the StackTree layout by enforcing restrictions on the horizontal positioning of cell tracks, similar to the requirements for dendrogram construction.

Junghans describes a force-directed potential-based edge routing method [16]. EVEVis implements this method for the generation of the cellular network diagram.

Using scale to change the focus of analysis from small details to large trends is a very common visualization technique [29] [4][11]. An early example was presented by Bederson [4]. An analysis of the usability of such interfaces was presented by Hornbaek [11]. EVEVis makes use of this technique in the StackTree layout to help the user hide unimportant details when looking for large- scale trends, while highlighting these same details when the user zooms in for close analysis.

There do exist several other tools designed for analysis and visualization of genetic algorithms [9] [17]. One such tool is GAVEL [9], which provides a visualization of the alleles and genes produced by a given genetic algorithm. EVEVis differs from GAVEL in that our visualizations focus primarily on the behavior of the evolved elements, rather than visualizing the genetic differences directly.

Node-link representations for the depiction of trees such as those in a phylogeny are very common. Representing the passage of time by using an axis in a tree layout has also been commonly implemented [1]. Biologists have also previously used continuous diagrams to represent trends in very large groups of studied element [10]. These continuous representations are essentially stack graphs for groups of cell properties over time. EVEVis attempts to combine these two representations with the StackTree layout.

*e-mail: bobmiller@ucdavis.edu

†e-mail:mozhaysk@ucdavis.edu

‡e-mail:iliast@cs.ucdavis.edu

§e-mail:ma@cs.ucdavis.edu

IEEE Symposium on Biological Data Visualization
October 23 - 24, Providence, Rhode Island, USA
978-1-4673-0004-9/11/$26.00 ©2011 IEEE

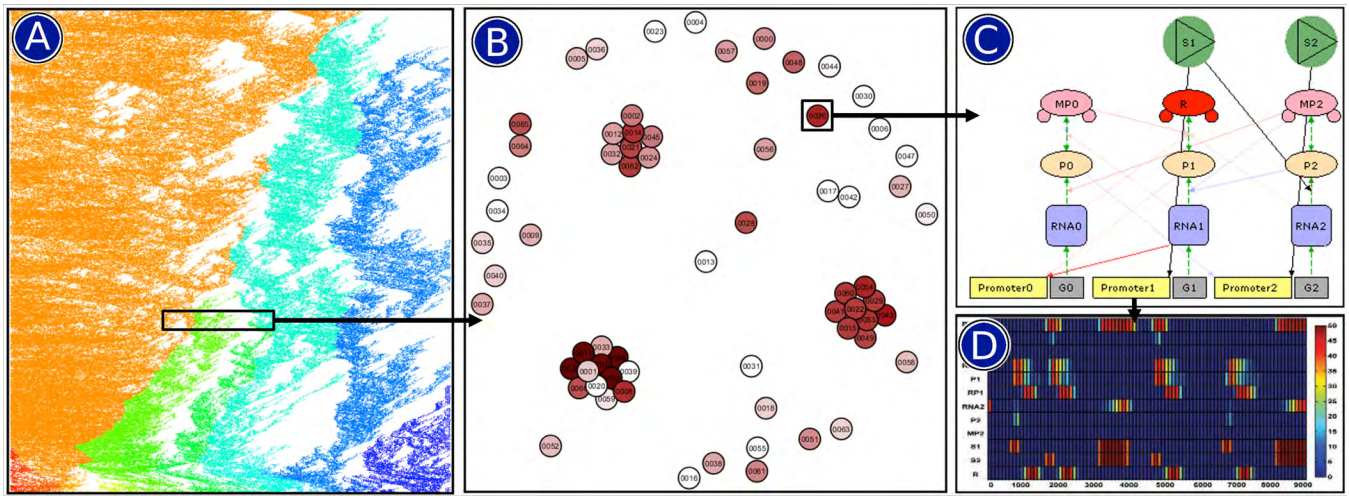

Figure 1: Scales of visualization in EVEVis. Panel A shows the visualization of the phylogeny produced by an EVE run, where different colors represent different evolved species in the simulation. Panel B shows a population of cells selected from Panel A for detailed visualization via clustering based on similarity of cell behavior. Panel C shows the behavioral network corresponding to one of the cells from Panel B. Panel D shows a heatmap representing the behavior of individual nodes in the cellular network from Panel C under a given input. The black arrows show how a user can transition from each visualization scale to the next.

3 Data

EVE (Evolution in Variable Environments) is a cellular evolution simulator for which we have developed a visualization tool we call EVEVis . The data produced by EVE can be partitioned into several categories, including a phylogenetic forest representing the parent-child relationships between every cell produced over the course of the entire simulation, population-level data containing summaries of characteristics of cells coexisting simultaneously at any moment in the simulation, cellular network data representing the internal structure of simulated cells, and node-level data describing the behavior of cellular components under given inputs.

3.1 Phylogenetic Data

The phylogenetic forest is a representation of the parent-child relationships between the cells in the simulation. It is a forest, rather than a tree, because EVE generates some random cells with no parent throughout the simulation. These random cells and their offspring each generate genetically distinct phylogenetic trees. Most EVE simulations contain only asexual reproduction, so descendants of a cell only inherit traits of a single ancestor. Representations of the phylogenetic tree are made more complicated by the fact that, in many cases, cell mutation does not occur during the creation of an offspring cell, but instead during a cell's normal life span. Throughout an EVE simulation, the number of living cells remains approximately constant.

3.2 Cell Characteristics

Cells in the data generated by the EVE simulator are stored using several different kinds of information. For each cell generated during the simulation, EVE stores the following data:

- The number of triplets, or genes
- Numerical description of the probability of various types of cell mutation over fixed time periods
- Energy level of the cell
- A "fitness" characteristic describing the cell's measured ability to conform to an expected output
- A matrix representing the network of interactions between the genes in the cell

4 EVEVis: Overview

EVE generates a multitude of different kinds of data, describing the different portions of its underlying genetic algorithm. Multi-scale visualization is therefore a natural method to fully visualize the data generated during a simulation run. We designed EVEVis to be such a multi-scale visualization tool.

