LineageD: An Interactive Visual System for Plant Cell Lineage Assignments based on Correctable Machine Learning

Jiayi Hong, Alain Trubuil, Tobias Isenberg

Eurographics Conference on Visualization (EuroVis) 2022
R. Borgo, G. E. Marai, and T. Schreck
(Guest Editors)

Volume 41 (2022), Number 3

LineageD: An Interactive Visual System for Plant Cell Lineage Assignments based on Correctable Machine Learning

Jiayi Hong[1], Alain Trubuil[2], and Tobias Isenberg[1]

[1]Université Paris-Saclay, CNRS, Inria, France
[2] Université Paris-Saclay, InraE, France

Abstract

We describe LineageD—a hybrid web-based system to predict, visualize, and interactively adjust plant embryo cell lineages. Currently, plant biologists explore the development of an embryo and its hierarchical cell lineage manually, based on a 3D dataset that represents the embryo status at one point in time. This human decision-making process, however, is time-consuming, tedious, and error-prone due to the lack of integrated graphical support for specifying the cell lineage. To fill this gap, we developed a new system to support the biologists in their tasks using an interactive combination of 3D visualization, abstract data visualization, and correctable machine learning to modify the proposed cell lineage. We use existing manually established cell lineages to obtain a neural network model. We then allow biologists to use this model to repeatedly predict assignments of a single cell division stage. After each hierarchy level prediction, we allow them to interactively adjust the machine learning based assignment, which we then integrate into the pool of verified assignments for further predictions. In addition to building the hierarchy this way in a bottom-up fashion, we also offer users to divide the whole embryo and create the hierarchy tree in a top-down fashion for a few steps, improving the ML-based assignments by reducing the potential for wrong predictions. We visualize the continuously updated embryo and its hierarchical development using both 3D spatial and abstract tree representations, together with information about the model's confidence and spatial properties. We conducted case study validations with five expert biologists to explore the utility of our approach and to assess the potential for LineageD to be used in their daily workflow. We found that the visualizations of both 3D representations and abstract representations help with decision making and the hierarchy tree top-down building approach can reduce assignments errors in real practice.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization; User interface toolkits;***

1. Introduction

How individuals of species develop is an essential question in specific subfields of biology. By looking at the historical development of a plant embryo, botanists investigate whether the plant develops normally or not. As an embryo develops, single cells divide into two new cells and these parent-child relationships can be described using temporal hierarchies. The process of creating these hierarchies is called *cell lineage assignment* and requires biologists to (1) take images of a multi-cell plant embryo using a confocal microscope, (2) segment the resulting 3D volumetric data to identify the embryo cells, and (3) indicate pairs to reconstruct the hierarchy backward toward the root as the first cell (i. e., the *ovum*) of the embryo.

In our collaboration with botanists, we identified several parts of this workflow that can be improved with a dedicated visual analytics tool. Existing tools (a) only use two-dimensional slice-based interaction to mark sister cells, (b) do not take advantage of either 3D interaction or interaction with a visual representation of the hierarchy, (c) do not allow the scientists to interactively divide the 3D geometry into parts that can be identified as being part of early hierarchy levels in the cell division, and (d) do not provide any means to automate the cell assignment process based on known cell division patterns. Given these limitations of existing tools, the current, largely manual workflow of biologists does not scale to embryos with hundreds of cells (i. e., 7 or more generations of cell divisions). A purely automatic approach would also not be ideal as they want to retain manual control over the process, to be able to deal with incorrect assignments and the special behavior of mutants.

We thus developed LineageD to address these limitations. LineageD is based on specific domain requirements that we elicited with our collaborator and described in detail later in the paper. Specifically, it was important for our collaborator that the tool would be *interactive* to retain control over the hierarchy construction process, contain the familiar *3D representations* of the embryo state, display a *temporal hierarchy tree* that encodes abstract temporal information about cell divisions, *closely link* the 3D and abstract representations, and include *machine learning* approaches to propose cell lineages and reduce repetitive work. We built the tool based on these require-

ments, while ensuring that all views are interactive and proposed cell lineages could be efficiently reviewed and interactively corrected if needed. Our goal was for biologists to arrive at a final cell lineage more easily and with higher confidence. To better understand whether our design actually meets these goals, we also conducted an evaluation with five experts from the domain and described their feedback and our observations of their tool use.

Overall, in this work, we contribute (1) an interactive, web-based approach to establishing cell lineage, in which we combine 3D spatial and abstract data representations with correctable machine learning; (2) design considerations and decisions that led to our approach, including about the visual and interaction mappings that we used; (3) the approach of top-down and bottom-up hierarchy building and the interaction concepts that link the 3D and 2D representations at different scale levels; and (4) the evaluation with five experts to learn about the suitability of the employed approach, which also generated insights for applied visualization at large.

2. Related Work

Past work from three major fields is important for our own: the visual encoding of hierarchical structures, the use of visualization techniques in biological applications, and the application of machine learning in visualization. We review these fields below.

2.1. Visualization of and with Hierarchies

Hierarchical structures are essential for organizing many types of data and exploring inner relationships [SHS11]. Such inner correlations can be, e. g., about classification [BHGK14, BPT*11], family development [BDF*10], and evolution [HTS09, LPCB04]. Over the past 40 years, much research has been conducted to design and compare different visual representations (e. g., treemaps, sundown charts, sunburst charts, and icicle plots) of both 2D and 3D tree structures (e. g., [RMC91, MGT*03, Sch11]), to analyze hierarchical data (e. g., [SIL05, TS07]), and to help with visual decision making (e. g., [ATS95]). Work has also been done to evaluate the different representations [WYM19] and to propose guidelines for future design in hierarchical trees [KHA10]. However, most of these 3D trees were designed to add one dimension to store more information [SKW*07, TON03, LSTD17]. Our collaborating biologists are most familiar with 2D node-linked diagrams for representing the hierarchy, while the actual cell shapes have to be interpreted in 3D views to assist them with making lineage decisions. We thus rely on a combination of both a 3D view and a 2D tree. One of the most closely related approaches is HyperLabels [KIK*21], which uses a model's hierarchy to navigate the 3D structures. While Kouřil et al. also combine the abstract hierarchy data with 3D shapes that represent the actual physical entities, their abstract data representations do not represent additional information such as, in our case, the plant embryos' temporal development.

The size of datasets is constantly increasing, so researchers have also explored methods to flexibly adjust the space a tree representation takes to increase its efficiency [LTX*20, BRZW17, Ahu86]. For treemaps, for example, Tu et al. [TS07] used a spiral layout to visualize the updates of the hierarchical data. Similarly, van de Wetering et al. [vdWKB20] redesigned the icicle plot to reclaim empty space, to avoid the situations where nodes deeper in the hierarchy have less space. These techniques are not applicable in our scenario because we need to show all nodes equally such that we do not have excess empty space. Moreover, we do not have the full hierarchy at the start, and biologists are not always certain about the cell lineage hierarchy root in the beginning.

Another important aspect of our hierarchies is that they represent temporal development, and researchers have visualized temporal data with trees before. For example, Carvalho et al. [dCMdM16] combined aspects of treemaps with a calendar, so that users could target data within a certain time slot. In our case, the division time is different for each cell and the former and latter stages should always be presented for reference, so that we cannot use filters to target at a single level. Shen [She98] also used index to mark the cell division within a time interval, but it is hard to read especially the tree is large. We thus encode the time periods with the layers.

In addition to innovative layout design, some work also focused on improving the interaction with trees to adapt the substructures and compare the results with the original data, which we also need in our work. AdaptiviTree [TSLR07] encoded real-time game status information and indicated changes by color. Other work [BMW16, BvH*11] used visual cues to allow viewers to compare different trees that were shown horizontally, with color-coded similarity metrics based on juxtaposed icicle plots. Because we use the trees proposed by ML only as an initial guess and there is no ground truth, we do not show multiple trees and simply rely on color coding to encode differences. Related to this approach is the encoding of the uncertainty of the input data, for which researchers have used, e. g., glyphs [LRCP07, YWL*20]. While these methods can handle diverse forms of uncertainty including locations [LRCP07] and subtree structures [LRCP07], we can also use color-coding to indicate the ML model's certainty with respect to its results.

2.2. Biology-related Visualization

Biologists often study complex patterns or processes. While various stakeholders have different levels of interest in the respective datasets [LPCB04], we focus on interactive tools for experts—specifically botanists interested in plant embryo development.

Within the biology domain, several visualization approaches and tools are currently being used. For example, Briggs et al. [BPT*11] combined dendrograms and heatmaps for the visualization of taxonomic diversity. Such representations, however, are usually not space-efficient [MR10]. Moreover, we need to show both abstract cell lineage data as well as the actual 3D shape of the entities. Numerous platforms (e. g., OsiriX [RSR04], Fiji ImageJ [SACF*12], and Avizo) are designed to display such multi-scale 3D data, and these are currently being used by our collaborating experts to establish the cell lineage (e. g., the TreeJ plugin for Fiji ImageJ). Within academic research, Leeuw et al. [DVV*00] explored the visualization of time-dependent confocal data. For visualizing inner cell structures, Mindek et al. [MKS*18] proposed a multi-pipeline for visualizing the organelle system. Similarly, other work [LLC*19, HMK*20, HKM*22] proposed different systems to interact with diverse scales of cell or genome datasets. In contrast to these approaches, however, we are not interested in the representation of spatial scale but in the temporal development in the

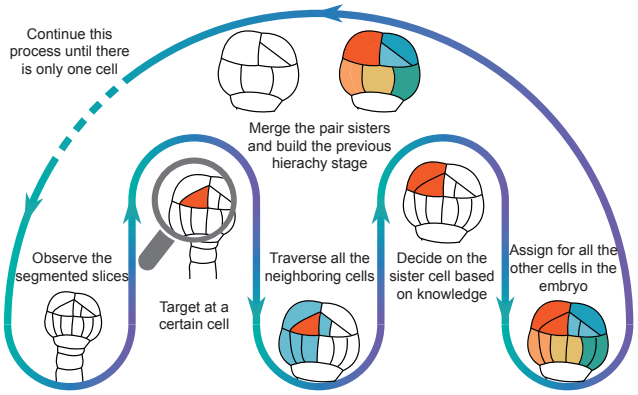

Figure 1: *A 2D illustration of biologists' cell lineage workflow.*

form of cell lineage. Of course, also general tools exists that could support the representation of 3D shape such as VTK [SLM04] which was used to create visual representations of biological data (e. g., [RSR04, SACF*12]) and ParaView [AGL05]. We also use the VTK library for creating the visual representations in our tool, but our emphasis is on the combination of 3D spatial and abstract data representations and their interactive use for the establishment of cell lineage for plant embryos.

