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.

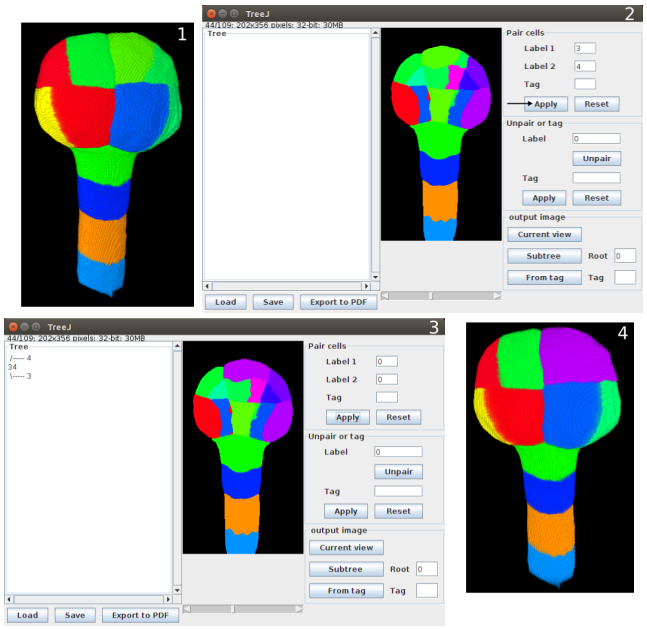

Figure 6: *Screenshots of the interface in TreeJ (a plugin to Fiji ImageJ) that the experts traditionally use to specify the cell lineage. Users primarily interact in a 2D slice of the microscopy data, colored by the cell's assigned ID, to select a pair of cells to merge. The resulting tree is then represented 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.*

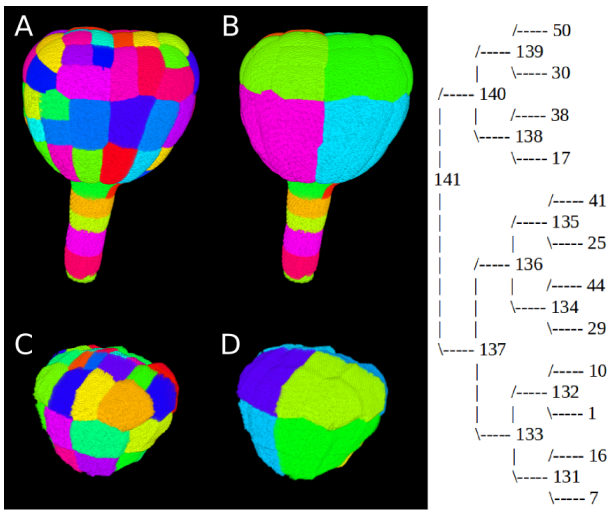

Figure 7: *Example cell lineage result produced with TreeJ that shows several development stages (A–D) of the cell in the 3D view and the corresponding text-based lineage hierarchy. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.*