EVEVis has a visualization component for each level of data generated by EVE. Figure 1 shows how EVEVis incorporates each of these components into a cohesive interface.

Using EVEVis, the user starts with a general view of the entire phylogeny produced during the simulation, as shown in Figure 1a. At this scale, it is possible to analyze the changes in cell or population characteristics that take place over long periods of the simulation. It is also possible to zoom in to see the changes in characteristics between individual cells and how these cells are genetically related.

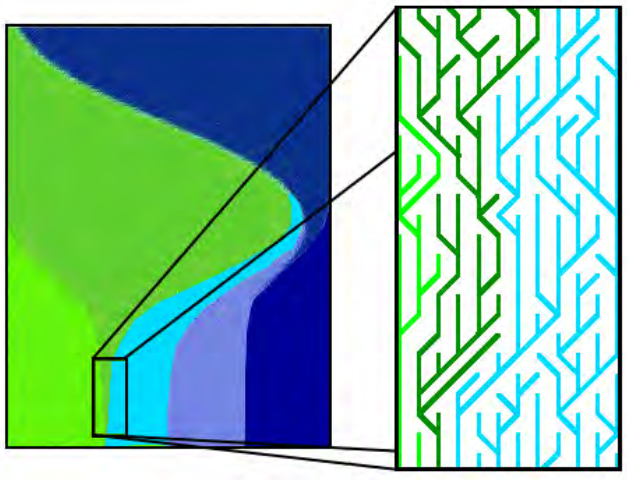

Figure 2: Concept of StackTree layout, transitioning between a stack graph visualization of cellular properties to a node-link tree layout showing relationships between individual cells.

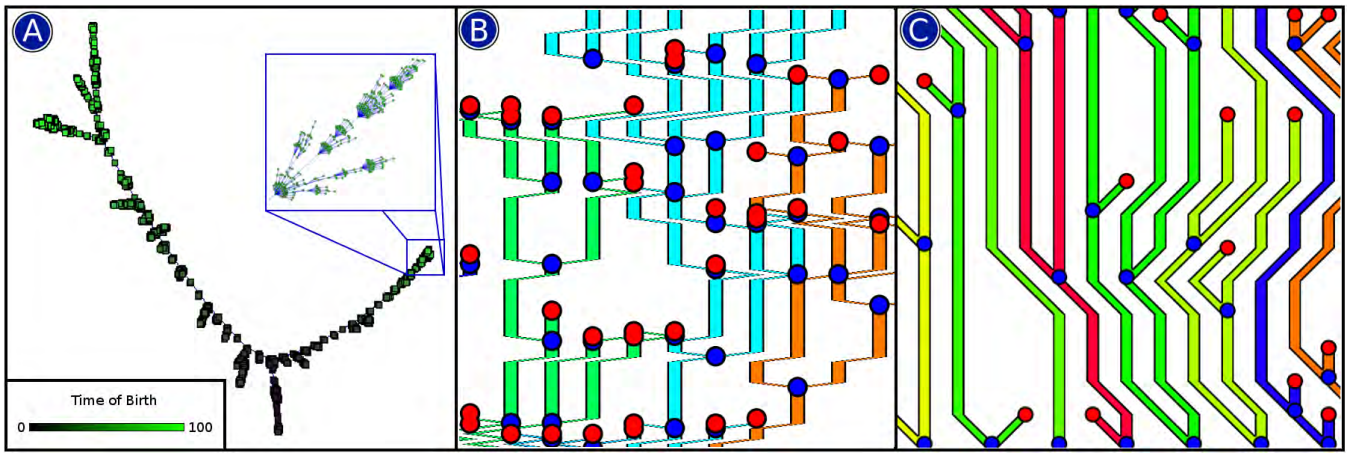

Figure 3: Precursors to the StackTree layout. Panel A shows the initial attempt to represent a phylogeny as a node-link diagram drawn with the FM3 layout technique. Large scale structure is visible, but small-scale structure is difficult to observe. Furthermore, there is no indication of time in the diagram (unless nodes are colored by their time of birth, as shown), so it is difficult to determine which cells existed simultaneously in the simulation. Panel B shows an early attempt at the node-link component of the StackTree layout. The vertical axis now represents time, but analysis remains difficult because tracks representing individual cells zigzag horizontally and blend together. Panel C shows the final form of the node-link component of the StackTree layout. The vertical axis now represents the sequence of events, so earlier births and deaths are displayed lower in the diagram. This has the effect that cells represented on the same horizontal line coexisted simultaneously in the simulation. Horizontal motion of cell tracks is restricted between rows, preventing cell tracks from blending together.

From the phylogenetic visualization, the user can select a group of cells to visualize that population, as shown in Figure 1b. At this scale, the cells can be clustered and recolored so that the user can analyze characteristics of the population, such as the number of distinct species of behavior and how some basic properties of the internal behavior of the cells differ within and among these species. From either the phylogenetic or the population level of visualization, it is possible for the user to select a single cell for detailed analysis. This produces a visualization of the behavioral network of the selected cell, as shown in Figure 1c. This network fully describes the behavior of the selected cell. Finally, the user may select a subset of nodes from the cellular network and generate a visualization of the behavior of these nodes under the given input, as shown in Figure 1d.

4.1 Phylogenetic Visualization: Overview

The visualization of the phylogenetic scale provides the initial, most general view in EVEVis. Figure 2 shows the basic concept behind this scale of visualization. In principle, when analyzing trends over the course of the entire simulation, a simple stack graph suffices, but when a user is interested in the interrelationships between individual cells, a node-link diagram showing the phylogenetic tree is necessary. We therefore developed a method for transitioning smoothly between a stack graph and a node-link diagram of the phylogeny.