2.3. Machine Learning in Visualization

In the visualization field, machine learning is used to create, improve, and assess all kinds of visualizations [ERT*17, WSZ*19, AWV*19]. Researchers also adopted machine learning models to develop pipelines for the whole process of visualizations [KMRV15, KAF*08]. In our case, we use a machine learning model in predicting assignments for a plant embryo and then visualize the predictions in the hierarchy. Unlike interactive machine learning (IML) where users could review and refine the model interactively [DK18] (i. e., an interactive correction of mislabeled data with the goal of getting a more accurate result from the model [XYX*19]), we treat the ML output only as a first suggestion and allow our users to directly interact with the output of the model for a more efficient correction. Also, researchers proposed different suggestions for designing such interfaces [SPG14, DK18, Hor99]. Inspired by these, we predict cell inheritance with a neural network model and display partial assignments to engage users [EFM16] and reduce manual labor.

3. Background and Design Considerations

As we just discussed, trees can be used to visualize cell hierarchies in embryos. Our collaborator also confirmed that biologists use and appreciate 3D views to assist them with understanding the spatial structure and to make decisions on the lineage. Here they rely on actual 3D meshes that they derive from the confocal microscopy slice data by segmenting and then processing them into the shapes of the embryo cells at the time of data capture. Yet, no suitable tools exist to determine cell lineage for several reasons:

A cell lineage tree only represents approximate division stages that apply to most cells, while some cells may remain unchanged from a given level to the next and the lineage hierarchy needs to account for such cases. Consequently, there exists no single generic

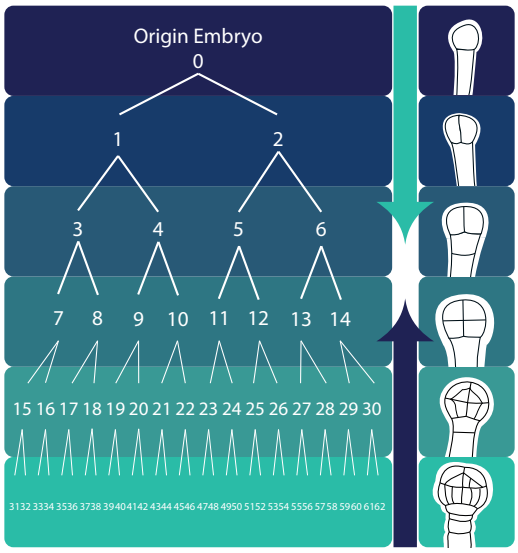

Figure 2: *An ideal 32-cell embryo cell lineage hierarchy, which can be established by reasoning bottom-up and top-down.*

(binary) tree that would represent the development of all embryos. In addition, the tree does not yet exist at the start of the process and cannot be computed from the data, instead, it has to be established step-by-step by the experts based on what they observe in the 3D spatial and abstract data. Figure 1 illustrates this process, in which the experts reason based on the embryo's configuration at a given time to establish how it developed into this stage from a previous set of cells, and then they repeat the process. Existing tools are usually created to focus either on 3D spatial or on abstract data representations, while in our case both types of representations are tightly coupled and biologists need both to make decisions (the spatial cell data as well as abstract data such as the hierarchy as established so far and information such as shared area—either type alone is not sufficient).

Based on all these constraints, we set out to design a new interactive tool for establishing and visualizing the development of plant embryos. To reduce the tediousness of having to make many lineage decisions for larger embryos, we targeted a process that relies at least partially on machine learning, yet also leaves the biologists in control and allows them to override decisions as needed. We also based our design on inspirations from existing tree representations such as icicle plots [WYM19]. Moreover, we followed the principles and advice on designing hierarchy structures [KHA10], cell visualization systems [DVV*00], and mixed-initiative user interfaces [Hor99]. Next, we summarize our design considerations.

D1: Provide an overview of the hierarchical structure to indicate the cell division process. We aim to create complete and accurate cell division hierarchies so that plant biologists can study the division history. Yet every cell divides in a different time and it is impossible to measure the precise division time from a single confocal microscopy dataset. So we rely on the fact that an embryo's cells roughly divide within similar time ranges, and use these division times to build the hierarchy. The tree should also clearly emphasize cells that divide outside of the regular time range. The tree thus not only needs to contain the parent and children relationships but

also represents the time-dependent division process. As such, any node in the tree represents a given cell and thus its connected 3D representation at a certain stage, so navigation in the hierarchy needs to lead to adjustments of the 3D visuals.

D2: Support the history building in two directions. Biologists traditionally build the tree in a bottom-up approach. From the overall shape of the embryo, however, they can also deduct how it divided in the very beginning. Introducing this knowledge early could improve the ML predictions and make it easier for the experts to specify assignments. It should thus be possible for biologists to build the tree from two directions: bottom-up and top-down (Figure 2).

D3: Show related information to assist with decision making. According to our collaborator, biologists take the cells' shape, the shared surface between two adjacent cells, their approximate volumes, and the 3D positions into consideration when picking the sister for a target cell from its neighbors, and we need to show all this information. While the spatial properties are most effectively shown in a 3D view, the experts need to be able to correlate them to the other data. While 2D diagrams would be possible, each cell only has a few neighbors and a color coding thus promises to be better than separate 2D diagrams. In addition, we need to account for cell occlusion in the 3D view, so the system needs to support visibility techniques such as explosion views and layer peeling.

D4: Provide assistance for building the hierarchy with machine learning (ML). Our previous work [HATI21] indicated that, even with an interactive system, traversing neighbors to find sisters for more than 200 cells is too time-consuming, and ML-based assignment suggestions could greatly improve the workflow. These predictions then need to be checked by the experts with the help of an integrated hierarchy and spatial 3D interaction. Any mistake, however, would invalidate any prediction for earlier cell divisions. The system should thus use an ML model that predicts the most likely cell pairs for a given hierarchy level, and bases future levels on verified or corrected assignments. The full power of our approach relies on a fluid change between the bottom-up ML-based prediction, expert correction, and the top-down hierarchy specification so that the expert is always in full control over the process. At any time, the hierarchy view should show the status of the lineage, indicate decisions confirmed from the ML or corrected by the human, and navigate the temporal development of the embryo.

D5: Provide easy access to biologists who are not computers professionals. Many experts may be hesitant or not allowed to install extra software on their workstations. Moreover, such local installations would require cross-platform system development. We thus aim for a web-based implementation that facilitates easy access, interaction, and data sharing among researchers [LLC*19].

4. System Design

Based on these considerations, we designed our web-based tool LineageD with two major components: a front-end for experts to interactively establish the lineages and a back-end that comprises a module to process the input data for the analysis, a set of default embryos for testing, a pre-trained classifier, and the partial lineage data currently being worked on (D4, D5).

4.1. Front-end Interface

Our web-based front end needs to allow the biologists to explore the 3D information in its hierarchy-dependent configuration, traverse the cells at the different hierarchy levels and understand their spatial and derived properties, make decisions about the cell lineage for all cells, and then record these decisions to specify further hierarchy levels. We thus offer the following elements to assist the experts.

Our LineageD system (Figure 3) centers around two main views: the **main 3D view** of the embryo and the **hierarchy view** that shows the cell lineage tree the biologists are building. The hierarchy on the right of the interface initially consists just of small blocks, where each represents a single cell at the time when the confocal microscopy dataset was captured; i. e., the leaf nodes of the tree (D1). We chose a tree layout that grows from top to bottom (i. e., root at the top) to match the traditional format used by the biologists (we illustrate the design process in Figure 4). As the experts work on establishing the hierarchy and declaring sister cells, this view actually becomes a hierarchical representation that we based on (completely filled) icicle plots. Unlike node-linked diagrams that take more space and are harder to navigate in a wide tree, this mapping has the benefit of establishing clear layers, which can then be navigated with the help of the slider on the left of the hierarchy view. We compute the widths of the leaf nodes based on their normalized volumes such that the biologists can directly see the relative sizes of the cells' volumes in the tree (D3). The icicle mapping then also nicely reflects that the volume of a mother cell is roughly the sum of the volumes of its daughters in consideration that the potential volume loss is not essential to biologists, and that all cells of a given level together always form the whole embryo. We chose the vertical direction for the hierarchy layering because the biologists are used to this layout.

The hierarchy view is tightly linked to the main 3D view on the left. In the latter, we show the whole embryo at the chosen division time, i. e., the level selected in the (partial) hierarchy view on the left. A regular 3D representation has the problem of the dense packing of the embryo cells, which results in inner cells being fully occluded by the outermost cell layer. To address this visibility issue we offer two visibility management techniques: explosion views (e. g., [LACS08, TKS11]; see Figure 3–main 3D view) and layer peeling (e. g., [SSSS11]; see Figure 5–3), both controllable through sliders. The 3D view also provides typical 3D navigation interaction, including zoom, translation, and rotation to allow biologists to explore the spatial character of the embryo.

We closely coupled both the 3D view and the hierarchy, not only through cross-linked selections. When we select a given cell in either view, we not only select it in the other as well but also show this cell and its currently assigned sister (if any) in a secondary 3D view on the top, the **target and sister view** that shares the same camera orientation as the main view. Here we highlight the selected (target) cell in red and its assigned sister in its original color, while we show the rest of the embryo's cells transparently to illustrate the target and sister cell' relative positions in context. Unlike in the main view, in the target and sister view we always show all cells in a tightly packed configuration and do not apply any explosion. This way we allow the biologists to understand the combined shape of the proposed cell match and make it easy for them to identify