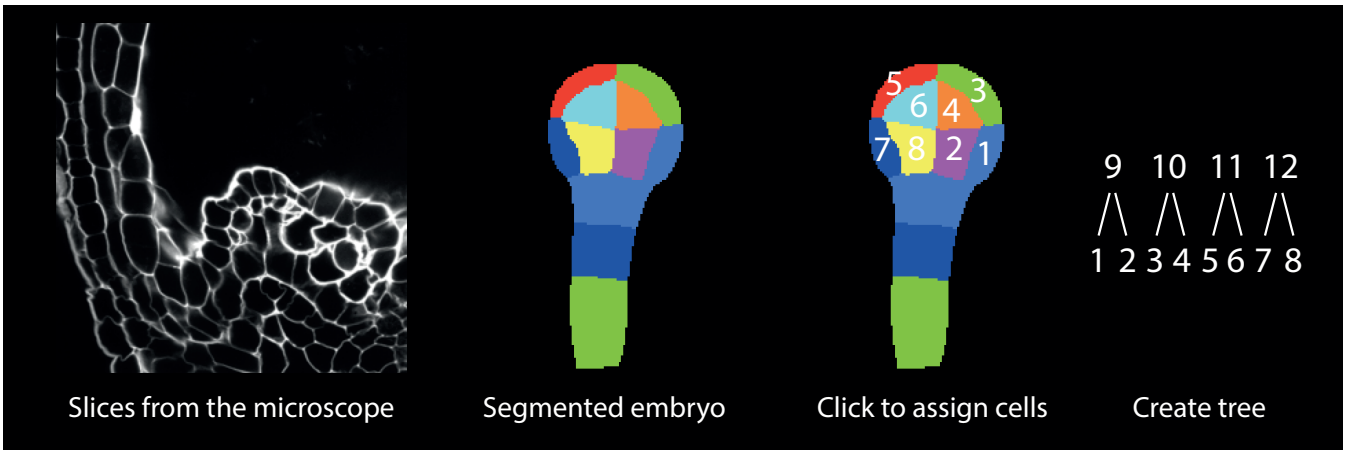

Figure 8: *An illustration of the traditional way of biologists doing cell lineage based on slices. The image insets in this figure are © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.*

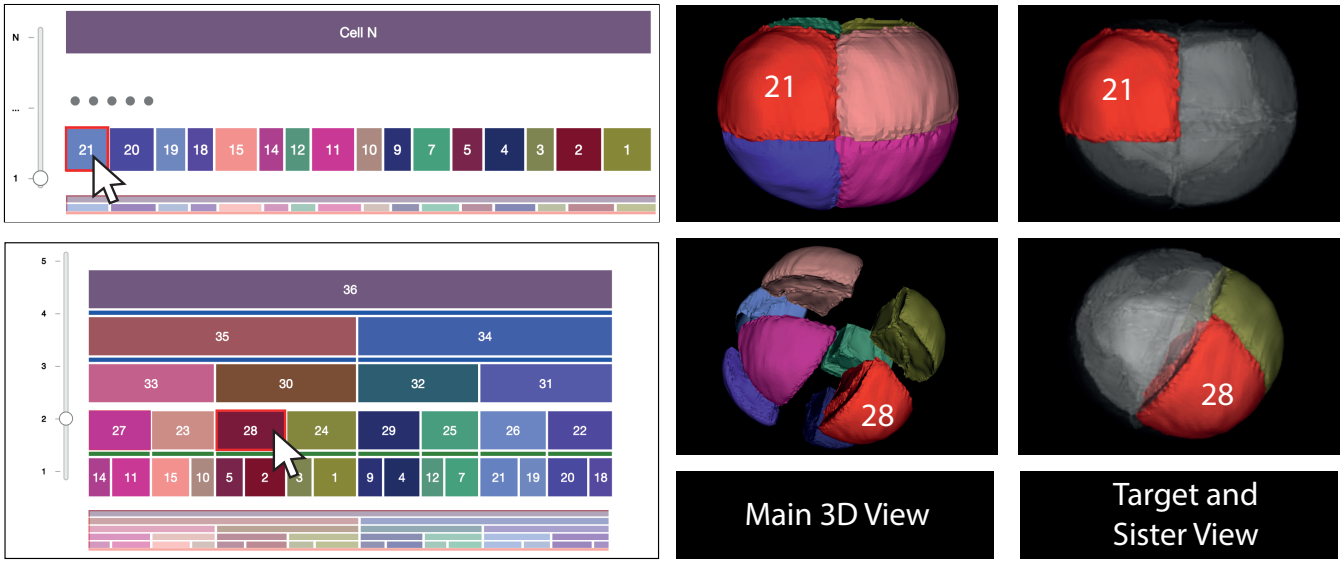

Figure 9: *Illustration of the interaction that links selections in the 2D hierarchy tree to the representations in the 3D view, including the highlighting of the selected cells and potential neighbors.*