4.1.1 The StackTree Layout

When initially designing the phylogenetic visualization, we chose to employ a traditional node-link tree layout using the FM3 layout technique [30]. We chose this layout because FM3 has been previously shown to be a good choice for displaying the structure of generic large graphs [8]. We added a bias force to the cells during the FM3 calculation, to push cells that evolved earlier in the simulation toward the bottom of the diagram, and to pull cells that evolved later toward the top. The resulting diagram is shown in Figure 3a. Unfortunately, most of the small-scale structure of the phylogeny was hidden by the large-scale structure, which led us to the conclusion that the FM3 layout was not suitable for large-scale phylogenetic analysis. Although large trends, such as the number of species that had emerged during the simulation, were visible, smaller trends were not. Furthermore, cells in the simulation exist over a period of time and can mutate during that time. Since the FM3 layout represented each cell as a single dot, there was no way to represent this aspect of the data in the diagram.

To solve the problems with the FM3 layout, we first considered existing layouts designed for large tree visualization [24], but in the end we chose to design a new tree layout specifically for the phylogenetic data produced by genetic algorithms. The vertical axis initially represented the time at which events in the simulation occurred, but due to the problems visible in Figure 3b, which will be described in detail later, the axis now represents only the sequence in which events occurred, forming a partial ordering. In addition, we no longer represent cells as nodes on the node-link diagram, but rather, as links. This change allows for the representation of changes in cell characteristics over the lifespan of the cell, as shown in Figure 5. This enables us to measure the lifespan of individual cells by simply subtracting the time of birth from the time of death.

The diagram is intended to highlight areas in the simulation in which a cell is drastically more successful than the cells it coexists with. The problem with this approach is that during most of the simulation runtime very few birth and death events occur, so much of the diagram space is wasted on uninteresting parts of the simulation. Worse, when such successful cells do appear, they tend to give rise to an exponential number of offspring, as the descendants outcompete other living cells in the simulation. When using the vertical axis to represent time directly, this sudden burst in reproduction produces sudden horizontal bursts in the generated diagram. As Figure 3b shows, such bursts make it difficult to track the path of cells because the other existing tracks are routed suddenly to the left or right to avoid colliding with the newborn cell tracks. Furthermore, these bursts occupy a relatively small area of the diagram, so analysis of such successful cells is difficult unless the user zooms far into the diagram.

To resolve the issue of events in the diagram occurring sparsely, the constraint that the vertical axis of the diagram exactly represent the time at which events occurred was relaxed to the requirement that higher portions of the diagram always represent events that took place later in the simulation than the events represented by

lower portions of the diagram. This allows regions sparsely populated by birth and death events to be collapsed into smaller regions that are more densely populated, which eases analysis. To alleviate the problem that successful cells produce sudden bursts in the diagram that are difficult to analyze, we instituted a constraint that the tracks representing cells have a maximum horizontal distance they can travel between lines in the diagram. This eliminates bursts such as those shown in figure 3b, by limiting the horizontal growth of a family of successful cells to a linear, rather than an exponential factor.

Thus far, we have discussed only the vertical distribution of cells. In the final diagram, it is required that birth and death events are plotted on the diagram vertically, from bottom to top, in the order in which they occurred in the simulation. The result is that the cells represented on the same horizontal line in the diagram existed simultaneously during the simulation. We determine the horizontal position of each cell by four requirements, illustrated by Figure 5. If these requirements cannot all be met simultaneously while inserting an event into a row, then the row is ended and the event is inserted into the next row. The first requirement for the horizontal position of cells is that when a cell undergoes mitosis (splits), the tracks in the diagram representing the offspring must diverge from the track that represented the parent, as shown in Figure 5a. When considering a cell split, both offspring are initially identical to the parent cell, with the exception of any mutations that may have occurred during the process. For simplicity, EVE therefore considers mitosis as a parent cell giving birth to a single child, with the parent's cellular ID continuing after the split. In EVEVis, the parent cellular track is always plotted to the left of the child's cellular track. This behavior is also represented in Figure 5a. The second requirement for horizontal placement of cells is that no cell tracks may cross at any point in the diagram. These first two requirements enforce a relationship that is shared by dendrograms, where, from any initial cell, more distant cells horizontally in either direction are also more distant genetically, as shown in Figure 5b [30]. Since genetically similar cells also tend to be behaviorally similar, this property has the advantage of clustering groups of cells with similar behaviors closer together in the final diagram. An example of this behavioral clustering can be seen in Figure 5a. The third requirement for horizontal placement of cells has been mentioned earlier: no cellular track may move horizontally by more than some fixed maximum distance between any two consecutive steps in the diagram. This restriction is shown in figure 5c. By default, EVEVis sets this maximum horizontal distance to be equal to the vertical distance between two consecutive steps in the diagram, which places an upper bound of 45° on the angle of inclination of any cellular track at any point in the diagram. The purpose of this requirement is to limit the procreation of cells to linear horizontal

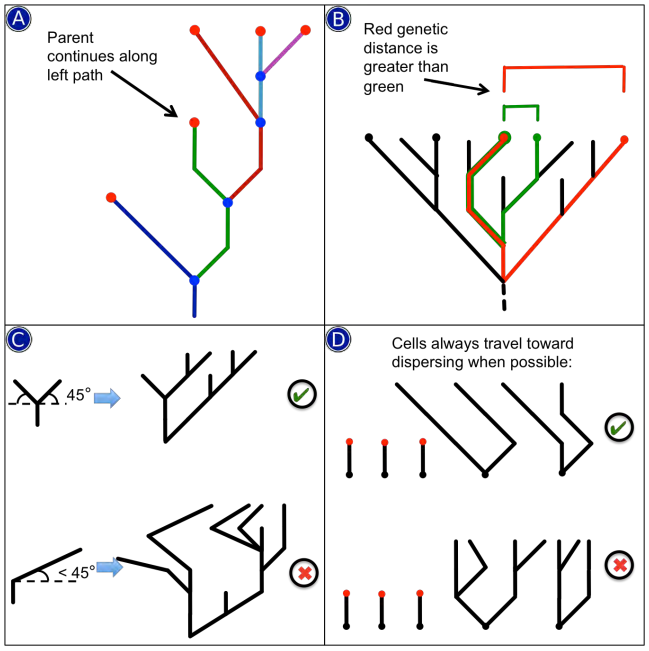

Figure 5: Restrictions on horizontal placement and movement of cell tracks to make the horizontal axis a metric of genetic distance (panels A and B), to prevent cell tracks from blending together (panel C), and to ensure that cell tracks remain distributed across the width of the diagram (panel D).