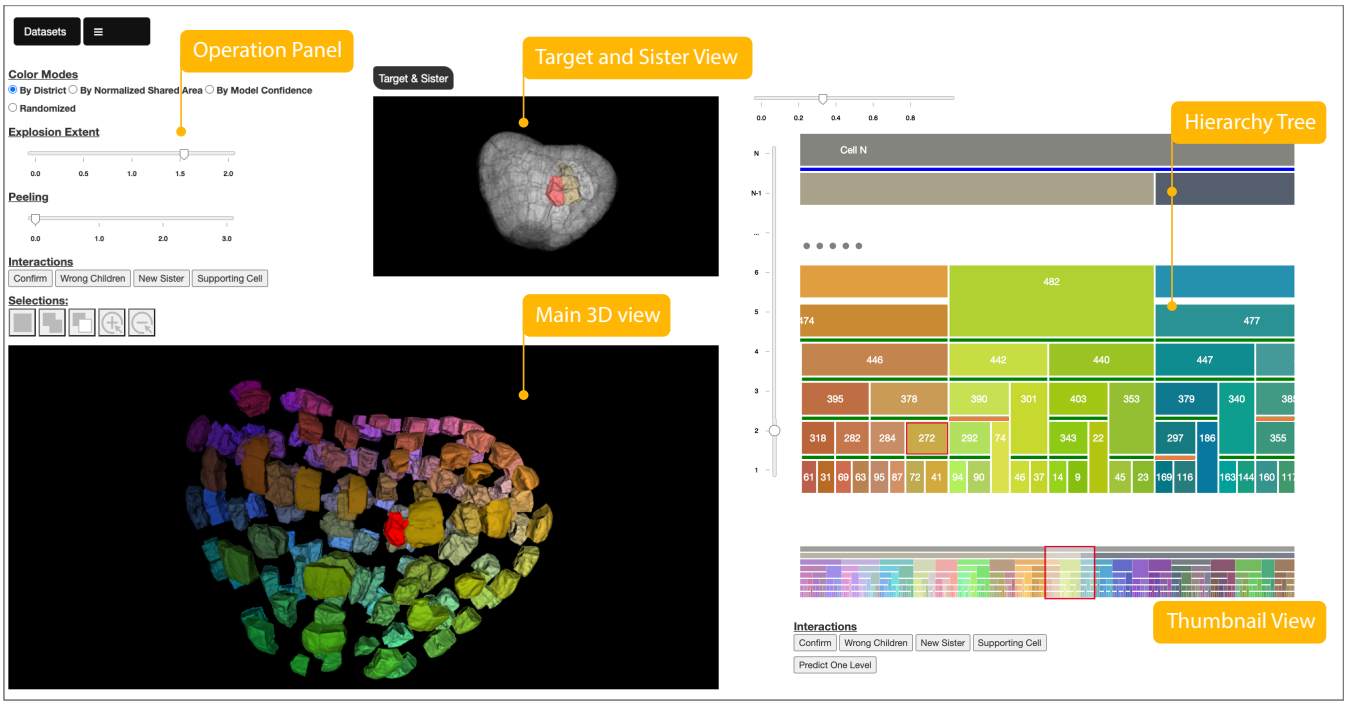

Figure 3: *Screenshot of LineageD, for the example of assigning a 256-cell embryo (in progress). The core elements are the* ***Main 3D View*** *of the embryo, here exploded and with the target cell highlighted, and the* ***Hierarchy Tree*** *of the lineage, which is interactively established by the biologists. The* ***Target and Sister View*** *shows the relative position of the target cell within the whole embryo and its proposed sister. The* ***Thumbnail View*** *of the tree provides the context of the full hierarchy. Finally, the* ***Operation Panel*** *supports further control of the tool.*

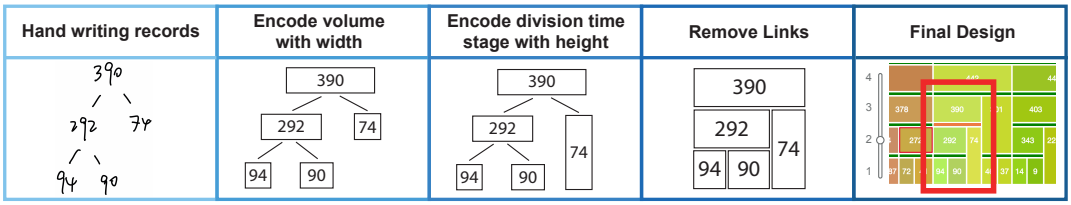

Figure 4: *An illustration of our reasoning for how we designed the lineage hierarchy tree representation, based on the traditional hand-written or (ASCII) text-based records used by the biologists.*

incorrect assignments. We specifically decided to only provide a single target and sister view. Initially we considered showing just the target and sister cell, a view of the target cell's children without context, as well as a view of all direct neighbors of the target cell. In a pilot study, however, we found that these views were confusing to our collaborator, and we settled on only showing the target and sister view as a secondary 3D view and moved some of the other functionality to be accessible on demand (see below).

We use the same color mapping in all views, both the 3D views and the hierarchy. We initially color-code both views by spatial location [OBCS*12], which we calculate based on the relative 3D positions of each cell's center of mass within the whole embryo, mapped to RGB. This mapping allows the users to clearly see if two cells are located close to each other, even in the abstract hierarchy view. During the later interaction we also offer color-coding by trust of the ML model, by the shared area between two matched cells, and a random color mapping that biologists are used to from their existing tools and that allows them to easily see the decomposition of the embryo into the cells of a given hierarchy level (D3).

Because the biologists can also use a top-down approach to establish the lineage (in addition to bottom-up), we need to represent both the top and the bottom section of the hierarchy at the same time—even if the tree has not been fully specified. We thus split the hierarchy into a **top tree** and a **bottom tree** (D2). For the top tree, we start with the root, i. e., the single *ovum* cell. As new layers are established top-down, we add them below the root. We cannot be sure about the layer number for the top-down layers at the beginning, so we named them N, N-1, etc., until we complete the hierarchy. We know all cell elements at the lowest level, so we can ensure that both representations correctly reflect this containment relationship. To ensure smooth layer navigation, as long as the hierarchy is not completed we introduce an in-between section between the top and the bottom tree marked by dots. We also add a corresponding layer entry to the hierarchy slider on the left, with newly established bottom-up

layers being added directly below it and newly established top-down layers being added directly above it.

Due to the exponential nature of cell division, the corresponding cell lineage tree can become very large horizontally very quickly, already for embryos with only a few generations of development. We thus provided two means to cope with the resulting view limitations. First, we provide a slider on the top of the hierarchy to control the horizontal zoom, allowing the users to see the whole tree in one view or to zoom into details. For the latter case we can ensure that the respective labels of the cells (numbers in our case) are visible, yet a viewer may lose the context of what part of the hierarchy they are examining. We thus also added a horizontal World-in-Miniature **thumbnail view** to make it easier to navigate the wide icicle plot hierarchy tree and which is often used in virtual reality (e. g., [SCP95, DEJW21]) and, more relevant to our case, in 2D video games (then often called "mini-map"; e. g., [Gek16]). This view, which we include below the hierarchy view, always shows the whole hierarchy and highlights the subsection that is currently visible in the detailed hierarchy view and facilitates navigation.

Finally, most of the controls for the interaction (which we discuss in more detail below) are then assembled in the **operations panel**. It also contains a menu to load new datasets or export results as well as instructions for users about our 3D interaction mappings.

4.2. Server-side Back-end

Our tool's server-side takes care of data processing and analysis. Our datasets comprise the cells' names (i. e., numeric labels) and mesh data for their surfaces. The experts create these datasets with dedicated tools (e. g., Avizo, Fiji [SACF*12]). In the back-end we use this input to build the necessary data structures to later be able to construct the hierarchy. We also derive the needed information such as the 3D cell locations (centers), neighbor counts and respective neighbor look-up tables, normalized cell volumes, normalized cell surfaces, and normalized shared areas between neighboring cells. After the completion of the lineage, the back-end also allows us to export the results in the tree data format needed by the biologists.

An essential part of the back-end is the creation of the ML model (D4). For this purpose we also maintain a collection of 97 past (manual) cell lineage assignments provided to us by the biologists. For these we also computed the data-derived parameters mentioned above and used this data to train a neural network model to allow us to predict the lineage for future datasets. The classifier computes a probability for all possible direct neighbors to be sister cells, and we then pick the one that has the highest chance. This reduces the number of possible pairs, and we continue picking the pair with the highest probability until this probability falls below 50%. This prediction functionality can then be accessed from the front-end.

4.3. Interaction Design

Our interaction design centers around the processes of proposing new matches of likely sister cells (manually and with the help of our neural network model) and the verification of whether these matches are correct. Both processes rely on abstract as well as 3D information, so our expert users have to constantly switch back and

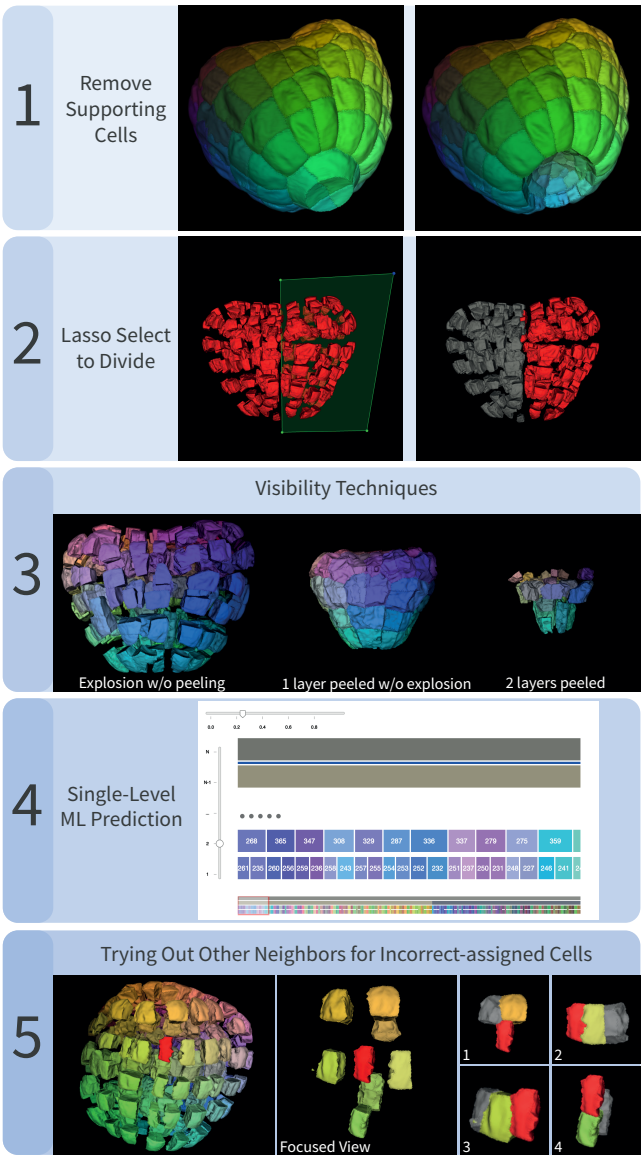

Figure 5: *Several interaction techniques in LineageD.*

forth between the abstract hierarchy view and the spatial 3D views. Next, we describe how, in LineageD, we support the biologists both in the top-down and the bottom-up lineage specification.

For a new dataset we show all cells as blocks on the lowest hierarchy level on the right, and in the 3D view we show the embryo with the cells colored based on their 3D location. Usually the first step of the biologists is to remove those cells from the consideration that are part of the *supporter* (i. e., those that connect the plant embryo with the rest of the seed) by selecting them in the 3D or hierarchy view and assigning them a respective label (Figure 5–1).

With the top-down specification we offer a new functionality not supported by the traditional tools. We take advantage of the fact that, for most embryos, the first few cell division generations happen in a predictable way—one that can still be reconstructed from the geometric shape of the embryo and its cell decomposition,

even at a later time. For this purpose biologists start to explore the shape of the embryo in the 3D view, usually via rotations. To better see and understand the inner structure, they can also use the explosion functionality. Then they decide on the most likely initial division—often a rather straight subdivision between the existing cells. We then allow them to mark one half using a spatial lasso selection [Luc05] (Figure 5-2) to register the selection as the first subdivision, which can also be further refined.

As they continue this top-down process, it is important to note that the interaction naturally relies on the selected level in the hierarchy. For example, to specify the third generation $N-2$ (i. e., the second split), the experts first select one of the daughters of the *ovum* in the hierarchy. This highlights all parts of the embryo that belong to this cell at the $N-1$ level in red, as if this was a single cell. Any potential explosion setting, however, applies to the level *currently* specified in the hierarchy just below; at the beginning this would be level 1.[†] The following lasso operation on a subset of these cells then selects them for the next split of only the selected cell, in a context-aware fashion [YEII16]. This step would then have to be repeated for splitting the second cell of the $N-1$ level. To mark these fully human-decided divisions, we display blue bars between the divided parent cells and their children in the hierarchy tree Figure 3. Naturally, this top-down process only works for a few generations, but it provides valuable constraints both for a manual and for the ML-supported process that we describe next.