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

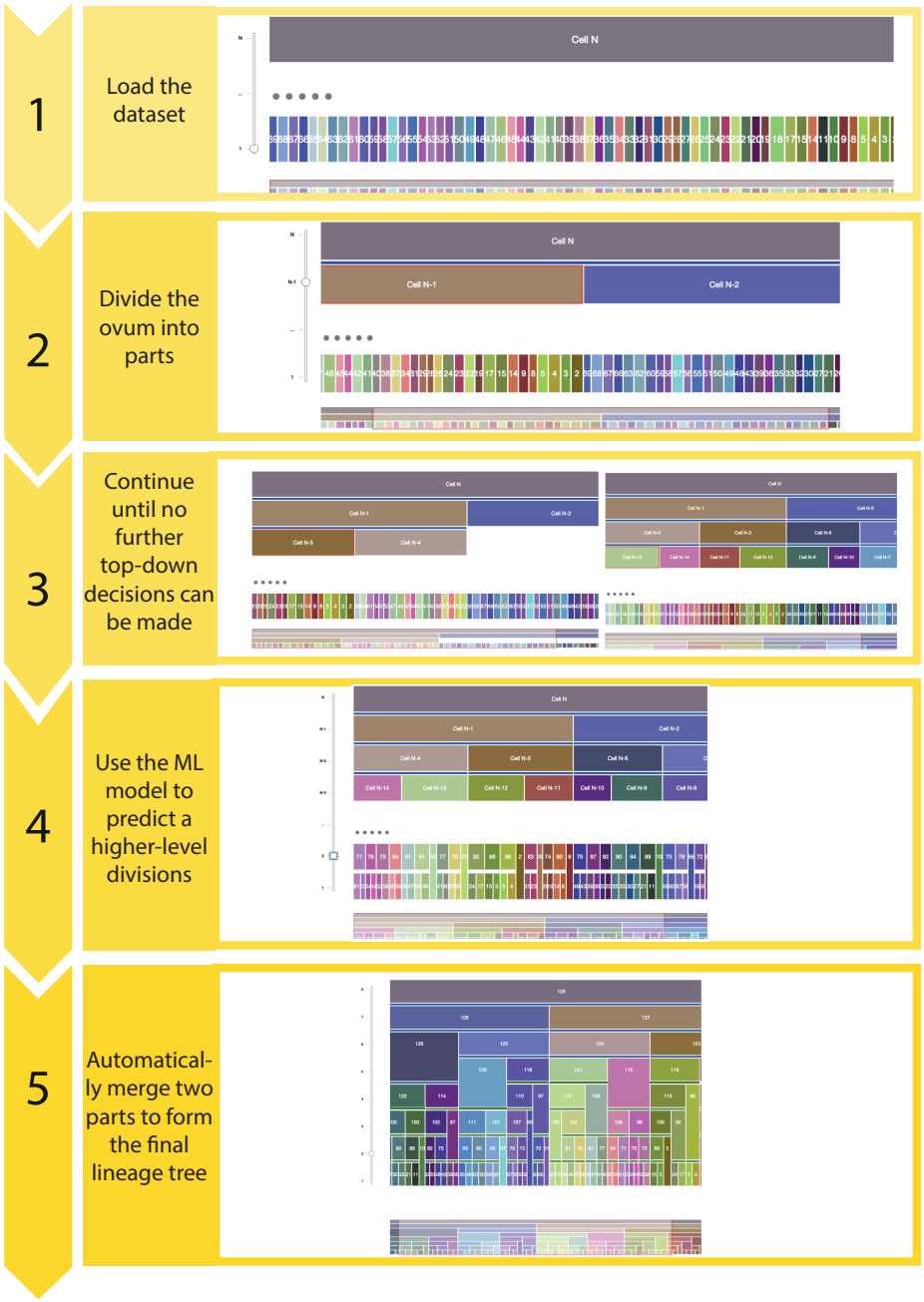

Figure 10: *Illustration of the typical interaction workflow.*

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.