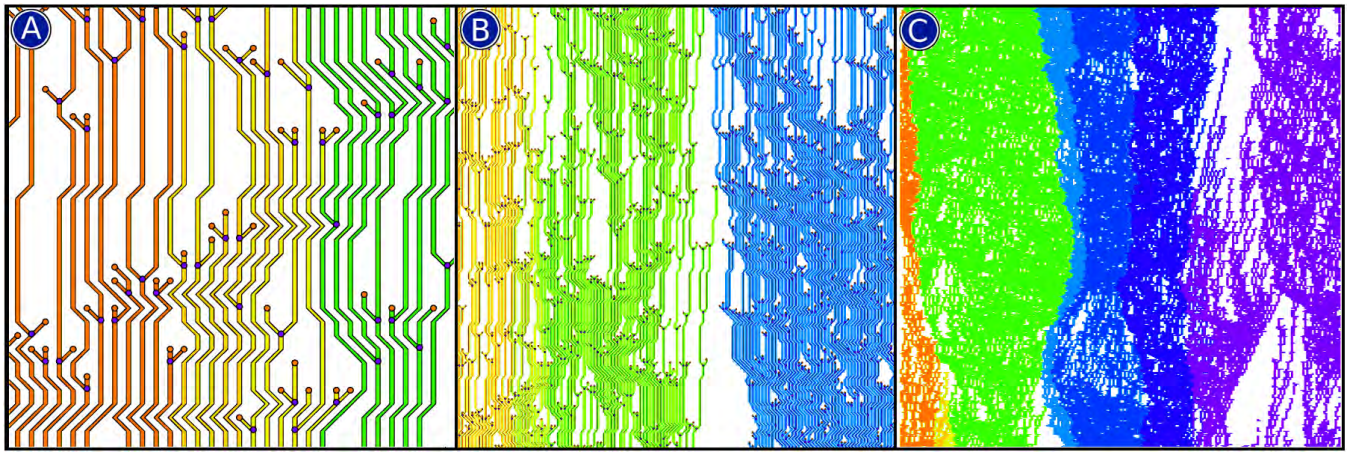

Figure 4: Transition of the StackTree node-link diagram for detailed analysis of individual cells (Panel A) to an approximation of a stack graph visualization (Panel C). Panel B shows an intermediate step. Each horizontal line represents a state of the population during the simulation, and the vertical axis represents time (up is later). Using the constraints described in Figure 4, the diagram is generated from the bottom up by attempting to represent the changes from the population represented by the previous horizontal line and the current population. When a change cannot be represented within the above constraints, the current horizontal line is completed and the change is pushed up to the next horizontal line. When all the population changes in the simulation have been recorded, the diagram is complete. This particular figure shows a competition between several species, each represented by a different color.

growth, which forces periods of rapid cellular growth to occupy a larger area of the diagram so that they are easier to detect during analysis. The fourth requirement for the StackTree diagram is that when multiple options satisfy the other three requirements, cells must migrate toward equal dispersion across the diagram's width. Without the requirement to maximize dispersion on each row, living cells quickly clump together and severely limit the number of birth and death events that can be placed on individual rows, while still satisfying the other requirements. This greatly increases the height of the diagram and reduces the density of events plotted in the diagram. This requirement and the behavior enforcing it are shown in Figure 5d. The final StackTree diagram is shown in Figure 4a. Birth and death events are represented by circular glyphs, and can be recolored by any characteristic of the cell that is available at the time of birth or death, such as the number of nodes contained in the cellular network. Three additional characteristics can be visualized over the course of the cell's lifetime by independently recoloring the center, border, and background of a cell track. In general, applying more than one color to a cell track appears to generate too much visual clutter to be useful.

4.1.2 Stack Graphs for Large Trends

The tree layout described in the previous section is useful when attempting to analyze differences between small groups of related cells, but it is not directly useful for the analysis of large general trends that occur over long periods of the simulation, because the glyphs for birth and death events clutter the diagram as the user zooms out, and because the colors representing each cellular characteristic become difficult to distinguish at that scale. Additionally, the white space between the cell tracks causes a desaturation of the diagram when viewed at this zoom level, which further obfuscates the large-scale trends in the simulation.

To alleviate these problems, the tree diagram described in the previous section smoothly fades into a simplified version as the user zooms out. The simplified version of the diagram does not display glyphs for the representation of birth and death events, which greatly reduces clutter and does not interfere with analysis, since birth and death events are specific to individual cells and are therefore inconsequential to large-scale trends in the simulation. Although three cell characteristics can be displayed in the magnified version of the diagram, only the coloring chosen for the center of the cell tracks is retained when the user zooms out. This allows the user to see the large-scale trends of the cell characteristic, as shown in Figure 4c. The transition between the two scales is shown in Figure 4b.

4.2 Population View

Once the user finds an interesting portion of the simulation to analyze from the phylogenetic diagram, it is likely that he or she will want to analyze that section in more detail. Relationships between cells and some cellular characteristics are already visible at the phylogenetic scale, but some characteristics are difficult to analyze in that layout. At this point, the user can select a set of cells from the phylogenetic diagram, which are then used to generate a population-level visualization, as shown in Figure 6. The advantage of visualization at this level is that both spatial dimensions are available for the display of cellular characteristics. Figure 6 shows a layout where the cells have been clustered based on their output signal, then have had a force-directed layout in two dimensions to show the results.