For the bottom-up process we follow the biologists' traditional work process, as illustrated in Figure 1. They would start to traverse all cells at the lowest level and specify likely matches by examining each selected cell's direct neighbors to find its most likely sister. To reduce this workload, we can now use our neural network model and make ML-based predictions for the lineage. Initially we had used this process to predict the entire lineage, but in pilot studies it quickly turned out that this approach was not ideal. The experts disagreed with certain mappings, rendering the entire remainder of the path toward the root node obsolete when they corrected the mistakes. We thus now restrict the classifier to only predict the lowest non-specified hierarchy level (Figure 5-4).

After such a ML prediction step, the biologists can traverse the newly found pairings and either confirm or reject them (D4). To do that they can select a daughter cell in the hierarchy or 3D view, which shows it in a red highlight color, with the rest of the cells retaining their colors according to the chosen color scale (Figure 5-5-left). In the target and sister view, we also show its currently assigned sister in color, while the rest is transparent. A single click on a mother cell[‡] is similar, only then we highlight the whole mother in red and show no sister. With these techniques the experts can assess the mother cell's shape and decide if it is plausible. To better understand the alternatives, the experts can also double-click in either the hierarchy or the 3D view on a daughter cell, which leads to the 3D view showing a focused view with only the direct neighborhood of the selected cell (Figure 5-5-center), while the target and sister view then only shows the cell and its currently specified sister. Upon a further click on one of the neighbors in the main 3D view we show this neighbor also in the target and sister view, along with a potentially assigned sister of the neighbor (Figure 5-5-right).

With these different mechanisms to see detailed views, the biologists then make a decision about whether the pairing is plausible or not. If the biologist then confirms a proposed mapping when a given mother cell is selected, we mark that pairing as confirmed by showing a green bar between the mother cell and its daughters in the hierarchy (Figure 3–right). If they disagree with the pairing, we break it and, for the time being, treat the cells as non-dividing cells for the chosen hierarchy transition. They can be left as such or can be re-assigned. In the latter case, biologists select one cell, show its neighborhood with a double-click, and then pick one of the neighbors to mark it as the new sister. To distinguish such human specifications from confirmed ML pairs, we mark them with an orange bar in the hierarchy between the mother and its daughters.

After examining all pairs of a given level and confirming or correcting them, the experts can then go on to predict the next level. We treat any pairing that has not been explicitly confirmed as implicitly confirmed in that case, and also mark it with a green bar. The process then continues until the whole hierarchy is specified, the top and bottom trees merge, and we can export the result.

4.4. Implementation Details

We implemented LineageD in *JavaScript* and *Express.js* (v. 4.17.1). For the front-end, we rely on *D3* [BOH11] to realize the hierarchy tree rendering and render the embryo in the 3D views with *vtk.js* (v. 19.0.4) [SML96]. We implemented the back-end in *Node.js* and used *MongoDB* to create our database. We used *TensorFlow*'s (v. 2.8.0) *Keras* library to train the multi-layer sequential neural network model offline with *Python*. We created two hidden layers with *ReLU* activation and adopted *softmax* for the final output layer to get the probability score, and the model's single-pair accuracy reaches approximately 94%. The raw data is produced by *Avizo* and contains the cells' information including names, vertices, and triangles. We then parsed the raw shape data and set each neighboring pair as one record. For each pair, we extracted features for the training data based on the recommendations by the biologists: the normalized distance between the cell pair, their normalized shared surface area, their neighbor counts, their normalized volumes, normalized surface area, and the directions of neighbors with target cells. With *TensorFlow.js* (v. 3.8.0), we used the trained model to predict assignments in the website back-end. For an embryo with 256 cells, such as one of our default datasets that we show in Figure 3, there are more than one million vertices with more than five million triangles. Depending on the network speed when connecting to the Web tool, typical datasets can thus be loaded in less than a minute and can be rendered at interactive speeds (approx. 20 fps for the 256-cell example on a macOS with AMD Radeon Pro and Intel UHD Graphics 630).

5. Empirical Case Study Evaluation

While we developed our overall approach and our LineageD tool based on conversations with our expert collaborator, we were also interested in feedback from independent plant biologists, by whom

[†] If the *ovum* cell was selected instead, however, after a first top-down division specification, the explosion would only separate the two daughter cells of the *ovum* to emphasize this position in the lineage tree.

[‡] For performing such a selection in the 3D view one has to first select the corresponding hierarchy level in the hierarchy view, e. g., via the slider.

we envisioned our tool to be used. For this purpose it does not make much sense to run a quantitative experiment with many participants because, first, we do not have a reasonable alternative technique to which to compare our approach and, second, we would not find enough participants skilled in the traditional lineage process to draw meaningful conclusions. Instead we conducted case study evaluations [KPP95] with expert participants, as it has been done in the past in the visualization field (e. g., [LRF*11, KGP*12, HRD*19, WBR*20]). The benefit of this approach is that experts are trained in the traditional cell lineage process and can thus assess our new tool based on their experience.

In this section we present the overall experimental setup, our observations, and the user feedback from the biologists interacting with the system. The specific embryo we used in the study was a real *Arabidopsis thaliana* dataset provided by our collaborators, which comprises 64 cells (not counting the cells of the supporter). Datasets that are needed to be analyzed in practice typically range from several tens of cells to several hundreds of cells, so this example dataset is realistic (and still manageable in our study) as well as already challenging to analyze for biologists, and they could compare this experience to their past work. The whole investigation was approved by our institution's (Inria) ethical review board (AVIS n° 2021-46) and we pre-registered the study setup and design (`osf.io/rhyg4`).

5.1. Participants

We recruited 5 expert biologists (2 females, 3 males, ages 31–54 years, mean 43.4 years) via social networking and mailing lists, denoted as P1–P5. We offered prospective biologists free access to our tool as compensation. Before the study, we asked them to fill in a consent form, a research media records release form, and a background and demographic data collection form. All of the participants had a PhD degree and have been doing post-PhD research on plant biology or related fields for 2.5–25 years (mean: 14.1 years, sd: 9.4 years, median: 12 years). Two of them were experts in plant biology, while the other ones focus on computational biology, biomathematics, or bio-images processing. We conducted our study via videoconferencing due to the COVID19 limitations.

5.2. Procedure

Our case study experiment consisted of two parts: first we conducted an observational study of the experts using our tool with the example data, followed by a survey and a semi-structured interview.

Observational study. We conducted the experiment with each participant individually. We began by presenting a brief introduction about the features of LineageD. Next, we asked the participant to open the website on Chrome using the credentials we provided and to share their screen. We then briefly explained to them the components of LineageD and introduced the main interaction mechanisms, as explained in Section 4. Then the participant could freely explore and establish the lineage for the 64-cell dataset. We used the think-aloud protocol and with the participant's permission, we recorded the screen and audio during the process for our later analysis.

Survey and semi-structured interview. To better understand the usability of the tool and avenues for future improvement, we distributed a questionnaire which was designed based on the System Usability Scale (SUS) [HN93]. We also conducted a semi-structured post-study interview to understand our expert participants' usage experience and insights they may have gained.

5.3. Study results

A whole participant session took 60 to 100 minutes. All biologists reported that the 64-cell embryo was challenging for them to assign, and none of them finished the complete hierarchy history reconstruction within the study session. This was also not to be expected since they had to learn and get used to the new tool and its interaction concept first as well as answer our questions in the experiment. All of the participants, however, said that they plan to work with the tool more after the study and try it in practice.

Learning the Functionality. We found that three of our participating experts traditionally used 2D slice-based interaction (i. e., in TreeJ) to navigate the embryo dataset and to mark sister cells, as opposed to using 3D interaction with a surface-based dataset that we offer. This means that participants first had to get used to the fundamental 3D interaction techniques (rotation, translation, scaling) in projected 3D views. They also traditionally used text-based node-link representations of the hierarchy (see Appendix A), so they had to get used to the icicle plot representation we used. Web-based implementations as we used for LineageD were also new to one participant: P1 thought that the tool is quite different from the traditional software he is using. He stated that they generally use desktop software, but he believes that with sufficient time to play with it, most people could learn the new tool quickly. A final aspect was that P3 said that it was difficult for her to learn the actions to expect from the system based on specific input, such as what part of the tree to which the prediction applied. Overall, these observations meant that the training phase in our experiment was relatively long for four out of the five participants, but we expect that with more training, they can get used to the new interaction concepts.

Interaction Design. After the training we observed how the experts freely specified the lineage of the cells in the embryo. While they all used different strategies, they all heavily relied on both the 2D abstract tree and the 3D views. Four of them primarily used the hierarchy view to select the cells for checking the predicted assignments. They used the 3D view only as a source to understand the cells' shapes and their locations within the embryo, rather than to select the cells in this view. P2, in contrast, used the hierarchy and the 3D views evenly to select the cells to check. P1 and P4 explicitly stated that they appreciated the linking between the tree and 3D views. Moreover, P2, P4, and P5 specifically expressed that the approach to creating the hierarchy tree in a top-down fashion was very interesting to them—as they do not have this ability with their traditional tools. For example, P4 suggested that, for large embryos, biologists could specify the lineage separately based on the top-down divisions or look only at a subset of the top tree as a form of grouping to reduce the complexity. In addition, P2 and P4 mentioned that the peeling and explosion functions have a great potential to help them with their decision making. Overall, P1, P2, and P4 stated that using the tool has the potential to reduce assigning time and possible errors, despite requiring about double the clicks as the traditional approach according to P1. This applies especially to embryos with more than 20 cells because it is very difficult to do it manually. P2 also believed that LineageD could improve his confidence in assignments.

Interface and Visuals. All participants stated that they appreciated the visual representations, both 3D and 2D. They also liked, for example, the mechanisms for encoding the cell volumes in block widths and the coloring by shared surfaces or location. Overall, P1 considered LineageD much more powerful than any tool that they currently have, especially in selecting cells from both 3D environments and the tree. P2 specifically appreciated the target and sister view that shows the proposed pairing as well as the transparent context. He liked the representations and interaction design of the tree, but also thought it may take time for biologists because they are used to node-link trees. He also added that some other functionality may need to be adjusted further to the mental concepts of the biologists, but thinks this is a normal process that he would expect for most tools. Overall, all comments about the design were on the positive side. For instance, P3 thought that all individual components are needed and easy to use and she liked the hover effects in the hierarchy tree. P4 was happy to see that the tool is versatile, because he could use peeling and explosion independently. P5, finally, liked the hierarchy tree and thought all the options are good.

Understanding the Embryo's Development. We also asked biologists whether the tool helps them to better understand the embryo's development. All participants agreed that this is the case. P1 envisioned that, with the hierarchy tree, he could perform quantification, data analysis, and create models because we show a lot of additional information about or related to the lineage. P2 was excited about the possible insights and wanted to try other datasets with the tool, such as animal embryos. P3 thought that some visualizations, like the target and sister view, make it easy for her to observe the development of the embryo's organization over time. P4 said he better perceived certain features of the embryo with the information we presented in the tool, such as the cells' volumes and shared surfaces. P5, finally, said that 3D views are essential for getting a good understanding of the embryo.