4.3 Cellular Network View

From either the population scale or the phylogenetic scale, an individual cell may be selected for detailed analysis. At this point, the user can select a cell and generate a visualization of the behavioral network of that cell. These cellular networks are used by EVE to simulate the behavior of the simple metabolic pathways that arise in biological cells. Each node in the network represents a cellular component that reacts to some number of input signals from other cellular components and produces a number of output signals that are fed to other cellular components. How these nodes are connected and how powerful their signal is to other nodes can be determined through the analysis of a graph such as the one shown in Figure 7. The glyphs representing the different components and the hyper-graph style of the visualization are taken directly from visualization techniques formerly generated by hand [31][32].

There are several nodes that differ from the others in the visualization. The red node at the upper left corner of Figure 7 represents the cell's output signal. In EVE, this signal represents the cell's attempt to consume a resource external to the cell. The cell has no direct knowledge of the resource's availability, but the cell is connected to one or more input signals, which are represented by the green circle nodes at the upper right corner of Figure 7. These input signals generally do not directly correlate with the existence of the available resource, but cells are expected to evolve to perform some calculation on the given inputs to predict the availability of the resource. For instance, an EVE simulation may expect the cells to evolve an XOR network by making the resource available only when one of two signals is active, but not both. Cells which evolve this behavior will be rewarded by the simulation and will be more likely to produce offspring, while cells that do not will face a significant disadvantage and will likely go extinct.

The scientists who developed EVE are accustomed to a particular layout for the cellular components. The layout of the nodes in this visualization defaults to a standard grid with the different types of cellular components placed in rows. We employed an energy-based edge routing method to route the edges between the components [16]. By moving the mouse over an edge, the user can highlight the edge and its connected nodes and can view its connection weight, which is normally hidden to reduce visual clutter.

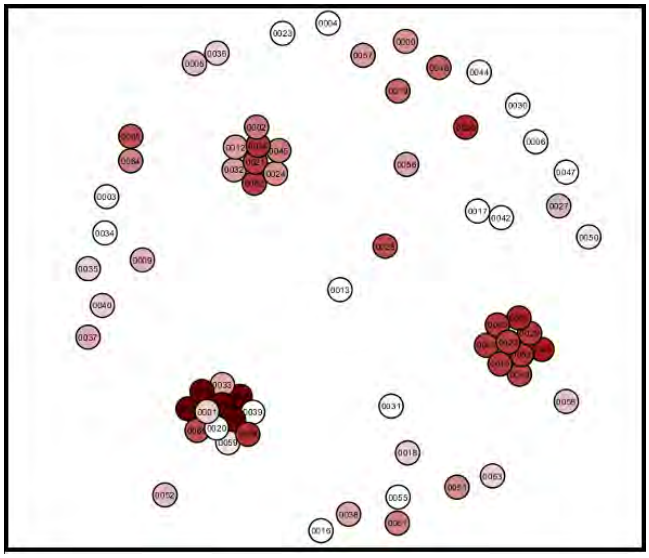

Figure 6: Population visualization. At the population scale the vertical axis no longer represents time, so we may choose a layout that uses position to show cellular characteristics. In the figure, the average square distance between the output signals of each cell is calculated. A force-directed layout is applied so that distance between two plotted cells forms an approximate metric for the difference in behavior of those two cells. pair. Cells in this diagram are colored so that redder cells represent cells that are more able to accurately simulate the target output signal for a given input.

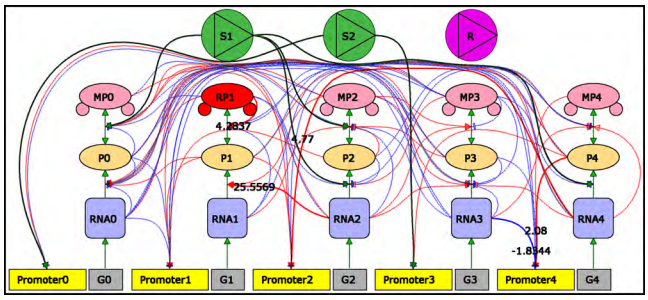

Figure 7: Cellular behavioral visualization. Each of the different nodes in the graph represent simulated biological components. Edges between nodes represent chemical pathways that occur between cellular components. Some pathways are persistent, such as the connections between RNA, proteins, and modified proteins, which together form a construct called a triplet. The red node represents the output connection of the cell, while the green triangles represent the connections to the input signals for the cell. Connections can be filtered to remove clutter. Additional information about this type of diagram is available in [31].

Also shown in Figure 7 by the bold highlighted edges is the minimal network for the cell. This minimal network is computationally expensive for EVE to find, but it produces approximately the same behavior as the complete network, with far fewer edge connections. To generate this network, EVE removes one edge at a time and then simulates the cell behavior. If the removal produces a drastic change in the output signal, then the edge is restored to the graph. This continues until any edge removal produces a significant change in the output signal. Once generated, these networks are significantly easier to analyze because they often contain only a small fraction of the original edges. However, because generation of such networks takes such a long time, it is not feasible to generate them during the initial simulation run. Therefore, EVEVis can be used to find cells that are of interest, at which point the minimal network can be calculated and displayed.

4.4 Node View

The lowest and most detailed scale of visualization available in EVEVis is at the cellular node scale. The user selects a subset of the nodes in a cellular network, which generates a visualization of the behavior of these nodes, as shown in Figure 8. Many biologists are already accustomed to analyzing the behavior or expression profiles in microarray data using heatmaps because of the chemical nature in which the data is normally obtained [33]. Therefore, the default method of visualization for node behavior in EVEVis is a heatmap, as shown in Figure 8a.

However, an alternative visualization of node behavior is also available in EVEVis, which we believe is in many ways superior to the heatmap version. As shown in Figure 7, cellular networks are divided into subsets of 3 nodes each, called triplets. There exists a set of predefined connections within each triplet that simulates the structure of elements in biological cells. When viewing the output of a triplet on a heatmap, the behavior patterns of the three components are shown separately. However, another option for the visualization is a simple overlapped line plot that shows the behavior of all three components of a triplet on the same axis. This type of plot is much more objectively measurable than a heatmap, because it is significantly easier to judge the shape of a line plot than to analyze the difference between hues on a heatmap.

5 Discussion and Future Work

EVEVis is written entirely in C++ with heavy dependency on the Boost library and Qt framework. Since each component is cross-platform, EVEVis should work on any platform. Generation of the StackTree layout requires precomputation when a dataset is loaded for the first time. The time to precompute a StackTree is linear in the number of cells generated during simulation, and takes approximately 30 minutes on a 2.2 GHz Intel Core i7 with 4 GB of RAM, when calculated for a standard EVE dataset (approximately 1,000,000 cells). After precomputation, the StackTree layout is saved to disk and can be reloaded in constant time.

The StackTree layout is intended to depict a node-link diagram colored according to cell characteristics in a phylogeny when the user views the diagram at high magnification, and to depict a stack graph for those same characteristics when the user views the diagram at low magnification. To produce node-link layouts suitable for analysis, EVEVis requires blank spaces to be distributed throughout the diagram. When zooming out, these spaces cause slight variations from a proper stack graph visualization of the characteristics. As the user zooms out from the node-link view, the blank spaces should collapse horizontally so that a true stack graph is visible when zoomed out. This capability is targeted for a future version of EVEVis.

EVEVis is useful for examining population dynamics in EVE simulations, as well as to determine where, when and how specific types of cellular behavior evolved within the simulation. From this information, scientists can infer information about how simple cellular networks function.

Currently, EVEVis provides no support for analysis of datasets with sexual reproduction. The StackTree layout presented in this paper does not apply to datasets with sexual reproduction because the resultant genealogy is not a tree, but a graph. Applying similar techniques to sexual genealogies is a problem that will be addressed in a future version of EVEVis.

Future work for EVEVis also includes adapting the architecture to visualizing data produced by other genetic algorithms. The multi-scale architecture employed by EVEVis would work effectively for this purpose, because any such algorithm produces a phylogeny of objects. For any such data, the StackTree layout can be

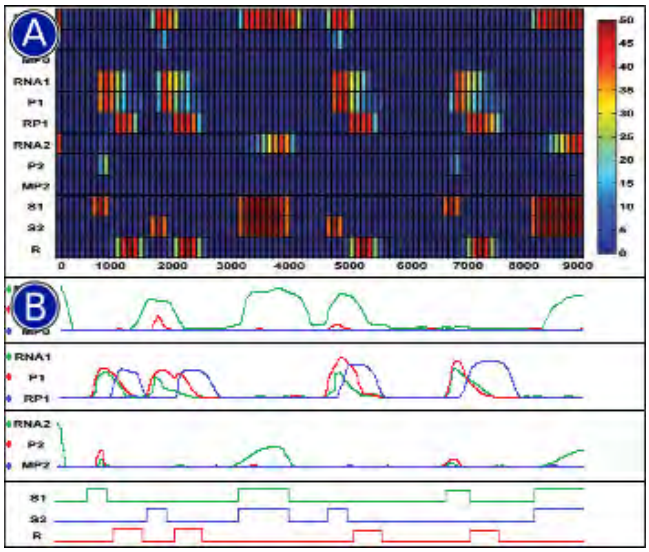

Figure 8: Representations of cellular node behavior, Panel A shows a traditional heatmap representation of the behavior of a subset of the nodes shown in Figure 7. Panel B shows an alternative line-plot representation of the same behaviors. Scientists analyzing cellular metabolic pathways are more accustomed to the heatmap visualization because it is similar in appearance to the microarray data collected from biological experiments [33].

used to visualize the phylogeny, a population-level visualization can be generated, by clustering based on object characteristics, and the properties of each generated object can be displayed using techniques applicable to the objects undergoing evolution.

Additional work for EVEVis will involve adaptation of the software and the StackTree algorithm to enable parallelization and to function as an in-situ visualization for EVE.

6 Conclusion

We have introduced EVEVis, a multi-scale visualization system that is specifically designed to display the output of the cellular evolution simulator EVE. Four scales of visualization are supported. The most general scale visualizes the phylogenetic relationships between individual cells using a novel forest layout, which we call the StackTree layout. The StackTree layout is applicable to any situation where the input is a forest with timestamps at each internal node representing 'births', and at each leaf representing 'deaths', but is most effectively applicable when the total living population is limited to less than the square root of the total number of nodes in the tree. The StackTree layout has the capability of showing individual differences during close analysis, and of reducing to a simple stack graph to show general trends, when viewed from a distance. Figures 9 and 10, which appear on the final page after the references, show additional examples of the use of the StackTree layout.

The other three scales of visualization generated by EVEVis show populations, individual cellular networks, and individual elements contained within the cellular networks. Each of these detail levels of visualization is generated using preexisting techniques.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants CCF-0808896, CNS-0716691, CCF 0811422, CCF 0938114, CCF-1025269, and OCI-0941360, and the UC Davis opportunity fund.

References

[1] M. Bamshad and S. P. Wooding. Signatures of natural selection in the human genome. *Nature reviews. Genetics*, 4(2):99–111, Feb. 2003.

[2] A. Barsky, J. L. Gardy, R. E. W. Hancock, and T. Munzner. Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. *Bioinformatics (Oxford, England)*, 23(8):1040–2, Apr. 2007.

[3] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: visualizing multiple experimental conditions on a graph with biological context. *IEEE transactions on visualization and computer graphics*, 14(6):1253–60, 2008.

[4] B. Bederson and J. Meyer. Implementing a zooming user interface: experience building Pad. *Softw Pract Exper*, 28(10):1101–1135, 1998.

[5] F. D. Ciccarelli, T. Doerks, C. von Mering, C. J. Creevey, B. Snel, and P. Bork. Toward automatic reconstruction of a highly resolved tree of life. *Science (New York, N.Y.)*, 311(5765):1283–7, Mar. 2006.