Machine Learning Experience. Next, we inquired whether our participants think that the automatic assignment based on machine learning helps them with the lineage process. P1 and P4 thought they cannot fully evaluate it because the experience time was not enough for assigning big embryos. Nonetheless, P1 and P5 reported that the ML model was essential and that it worked well for small embryos. P2 thought the machine learning can generally help a lot, but he would prefer the model to predict the lineage from both directions (top-down and bottom-up), rather than one direction here (bottom-up). P2 and P4 also were curious about how the model actually made the specific assignments. P3, interestingly, said that she expects the model to help even more for larger embryos with more cells. She was, however, confused about whether the prediction function is applied to the top-down or bottom-up approach, and like P2 also wanted a model to predict the top tree. She also recommended to pre-visualize the model predictions for biologists to choose from before integrating them into the lineage tree. P3 and P4 were not sure whether the time needed to check and correct the ML predictions would ultimately be less than manually specifying the lineage.

Improvements. We asked the experts what interactions or elements they missed in the interface. P1 and P4 would like to have functions to label or categorize the cells so that they would have more features to examine and analyze the situation. P1 and P3 also missed a function to temporarily hide selected cells to get access to the inner cells and to distinguish checked pairs from unchecked ones. Also, P3 wanted the ability to explicitly highlight the shared surface between two neighboring cells to help her to decide between two possible cell pairings. The System Usability Scale rating we received for LineageD was 68 on average, which is exactly the average SUS score [MMP12]. This means that we need to improve the overall usability of our tool, but this is to be expected for a research prototype. We will use the specific feedback that we received during the study to make these improvements.

6. Discussion

Based on the responses of our participants we can now summarize and reflect on the main take-aways from our work.

Influence of past interaction habits of experts. Our specific application domain required us to provide both abstract 2D and spatial 3D representations: the cell lineage hierarchy that the experts constructed along with data such as cell volumes or area of shared surfaces on the one side and the actual shapes of the 3D cells that make up the embryo on the other side. Interestingly, it turned out that most experts were not yet experienced with working with (projected) 3D representations, instead they used 2D slices from the segmented microscopy data to interactively mark cells. In addition, traditionally they do not use the hierarchy tree as an interaction proxy, and it only serves as the result of the lineage specification (also see Appendix A). This fact may be the reason that many of our experts were reluctant to work directly in the 3D view and, instead, primarily interacted with the abstract hierarchy as this somewhat resembled a version of their previous slice-based view (yet stripped from its spatial properties). More longitudinal studies will be needed to see if experts can learn the benefits of interacting in 3D space, and to take full advantage of the linked 2D and 3D views. Our observations, however, make this a likely development as some of our participants already understood the advantages of our 3D visibility techniques such as explosion and peeling.

Reliance on both 2D and 3D representations. Our specific application exemplifies a case where both 3D spatial and abstract information and, in particular, the tight connection between both is essential to solve a task. Nonetheless, both types of representations show different aspects. Only the tree view can show inheritance and, thus, only here can the biologists specify which cell division time stage they want to see. At the same time, only the 3D view can show aspects such as proposed sisters in a spatial context or the specific shape of a shared surface between two cells. Yet only together they provide enough information for the biologists to be able to solve their task. Moreover, unlike in many existing 3D interfaces, the navigation of the level in the abstract hierarchy actually has an impact on the shape composition itself (i. e., on the decomposition of the embryo into cells at the different hierarchy levels). In our case the actual compositional information is non-existent at the beginning and incomplete throughout the process, so the interaction effects change as further parts of the hierarchy are established. It is also interesting to note that, in both views, it is difficult to see all information for realistic datasets. In the 3D view we thus need to rely on techniques such as explosion and peeling, while in the 2D view we need the world-in-miniature thumbnail view.

Complexity of cell lineage. According to one of our participants, it is already very difficult for biologists to establish the lineage for embryos starting at a few tens of cells, especially when they want to reconstruct the hierarchy for several generations (i. e., from the observed generation to the very beginning of the division process). Our novel support of also top-down specification made this process easier for the experts, and all our participants took advantage of both ways of establishing the lineage. From their responses it seems that the top-down process makes it easier for them to control their interactions and have more confidence in the results, in particular since they deal with incomplete information most of the time. Nonetheless, further interactive assistance such as being able to focus and restrict operations (including the ML suggestions) to only subsets of the whole embryo, as suggested by some participants, may make the process even easier for them.

Generalization. Even though we focused on plant embryos, there is no reason that our findings could not also apply in a broader scope. In particular, biologists that study other kingdoms of life such as animals or fungi may similarly profit from our work directly. The most closely related past work on the depiction of parts may be research on the creation of assembly instructions [APH*03, LACS08, TKS11, GYL*13], even though the actual interactive control of such depiction was typically not discussed in much detail in such work. Other past work [KIK*21] specifically discusses the navigation of 3D data through hierarchy interaction, yet in our work we actually change the composition of the 3D scene through this input. We thus think that our interactive combination of abstract hierarchy and spatial 3D views can inform future visualization tools that need to support such inter-dependencies.

7. Limitations and Future Work

Both our specific realization of LineageD and our empirical evaluation have some limitations that we will discuss next. We only focus on major issues, while minor usability issues as highlighted by our study naturally also need to be addressed. First, the actual shape of the cells depicted in the 3D views is a result of an approximative surface reconstruction from the segmentation of the original volumetric data. We only relied on the shapes on the lowest level and merged them for higher stages in the hierarchy. It would be better to actually apply any new hierarchy data to the segmentation masks and then re-extract the cell shapes at the coarser levels, which we also plan to do in the future. Second, for our machine learning support we adopted a neural network model to provide predictions for biologists to interact with. Other machine learning models such as reinforcement learning agents, however, could also work in our scenario and we are interested in comparing different models in the future. Also, we are interested in improving the interactions with the model, like sorting the pairs with model confidence and enabling users to pick to decide. As suggested by the biologists, we could also look into showing the reasons for the suggestions of the ML model yet we are skeptical if this would lead to a helpful representation—at least for our current neural network model. Instead, a more promising way could be to take the biologists' manual corrections into account (which we currently do not do) and to propose and visualize alternative pairings, for the experts to pick from.

Our relatively short case study evaluation is also limited. First, we largely did not get past the initial learning curve for the experts. While this also showed some interesting aspects about their interaction habits, it would be very interesting to conduct a longer-term evaluation with our participants and see how their interaction changes over time. Such a longitudinal evaluation would also allow us to better understand the challenges and benefits of the 2D-plus-3D interface on a deeper level, leading to better design guidelines for future tools. Second, we also only used data of a single species in the experiment, and a longer-term evaluation would allow us to use a variety of stimuli and even non-standard cases such as the scans of mutant embryos—a current research topic in the domain.

8. Conclusion

Having developed LineageD together with a domain collaborator, we were surprised to see that most domain experts did not have even basic visual representations such as tree representations [Sch11] as interactive elements in their traditional tools. A similar situation happens in the top-down and bottom-up approach where biologists spend reasonable amount of time to get familiar with the interaction. Moreover, when we provided them with this tool along with a standard 3D representation of their data, they primarily interacted in the 2D abstracted hierarchy as opposed to the 3D spatial view that showed the relevant data (cells within the context of the embryo)—almost like an extension of their previous 2D interaction on segmentation slices. So one of the main results of our work seems to be that, as a community, we may need to spend more effort on studying how established representations (from our perspective) can actually be transferred to tools in our application domains. We plan to continue our work with the biologists to do just that, and to also see how LineageD can evolve over a longer time to better suit the needs of the biologists.

Another interesting takeaway is that we may need to look further into ways that machine learning can be integrated into our tools. It is not necessarily essential to get the best or most correct prediction from a model, but instead how we can provide means to verify or correct algorithmic suggestions, to speed up or make less tedious otherwise manual processes. For this approach it is not needed to completely understand why an ML technique made a specific suggestion, but instead to accept that ML is not always perfect and to provide means to quickly analyze results and adjust them.

Acknowledgments

We thank all participants in our evaluation for their input, the AVIZ team for their general feedback, and Hao Wu for technical advice. The work was partially supported by Inria's NaviScope project.

Images/graphs/plots license/copyright

With the exception of those images from external authors whose licenses/copyrights we have specified in the respective figure captions, we as authors state that all of our own figures, graphs, and plots in this article (i. e., those not marked) are and remain under our own personal copyright, with the permission to be used here. We also make them available under the Creative Commons Attribution 4.0 International (CC BY 4.0) license and share them at `osf.io/rhyg4`.

References

[AGL05] AHRENS J., GEVECI B., LAW C.: Paraview: An end-user tool for large data visualization. In *The Visualization Handbook*, Hansen C. D., Johnson C. R., (Eds.). Elsevier, Oxford, UK, 2005, ch. 36, pp. 717–732. doi:10.1016/B978-012387582-2/50038-1.

[Ahu86] AHUJA N.: Efficient planar embedding of trees for VLSI layouts. *Comput Vision Graph Image Process 34*, 2 (May 1986), 189–203. doi:10.1016/S0734-189X(86)80058-5.

[APH*03] AGRAWALA M., PHAN D., HEISER J., HAYMAKER J., KLINGNER J., HANRAHAN P., TVERSKY B.: Designing effective step-by-step assembly instructions. *ACM Trans Graph 22*, 3 (July 2003), 828–837. doi:10.1145/882262.882352.

[ATS95] ASAHI T., TURO D., SHNEIDERMAN B.: Visual decision-making: Using treemaps for the analytic hierarchy process. In *Proc. CHI* (1995), ACM, New York, pp. 405–406. doi:10.1145/223355.223747.

[AWV*19] AMERSHI S., WELD D., VORVOREANU M., FOURNEY A., NUSHI B., COLLISSON P., SUH J., IQBAL S., BENNETT P. N., INKPEN K., ET AL.: Guidelines for human-AI interaction. In *Proc. CHI* (2019), ACM, New York, pp. 3:1–3:13. doi:10.1145/3290605.3300233.

[BDF*10] BEZERIANOS A., DRAGICEVIC P., FEKETE J.-D., BAE J., WATSON B.: GeneaQuilts: A system for exploring large genealogies. *IEEE Trans Vis Comput Graph 16*, 6 (Oct. 2010), 1073–1081. doi:10.1109/TVCG.2010.159.

[BHGK14] BEHAM M., HERZNER W., GRÖLLER M. E., KEHRER J.: Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE Trans Vis Comput Graph 20*, 12 (Dec. 2014), 1693–1702. doi:10.1109/TVCG.2014.2346626.

[BMW16] BECK F., MELCHER J., WEISKOPF D.: Identifying modularization patterns by visual comparison of multiple hierarchies. In *Proc. ICPC* (2016), IEEE CS, Los Alamitos, pp. 9:1–9:10. doi:10.1109/ICPC.2016.7503712.