[6] O. Garcia, C. Saveanu, M. Cline, M. Fromont-Racine, A. Jacquier, B. Schwikowski, and T. Aittokallio. GOlorize: a Cytoscape plug-in for network visualization with Gene Ontology-based layout and coloring. *Bioinformatics (Oxford, England)*, 23(3):394–6, Feb. 2007.

[7] S. Hachul and M. Junger. Drawing large graphs with a potential-field-based multilevel algorithm. In J. Pach, editor, *Graph Drawing*, volume 3383 of *Lecture Notes in Computer Science*, pages 285–295. Springer Berlin / Heidelberg, 2005.

[8] S. Hachul and M. Junger. An experimental comparison of fast algorithms for drawing general large graphs. In P. Healy and N. Nikolov, editors, *Graph Drawing*, volume 3843 of *Lecture Notes in Computer Science*, pages 235–250. Springer Berlin / Heidelberg, 2006.

[9] E. Hart and P. Ross. GAVEL - a new tool for genetic algorithm visualization. *IEEE Transactions on Evolutionary Computation*, 5(4):335–348, 2001.

[10] M. Hegreness, N. Shoresh, D. Hartl, and R. Kishony. An equivalence principle for the incorporation of favorable mutations in asexual populations. *Science (New York, N.Y.)*, 311(5767):1615–7, Mar. 2006.

[11] K. Hornbaek, B. Bederson, and C. Plaisant. Navigation patterns & usability of zoomable user interfaces. *interactions*, 10(1):362–389, 2002.

[12] Z. Hu, J.-H. Hung, Y. Wang, Y.-C. Chang, C.-L. Huang, M. Huyck, and C. DeLisi. VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology. *Nucleic acids research*, 37(Web Server issue):W115–21, July 2009.

[13] T. Huan, A. Y. Sivachenko, S. H. Harrison, and J. Y. Chen. ProteoLens: a visual analytic tool for multi-scale database-driven biological network data mining. *BMC bioinformatics*, 9 Suppl 9:S5, 2008.

[14] D. Huson. SplitsTree: analyzing and visualizing evolutionary data. *Bioinformatics*, 14(1):68–73, Feb. 1998.

[15] D. H. Huson, D. C. Richter, C. Rausch, T. Dezulian, M. Franz, and R. Rupp. Dendroscope: An interactive viewer for large phylogenetic trees. *BMC bioinformatics*, 8:460, Jan. 2007.

[16] M. Junghans. Visualization of hyperedges in fixed graph layouts. Master's thesis, Brandenburg University of Technology, Cottbus.

[17] A. Kerren and T. Egger. EAVis: A Visualization Tool for Evolutionary Algorithms. *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, pages 299–301, 2005.

[18] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS computational biology*, 5(5):e1000385, May 2009.

[19] M. Kohl, S. Wiese, and B. Warscheid. Cytoscape: software for visualization and analysis of biological networks. *Methods in molecular biology (Clifton, N.J.)*, 696:291–303, Jan. 2011.

[20] I. Letunic and P. Bork. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. *Bioinformatics (Oxford, England)*, 23(1):127–8, Jan. 2007.

[21] A. Ma'ayan, S. L. Jenkins, R. L. Webb, S. I. Berger, S. P. Purushothaman, N. S. Abul-Husn, J. M. Posner, T. Flores, and R. Iyengar. SNAVI: Desktop application for analysis and visualization of large-scale signaling networks. *BMC systems biology*, 3:10, Jan. 2009.

[22] V. Mozhayskiy, B. Miller, K.-L. Ma, and I. Tagkopoulos. A scalable multi-scale framework for parallel simulation and visualization of microbial evolution. In *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*, TG '11, pages 7:1–7:8, New York, NY, USA, 2011. ACM.

[23] V. Mozhayskiy and I. Tagkopoulos. In silico evolution of multi-scale microbial systems in the presence of mobile genetic elements and horizontal gene transfer. In J. Chen, J. Wang, and A. Zelikovsky, editors, *Bioinformatics Research and Applications*, volume 6674 of *Lecture Notes in Computer Science*, pages 262–273. Springer Berlin, 2011.

[24] Q. V. Nguyen. A space-optimized tree visualization. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 2002:85–92, 2002.

[25] G. A. Pavlopoulos, S. I. O'Donoghue, V. P. Satagopam, T. G. Soldatos, E. Pafilis, and R. Schneider. Arena3D: visualization of biological networks in 3D. *BMC systems biology*, 2:104, Jan. 2008.

[26] J. Podani and D. Schmera. On dendrogram-based measures of functional diversity. *Oikos*, 115(1):179–185, Oct. 2006.

[27] P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization*, 4(3):191–205, June 2005.

[28] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–504, Nov. 2003.

[29] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 0(UMCP-CSD CS-TR-3665):336–343, 1996.

[30] B. Shneiderman. Interactively exploring hierarchical clustering results [gene identification]. *Computer*, 35(7):80–86, July 2002.

[31] I. Tagkopoulos. *Emergence of Predictive Capacity within Microbial Genetic Networks*. PhD thesis, Princeton University, 2008.

[32] I. Tagkopoulos, Y.-C. Liu, and S. Tavazoie. Predictive behavior within microbial genetic networks. *Science (New York, N.Y.)*, 320(5881):1313–7, June 2008.

[33] L. Wilkinson and M. Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, May 2009.