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3: Data-driven documents. *IEEE Trans Vis Comput Graph 17*, 12 (Dec. 2011), 2301–2309. doi:10.1109/TVCG.2011.185.

[BPT*11] BRIGGS B. R., POHLMAN J. W., TORRES M., RIEDEL M., BRODIE E. L., COLWELL F. S.: Macroscopic biofilms in fracture-dominated sediment that anaerobically oxidize methane. *Applied and Environmental Microbiology 77*, 19 (Oct. 2011), 6780–6787. doi:10.1128/AEM.00288-11.

[BRZW17] BURCH M., RASCHKE M., ZEYFANG A., WEISKOPF D.: A scalable visualization for dynamic data in software system hierarchies. In *Proc. VISSOFT* (2017), IEEE CS, Los Alamitos, pp. 85–93. doi:10.1109/VISSOFT.2017.16.

[BvH*11] BREMM S., VON LANDESBERGER T., HESS M., SCHRECK T., WEIL P., HAMACHERK K.: Interactive visual comparison of multiple trees. In *Proc. VAST* (2011), IEEE CS, Los Alamitos, pp. 31–40. doi:10.1109/VAST.2011.6102439.

[dCMdM16] DE CARVALHO M. B., MEIGUINS B. S., DE MORAIS J. M.: Temporal data visualization technique based on treemap. In *Proc. IV* (2016), IEEE CS, Los Alamitos, pp. 399–403. doi:10.1109/IV.2016.65.

[DEJW21] DANYLUK K., ENS B., JENNY B., WILLETT W.: A design space exploration of worlds in miniature. In *Proc. CHI* (2021), ACM, New York, pp. 122:1–122:15. doi:10.1145/3411764.3445098.

[DK18] DUDLEY J. J., KRISTENSSON P. O.: A review of user interface design for interactive machine learning. *ACM Trans Interact Intell Syst 8*, 2 (June 2018), 8:1–8:37. doi:10.1145/3185517.

[DVV*00] DE LEEUW W. C., VAN LIERE R., VERSCHURE P. J., VISSER A. E., MANDERS E. M. M., VAN DRIELF R.: Visualization of time dependent confocal microscopy data. In *Proc. VIS* (2000), IEEE CS, Los Alamitos, pp. 473–476. doi:10.1109/VISUAL.2000.885735.

[EFM16] EARLY K., FIENBERG S. E., MANKOFF J.: Test time feature ordering with FOCUS: Interactive predictions with minimal user burden. In *Proc. UbiComp* (2016), ACM, New York, pp. 992–1003. doi:10.1145/2971648.2971748.

[ERT*17] ENDERT A., RIBARSKY W., TURKAY C., WONG B. W., NABNEY I., BLANCO I. D., ROSSI F.: The state of the art in integrating machine learning into visual analytics. *Comput Graph Forum 36*, 8 (Dec. 2017), 458–486. doi:10.1111/cgf.13092.

[Gek16] GEKKER A.: (Mini) mapping the game-space: A taxonomy of control. In *Playfyl Mapping in the Digital Age*, Wilmott C., Perkins C., Lammes S., Hind S., Gekker A., Fraser E., Evans D., (Eds.). Institute of Network Cultures, Amsterdam, 2016, ch. 8, pp. 134–155. URL: https://hdl.handle.net/20.500.12657/29626.

[GYL*13] GUO J., YAN D.-M., LI E., DONG W., WONKA P., ZHANG X.: Illustrating the disassembly of 3D models. *Comput Graph 37*, 6 (Oct. 2013), 574–581. doi:10.1016/j.cag.2013.05.020.

[HATI21] HONG J., ARGELAGUET F., TRUBUIL A., ISENBERG T.: Design and evaluation of three selection techniques for tightly packed 3D objects in cell lineage specification in botany. In *Proc. GI* (2021), CHCCS, Mississauga, ON, Canada, pp. 213–223. doi:10.20380/GI2021.33.

[HKM*22] HALLADJIAN S., KOUŘIL D., MIAO H., GRÖLLER M. E., VIOLA I., ISENBERG T.: Multiscale unfolding: Illustratively visualizing the whole genome at a glance. *IEEE Trans Vis Comput Graph 28* (2022). To appear. doi:10.1109/TVCG.2021.3065443.

[HMK*20] HALLADJIAN S., MIAO H., KOUŘIL D., GRÖLLER M. E., VIOLA I., ISENBERG T.: ScaleTrotter: Illustrative visual travels across negative scales. *IEEE Trans Vis Comput Graph 26*, 1 (Jan. 2020), 654–664. doi:10.1109/TVCG.2019.2934334.

[HN93] HARPER B. D., NORMAN K. L.: Improving user satisfaction: The questionnaire for user interaction satisfaction version 5.5. In *Proc. Mid-Atlantic Human Factors Conference* (1993), pp. 224–228.

[Hor99] HORVITZ E.: Principles of mixed-initiative user interfaces. In *Proc. CHI* (1999), ACM, New York, pp. 159–166. doi:10.1145/302979.303030.

[HRD*19] HURTER C., RICHE N. H., DRUCKER S. M., CORDEIL M., ALLIGIER R., VUILLEMOT R.: FiberClay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Trans Vis Comput Graph 25*, 1 (Jan. 2019), 704–714. doi:10.1109/TVCG.2018.2865191.

[HTS09] HORN M. S., TOBIASZ M., SHEN C.: Visualizing biodiversity with Voronoi treemaps. In *Proc. ISVD* (2009), IEEE CS, Los Alamitos, pp. 265–270. doi:10.1109/ISVD.2009.22.

[KAF*08] KEIM D., ANDRIENKO G., FEKETE J.-D., GÖRG C., KOHLHAMMER J., MELANÇON G.: Visual Analytics: Definition, process, and challenges. In *Information Visualization*. Springer, Berlin, 2008, pp. 154–175. doi:10.1007/978-3-540-70956-5_7.

[KGP*12] KLEIN T., GUÉNIAT F., PASTUR L., VERNIER F., ISENBERG T.: A design study of direct-touch interaction for exploratory 3D scientific visualization. *Comput Graph Forum 31*, 3 (June 2012), 1225–1234. doi:10.1111/j.1467-8659.2012.03115.x.

[KHA10] KONG N., HEER J., AGRAWALA M.: Perceptual guidelines for creating rectangular treemaps. *IEEE Trans Vis Comput Graph 16*, 6 (Nov./Dec. 2010), 990–998. doi:10.1109/TVCG.2010.186.

[KIK*21] KOUŘIL D., ISENBERG T., KOZLÍKOVÁ B., MEYER M., GRÖLLER M. E., VIOLA I.: HyperLabels—Browsing of dense and hierarchical molecular 3D models. *IEEE Trans Vis Comput Graph 27*, 8 (Aug. 2021), 3493–3504. doi:10.1109/TVCG.2020.2975583.

[KMRV15] KEIM D. A., MUNZNER T., ROSSI F., VERLEYSEN M.: *Bridging Information Visualization with Machine Learning*. Dagstuhl Seminar Report 15101, Germany, 2015. doi:10.4230/DagRep.5.3.1.

[KPP95] KITCHENHAM B., PICKARD L., PFLEEGER S. L.: Case studies for method and tool evaluation. *IEEE Softw 12*, 4 (July 1995), 52–62. doi:10.1109/52.391832.

[LACS08] LI W., AGRAWALA M., CURLESS B., SALESIN D.: Automated generation of interactive 3D exploded view diagrams. *ACM Trans Graph 27*, 3 (Aug. 2008), 101:1–101:7. doi:10.1145/1360612.1360700.

[LLC*19] LEGGIO B., LAUSSU J., CARLIER A., GODIN C., LEMAIRE P., FAURE E.: MorphoNet: An interactive online morphological browser to explore complex multi-scale data. *Nat Commun 10*, 1 (June 2019), 2812:1–2812:8. doi:10.1038/s41467-019-10668-1.

[LPCB04] LEE B., PARR C. S., CAMPBELL D., BEDERSON B. B.: How users interact with biodiversity information using TaxonTree. In *Proc. AVI* (2004), ACM, New York, pp. 320–327. doi:10.1145/989863.989918.

[LRCP07] LEE B., ROBERTSON G. G., CZERWINSKI M., PARR C. S.: CandidTree: Visualizing structural uncertainty in similar hierarchies. *Inf Vis 6*, 3 (Dec. 2007), 233–246. doi:10.1145/1375939.1375944.

[LRF*11] LUNDSTRÖM C., RYDELL T., FORSELL C., PERSSON A., YNNERMAN A.: Multi-touch table system for medical visualization: Application to orthopedic surgery planning. *IEEE Trans Vis Comput Graph 17*, 12 (Dec. 2011), 1775–1784. doi:10.1109/TVCG.2011.224.

[LSTD17] LIMBERGER D., SCHEIBEL W., TRAPP M., DÖLLNER J.: Mixed-projection treemaps: A novel approach mixing 2D and 2.5D treemaps. In *Proc. IV* (2017), IEEE CS, Los Alamitos, pp. 164–169. doi:10.1109/iV.2017.67.

[LTX*20] LI G., TIAN M., XU Q., MCGUFFIN M. J., YUAN X.: Tree Illustrator: Interactive construction of tree visualizations. In *CHI Extended Abstracts* (2020), ACM, New York, pp. 1–4. doi:10.1145/3334480.3383150.

[Luc05] LUCAS J. F.: *Design and Evaluation of 3D Multiple Object Selection Techniques*. Master's thesis, Virginia Polytechnic Institute and State University, USA, 2005. URL: https://hdl.handle.net/10919/31769.

[MGT*03] MUNZNER T., GUIMBRETIÈRE F., TASIRAN S., ZHANG L., ZHOU Y.: TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans Graph 22*, 3 (July 2003), 453–462. doi:10.1145/882262.882291.

[MKS*18] MINDEK P., KOUŘIL D., SORGER J., TOLOUDIS D., LYONS B., JOHNSON G., GRÖLLER M. E., VIOLA I.: Visualization multi-pipeline for communicating biology. *IEEE Trans Vis Comput Graph 24*, 1 (Jan. 2018), 883–892. doi:10.1109/TVCG.2017.2744518.

[MMP12] MCLELLAN S., MUDDIMER A., PERES S. C.: The effect of experience on System Usability Scale ratings. *Journal of usability studies 7*, 2 (2012), 56–67.

[MR10] MCGUFFIN M. J., ROBERT J.-M.: Quantifying the space-efficiency of 2D graphical representations of trees. *Inf Vis 9*, 2 (June 2010), 115–140. doi:10.1145/1890886.1890889.

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: A flexible representation of maps between shapes. *ACM Trans Graph 31*, 4 (July 2012), 30:1–30:11. doi:10.1145/2185520.2185526.

[RMC91] ROBERTSON G. G., MACKINLAY J. D., CARD S. K.: Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. CHI* (1991), ACM, New York, pp. 189–194. doi:10.1145/108844.108883.

[RSR04] ROSSET A., SPADOLA L., RATIB O.: OsiriX: An open-source software for navigating in multidimensional DICOM images. *J Digital Imaging 17*, 3 (Sept. 2004), 205–216. doi:10.1007/s10278-004-1014-6.

[SACF*12] SCHINDELIN J., ARGANDA-CARRERAS I., FRISE E., KAYNIG V., LONGAIR M., PIETZSCH T., PREIBISCH S., RUEDEN C., SAALFELD S., SCHMID B., TINEVEZ J.-Y., WHITE D. J., HARTENSTEIN V., ELICEIRI K., TOMANCAK P., CARDONA A.: Fiji – An open-source platform for biological-image analysis. *Nat Methods 9*, 7 (July 2012), 676–682. doi:10.1038/nmeth.2019.

[Sch11] SCHULZ H.: Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications 31*, 6 (Nov./Dec. 2011), 11–15. doi:10.1109/MCG.2011.103.

[SCP95] STOAKLEY R., CONWAY M. J., PAUSCH R.: Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. CHI* (1995), ACM, New York, pp. 265–272. doi:10.1145/223904.223938.

[She98] SHEN H.-W.: Isosurface extraction in time-varying fields using a temporal hierarchical index tree. In *Proc. Visualization* (1998), IEEE CS, Los Alamitos, pp. 159–166. doi:10.1109/VISUAL.1998.745298.

[SHS11] SCHULZ H., HADLAK S., SCHUMANN H.: The design space of implicit hierarchy visualization: A survey. *IEEE Trans Vis Comput Graph 17*, 4 (Apr. 2011), 393–411. doi:10.1109/TVCG.2010.79.

[SIL05] SHI K., IRANI P., LI B.: An evaluation of content browsing techniques for hierarchical space-filling visualizations. In *Proc. InfoVis* (2005), IEEE CS, Los Alamitos, pp. 81–88. doi:10.1109/INFVIS.2005.1532132.

[SKW*07] SCHEDL M., KNEES P., WIDMER G., SEYERLEHNER K., POHLE T.: Browsing the web using stacked three-dimensional sunbursts to visualize term co-occurrences and multimedia content. In *Posters of IEEE VIS* (2007). URL: http://www.cp.jku.at/people/schedl/Research/Publications/pdf/cob_vis_2007.pdf.

[SLM04] SCHROEDER W. J., LORENSEN B., MARTIN K.: *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, New York, 2004. URL: https://vtk.org/vtk-textbook/.

[SML96] SCHROEDER W., MARTIN K., LORENSEN W.: The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *Proc. Visualization* (1996), IEEE CS, Los Alamitos, pp. 93–100. doi:10.1109/VISUAL.1996.567752.

[SPG14] STOLPER C. D., PERER A., GOTZ D.: Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Trans Vis Comput Graph 20*, 12 (Dec. 2014), 1653–1662. doi:10.1109/TVCG.2014.2346574.

[SSSS11] SULTANUM N., SOMANATH S., SHARLIN E., SOUSA M. C.: "Point it, split it, peel it, view it" Techniques for interactive reservoir visualization on tabletops. In *Proc. ITS* (2011), ACM, New York, pp. 192–201. doi:10.1145/2076354.2076390.

[TKS11] TATZGERN M., KALKOFEN D., SCHMALSTIEG D.: Multi-perspective compact explosion diagrams. *Comput Graph 35*, 1 (Feb. 2011), 135–147. doi:10.1016/j.cag.2010.11.005.

[TON03] TANAKA Y., OKADA Y., NIIJIMA K.: Treecube: Visualization tool for browsing 3D multimedia data. In *Proc. IV* (2003), IEEE CS, Los Alamitos, pp. 427–432. doi:10.1109/IV.2003.1218020.

[TS07] TU Y., SHEN H.: Visualizing changes of hierarchical data using treemaps. *IEEE Trans Vis Comput Graph 13*, 6 (Nov./Dec. 2007), 1286–1293. doi:10.1109/TVCG.2007.70529.

[TSLR07] TAN D., SMITH G., LEE B., ROBERTSON G.: AdaptiviTree: Adaptive tree visualization for tournament-style brackets. *IEEE Trans Vis Comput Graph 13*, 6 (Nov./Dec. 2007), 1113–1120. doi:10.1109/TVCG.2007.70537.

[vdWKB20] VAN DE WETERING H., KLAASSEN N., BURCH M.: Space-reclaiming icicle plots. In *Proc. PacificVis* (2020), IEEE CS, Los Alamitos, pp. 121–130. doi:10.1109/PacificVis48177.2020.4908.

[WBR*20] WANG X., BESANÇON L., ROUSSEAU D., SERENO M., AMMI M., ISENBERG T.: Towards an understanding of augmented reality extensions for existing 3D data analysis tools. In *Proc. CHI* (2020), ACM, New York, pp. 528:1–528:13. doi:10.1145/3313831.3376657.

[WSZ*19] WANG Y., SUN Z., ZHANG H., CUI W., XU K., MA X., ZHANG D.: Datashot: Automatic generation of fact sheets from tabular data. *IEEE Trans Vis Comput Graph 26*, 1 (Jan. 2019), 895–905. doi:10.1109/TVCG.2019.2934398.

[WYM19] WOODBURN L., YANG Y., MARRIOTT K.: Interactive visualisation of hierarchical quantitative data: An evaluation. In *Proc. VIS* (2019), IEEE CS, Los Alamitos, pp. 96–100. doi:10.1109/VISUAL.2019.8933545.

[XYX*19] XIANG S., YE X., XIA J., WU J., CHEN Y., LIU S.: Interactive correction of mislabeled training data. In *Proc. VAST* (2019), IEEE CS, Los Alamitos, pp. 57–68. doi:10.1109/VAST47406.2019.8986943.

[YEII16] YU L., EFSTATHIOU K., ISENBERG P., ISENBERG T.: CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 886–895. doi:10.1109/TVCG.2015.2467202.

[YWL*20] YANG W., WANG X., LU J., DOU W., LIU S.: Interactive steering of hierarchical clustering. *IEEE Trans Vis Comput Graph 27*, 10 (Oct. 2020), 46–50. doi:10.1109/TVCG.2020.2995100.

APPENDIX

Appendix A: Traditional Lineage Specification Tool

Figure 6 shows an example for a traditional tool (TreeJ plugin to Fiji ImageJ) to specify the cell lineage. Figure 7 shows the result of specifying the cell lineage for an *Arabidospsi thaliana* embryo of 122 cells. Figure 8 illustrates the overall traditional process further: we start from slices from confocal microscopy, segment this data, and then the biologists interact in the segmented slices rendered with ID colors to establish levels in the hierarchy.

Appendix B: Further Design Considerations for LineageD

Figure 4 illustrates how we arrived at our final design of the representation of the cell lineage hierarchy. We started from the hand-written (or text-based) records as used by the biologists in the traditional approach, encoded these as a traditional tree representation with the width of each node encoding the respective cell's volume, then elongated cells vertically to indicate the time period in which they existed in this configuration, and then removed the links between the nodes by arranging them in an icicle plot manner. The final step was then the addition of a color scheme and the interaction means as presented in the paper.

Figure 9 illustrates the linked interaction between hierarchy and 3D views in more detail. The first row shows that, upon clicking on a cell without a specified parent, the main 3D view highlights this cell and the target and sister view shows it in context. In contrast, if this cell already has a sister assigned (proposed by the neural network model or manually specified), then this sister is also highlighted in the target and sister view.

Appendix C: Availability of LineageD and Video Figure

The LineageD project website is available at `aviz.fr/Research/CellLineage`. Nonetheless, we realize that LineageD runs on our own servers and so we understand that resviewers may be concerned about maintaining their anonymity (even though we do not log any access to the tool; any logs that we used in our experiment were collected on the local machines of the participants). We thus also make a video figure available for reviewers that shows the main functionality and interaction that we discuss in the paper, and plan to make LineageD openly available once our paper has been accepted for publication. We also plan to make the sources available on a public repository such as GitHub to make it possible for others to also run a local copy of LineageD.

Appendix D: Typical Interaction Process

With LineageD, biologists first need to select their dataset of interest from the menu. After the dataset is loaded, we show the whole embryo overview in the main 3D view, one default ovum cell at the top of the tree, and all the leave cells displayed at the bottom of the tree (Figure 10–1). Then biologists can observe the whole embryo, hide the supporters, and try to divide it into parts to make further assignments easier. To do that, they need to target the tree's top cell (the ovum), and with the polygon selection technique select a group of cells that they are certain arose from the same cell.

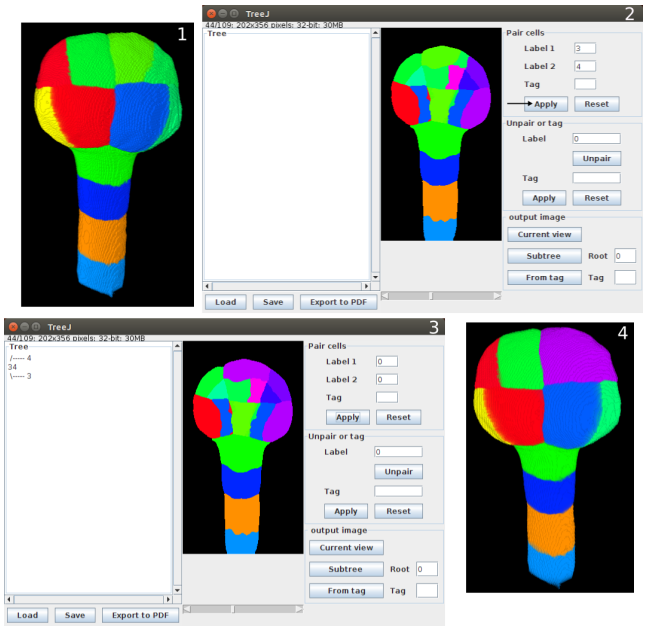

Figure 6: *Screenshots of the interface in TreeJ (a plugin to Fiji ImageJ) that the experts traditionally use to specify the cell lineage. Users primarily interact in a 2D slice of the microscopy data, colored by the cell's assigned ID, to select a pair of cells to merge. The resulting tree is then represented in a textual form with numbers as cell labels, and it is no interactive. The 3D view (marked 1 and 4) only serves as an additional view and is also not used for interaction. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.*

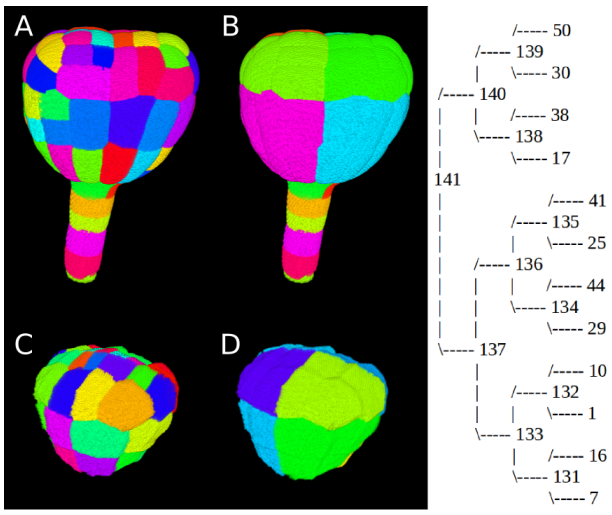

Figure 7: *Example cell lineage result produced with TreeJ that shows several development stages (A–D) of the cell in the 3D view and the corresponding text-based lineage hierarchy. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.*