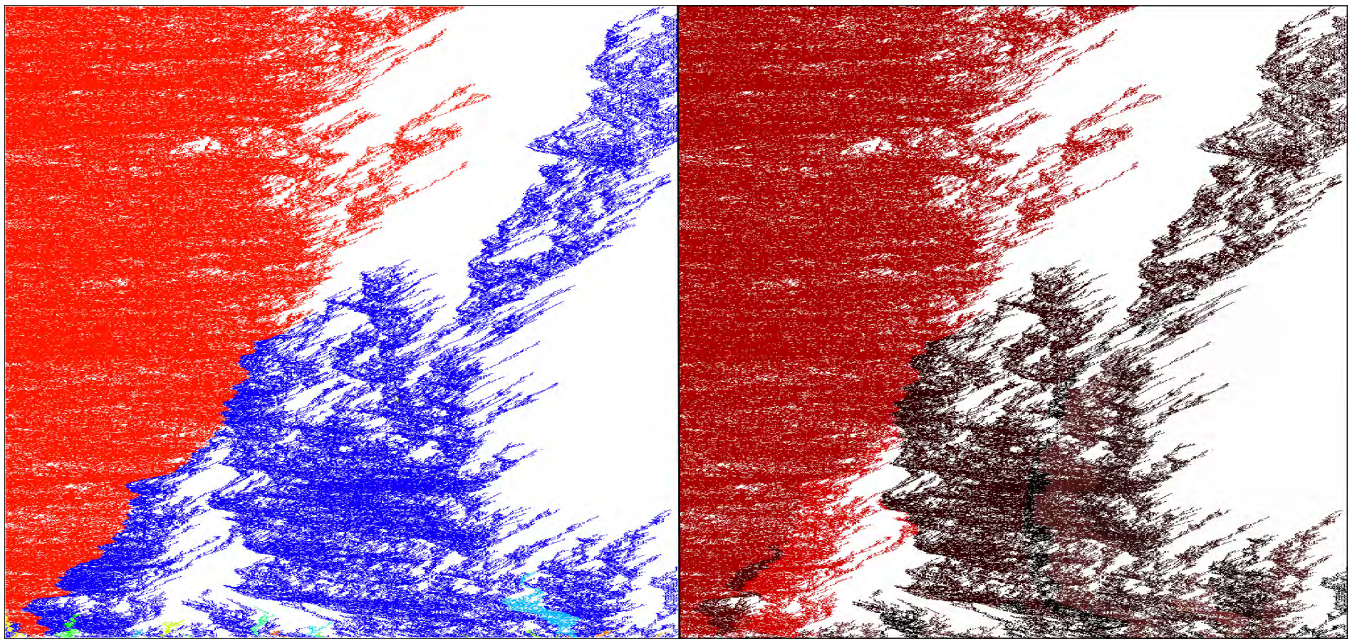

Figure 9: Competition between several evolved cell types. The left panel shows a StackTree visualization of competition between 8 species that evolved in separate simulations. Most are quickly outcompeted by the two most successful species, and finally one species completely overtakes the simulation. The right panel shows the visualization when recolored by average cell fitness, where a redder color indicates a higher fitness level. This makes it clear that the dominant species was consistently fitter than the species that went extinct.

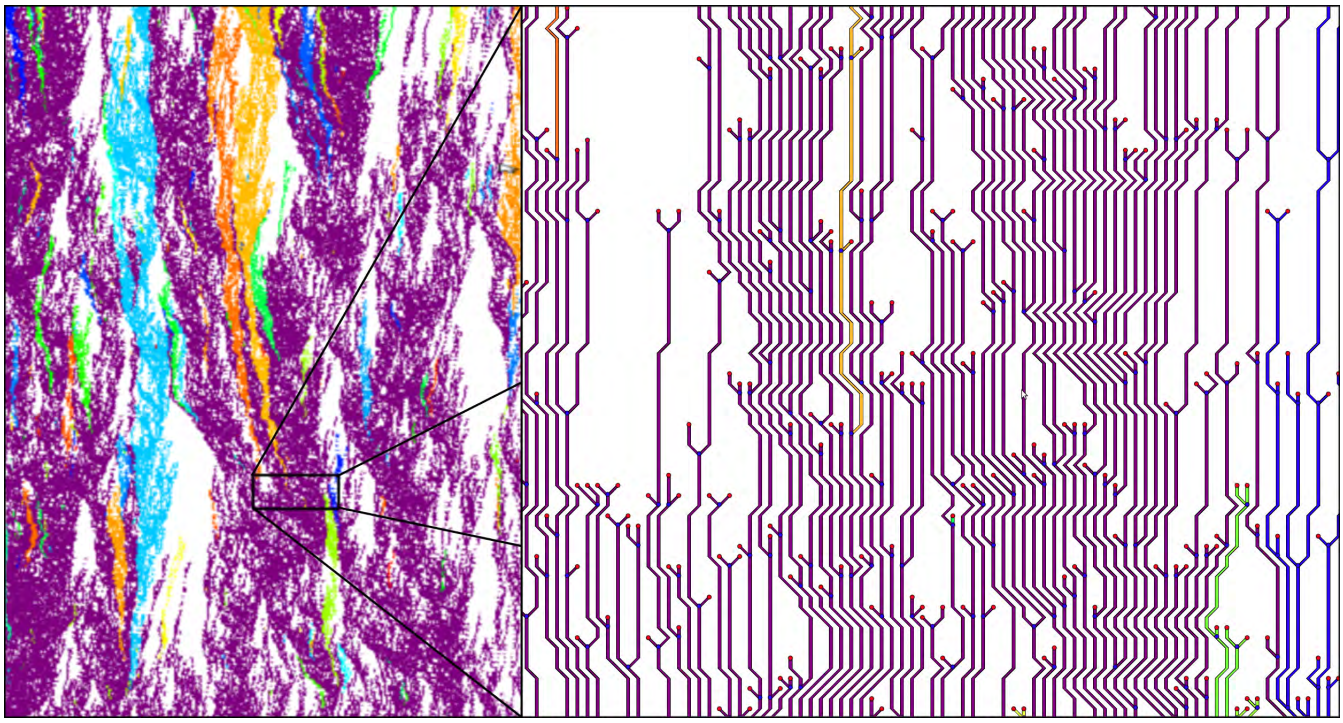

Figure 10: Tracking cellular mutations. The left panel shows a StackTree visualization colored randomly when the number of genes in a cell changes. Duplication and eradication of genes are common mutations in EVE simulations, and can generate large changes in cell behavior. The right panel shows the StackTree visualization at a high zoom level, so that the user can determine exactly when and where the mutation leading to one of the successful yellow species occurred.