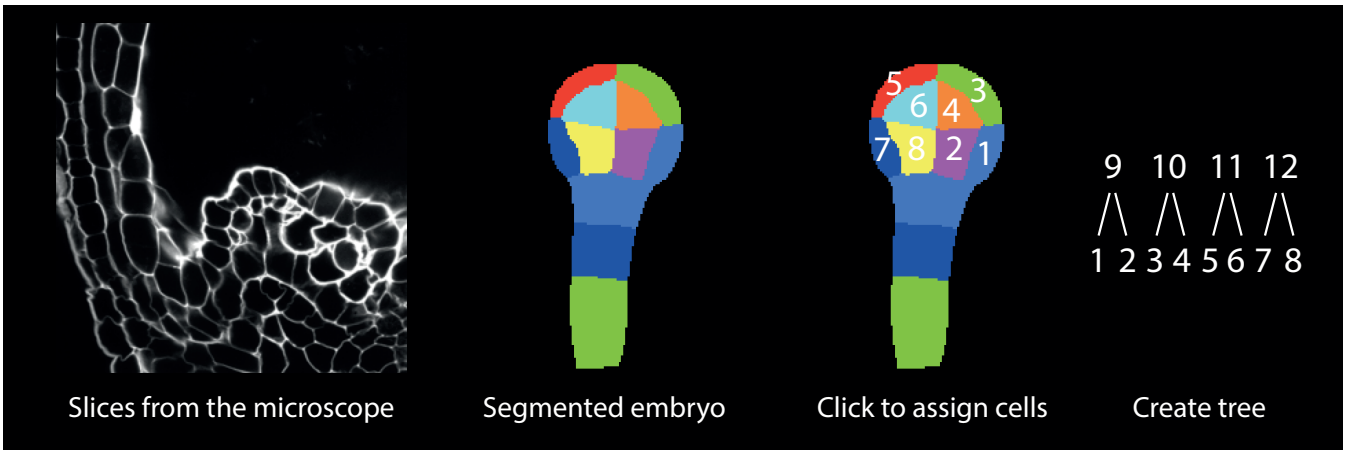

Figure 8: *An illustration of the traditional way of biologists doing cell lineage based on slices. The image insets in this figure are © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.*

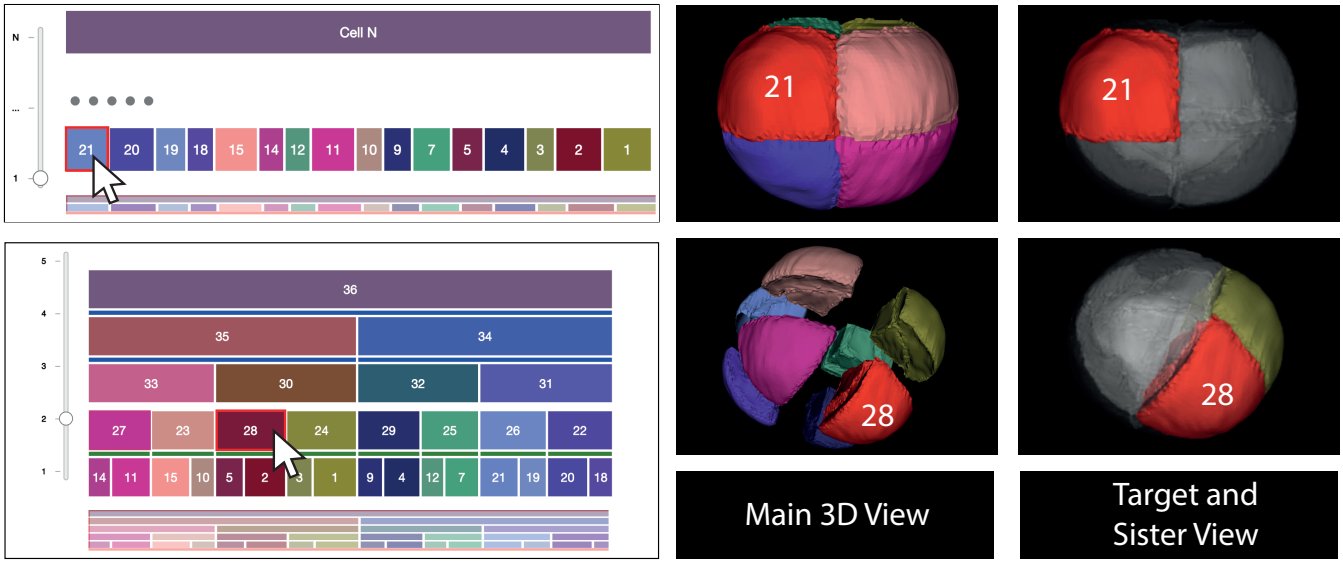

Figure 9: *Illustration of the interaction that links selections in the 2D hierarchy tree to the representations in the 3D view, including the highlighting of the selected cells and potential neighbors.*

They can adjust these divisions by selecting other cells to add to or subtract from the current selection. As they finalize the selection, the tree automatically is extended and a new daughter cell is added below the ovum (Figure 10–2). The biologists then continue this process until no further top-down decisions can be made, with the result that two or more layers at the top have been established (Figure 10–3). After the high-level divisions, biologists can use the ML model to predict a higher level assignment on the current tree. Next, they check the newly formed parent cells, and if they agree with the prediction results, they confirm the assignment. They can also disagree with the assignments and reassign a correct sister for the wrongly assigned cells. For a correction, they double-click the cell that has a wrongly proposed sister. Then, they traverse and single-click all the neighboring cells to check their properties by examining the respective color maps and the shared surfaces. Based on all this information, they then choose the most likely sister based on their experience and knowledge and mark it as such (Figure 10–4). After checking all the pairs in the current level, the biologists then continue to work on the next stage. This way the tree is being completed bottom-up. Thus, when the top part of the tree and the bottom part meet, we automatically merge both to form the final lineage tree (Figure 10–5). Finally, biologists can use the export function to record the assignments in their preferred format.

Appendix E: Data Structure

For each embryo, we start with with a dataset that comprises the 3D geometry (i. e., mesh information and a list of cell IDs) extracted by Avizo from the confocal microscopy data. We clean this data to ensure that every mentioned cell name actually has a geometry assigned to it, ignoring those IDs without associated meshes. We

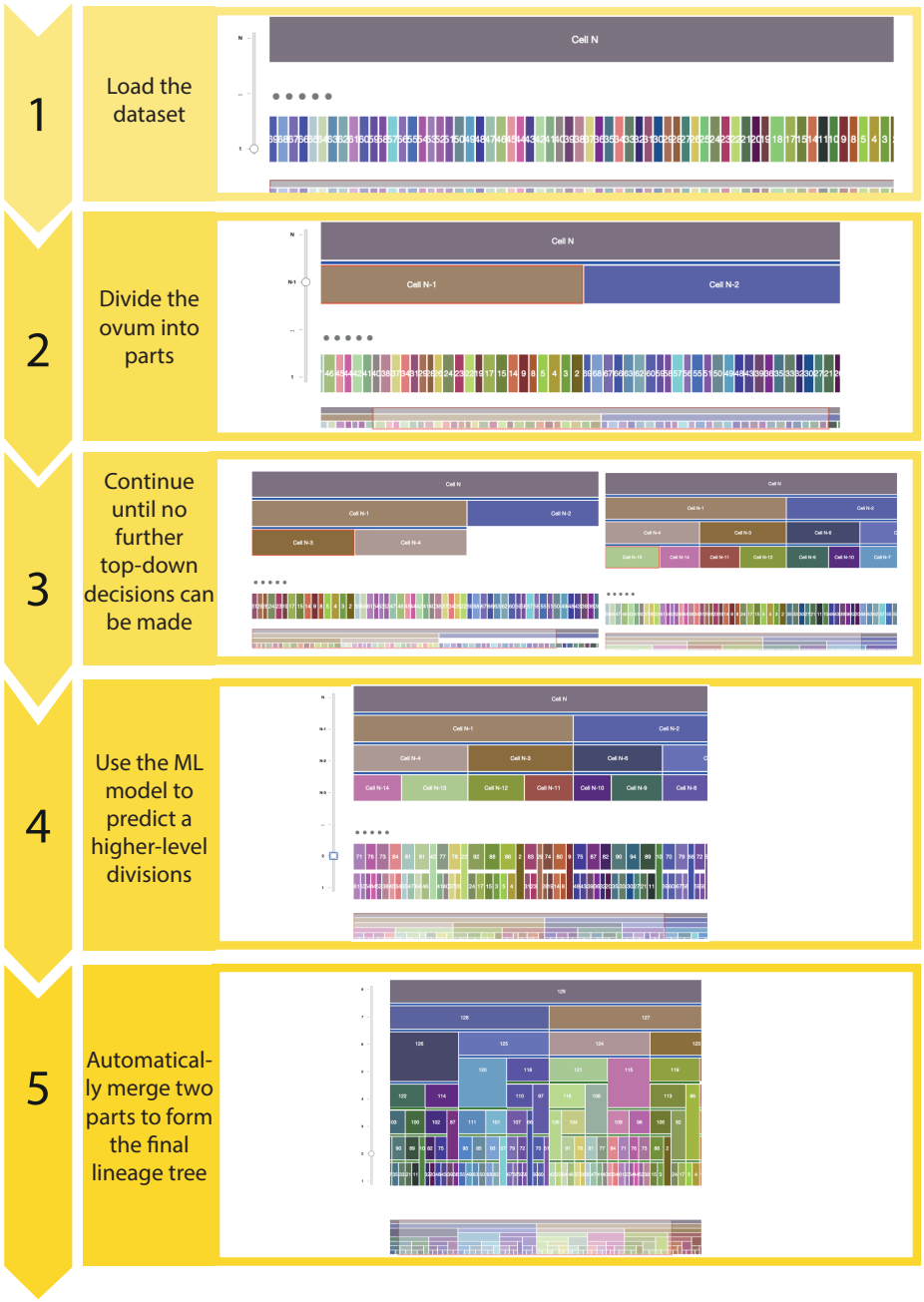

Figure 10: *Illustration of the typical interaction workflow.*

then parse the remaining data and extract all information related to the individual cells such as their centers, volumes, direct neighbors, and surface area. In addition, we prepare another data structure that will later hold the hierarchy information, and that needs to support specific cell orders in each layer based on the ML or manual assignments of sister cells. Moreover, this data structure needs to support the incomplete tree that has information on both ends and grows toward the middle. It gets updated with each cell assignment operation and also needs to handle non-hierarchy cells (supporters).

Finally, we use a third data structure for ML model training and prediction. Because the training records are information about pairs, we calculate, for every potential set of neighbors, the data such as distance and shared surface area, and normalize this information using all possible pairs within the same generation.