Visualizing and Comparing Machine Learning Predictions to Improve Human-AI Teaming on the Example of Cell Lineage

Jiayi Hong, Ross Maciejewski, Alain Trubuil, and Tobias Isenberg

Abstract—We visualize the predictions of multiple machine learning models to help biologists as they interactively make decisions about *cell lineage*—the development of a (plant) embryo from a single *ovum cell*. Based on a confocal microscopy dataset, traditionally biologists manually constructed the cell lineage, starting from this observation and reasoning backward in time to establish their inheritance. To speed up this tedious process, we make use of machine learning (ML) models trained on a database of manually established cell lineages to assist the biologist in cell assignment. Most biologists, however, are not familiar with ML, nor is it clear to them which model best predicts the embryo's development. We thus have developed a visualization system that is designed to support biologists in exploring and comparing ML models, checking the model predictions, detecting possible ML model mistakes, and deciding on the most likely embryo development. To evaluate our proposed system, we deployed our interface with six biologists in an observational study. Our results show that the visual representations of machine learning are easily understandable, and our tool, LineageD+, could potentially increase biologists' working efficiency and enhance the understanding of embryos.

Index Terms—Visualization, visual analytics, machine learning, comparing ML predictions, human-AI teaming, plant biology, cell lineage.

1 Introduction

In biology, a plant cell (the *parent*) normally divides into two daughters (or *sister*) cells over time, and an embryo grows to eventually comprise hundreds of cells. The sister cell was the other cell that divided from the same parent cell with the target cell. To explore the history of an embryo's development, biologists use a 3D microscopy snapshot and assign sister relationships for every cell in the embryo. They do this reasoning iteratively and backward in time across a series of snapshots to arrive at the previous cell division stage. For each generation, biologists search for the right sister cell of every cell. They continue this process until they finish all the generations and build the hierarchical tree. The datasets used in this process are extremely imbalanced because one cell can only have one correct sister cell, yet the cell usually has a dozen or more neighbors. As such, the manual assignment of the cell lineage for embryos of realistic sizes (several hundreds of cells) is extremely time-consuming and tedious. However, with the help of machine learning (ML), this procedure can be made significantly easier as it is a binary classification problem—two neighboring cells are sisters or not. In our previous work, we developed a tool called LineageD [16] with an ML model to help with cell assignments. Using only a single model, however, gave unsatisfying predictions mainly because of the limited datasets. Also, there is no guarantee that the same model will perform correctly (or incorrectly) for another cell pair at another division stage. For instance, although neural networks have a high accuracy rate, their recall is low. This means that for some specific cell divisions, they cannot provide precise predictions. In contrast, Gaussian naïve Bayes has a relatively high recall. Overall, in 99.69% of all cases at least one model gave correct predictions for our specific training data. The use of different ML classifiers can thus help the experts in such situations because together they can potentially cover most cases. Ideally, the biologist thus trains multiple models and explores which model or groups of models are most reliable for a given assignment.

In the visualization community, researchers have focused on finding and training an "optimal" model to solve a given domain problem [21], [45]. Visualization tools have been developed to illustrate all steps of the machine learning pipeline, including data processing, training, and evaluation (e. g., [33]). However, even highly optimized models with high accuracy still have the potential to provide wrong predictions. In our cell lineage scenario, if a model wrongly predicts the assignments in the first few generations, the predictions for the following generations are almost certainly incorrect as well. Thus, biologists cannot exclusively rely on a completely automatic ML process. Instead, a human-AI teaming approach is preferred, where experts can observe, control, and update the labeling process. However, little work has concentrated on how to enhance this *human-AI Teaming*[1] [12], [25] to assist experts in the decision-making process, rather than focus their efforts on improving a given model's performance. To fill this gap, we visually represent the different ML predictions to assist the biologists, allowing them to better understand the classifiers and enabling them to efficiently derive the correct lineage.

We collaborated with plant biologists and explored how to use Human-AI Teaming to help with the cell lineage problem based on our feedback from LineageD. We collected 93 embryo datasets

- *Jiayi Hong and Ross Maciejewski are with Arizona State University, USA. E-mail: {jhong76 | rmacieje}@asu.edu.*
- *Alain Trubuil is with Université Paris-Saclay, InraE, France. E-mail: alain.trubuil@inrae.fr.*
- *Tobias Isenberg is with Université Paris-Saclay, CNRS, Inria, France. E-mail: tobias.isenberg@inria.fr.*

Manuscript received 8 December 2022; revised 6 July 2023; accepted 1 August 2023. Recommended for acceptance by T. Ropinski. Author version; doi: 10.1109/TVCG.2023.3302308

1. We use the term *human-AI Teaming* in reference to systems whose machine intelligence modules can be controlled as well as potentially overruled by the human users based on their professional experience. For more discussion on this point please see our Section 6.

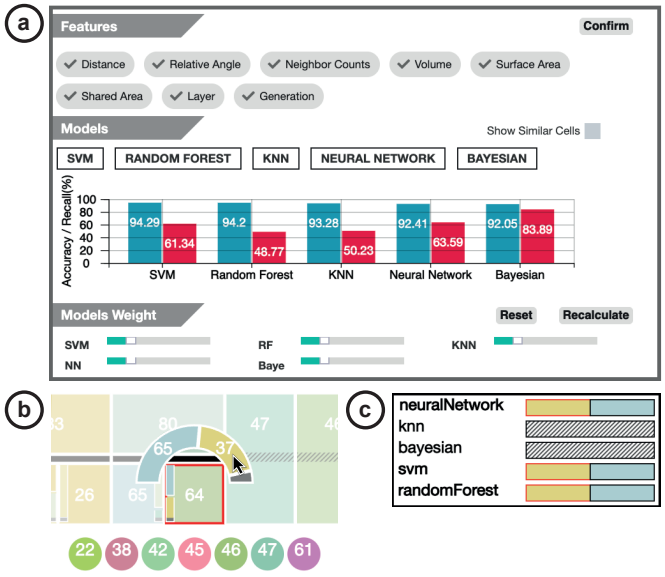


Fig. 1. The visualization design for our multiple machine learning models. (a) The overview of the model performance. Users can customize the features to train models. Also, they can change the model weights via the panel. (b) The detailed prediction of a specific cell (cell 64 as an example in this figure). The cells' ID numbers were pre-assigned in the input data provided by biologists. (c) The model results of the target cell (cell 64) and the interested proposed sister cell (cell 37) being sisters.

with manually specified lineage and extracted 47,132 cell division records with 12 features. We trained five ML models with this data: support-vector-machine (SVM), random forest, k-nearest neighbors (KNN), neural network, and Gaussian naïve Bayes (ordered in decreasing cross-validation accuracy rates) as shown in Figure 1(a). To help the experts in comparing these models, we provide them with a prediction overview of every cell in the hierarchical tree along with a detailed model view (Figure 1(b, c)). We sorted the hierarchical tree by the models' overall confidence and support similarity pattern detection for target cells such that the biologists can easily find possible mistakes. In addition, we visualize each model's accuracy rate and recall via cross-validation and also visualize their model weights. We use user-adjustable model weights as a proxy for the user's trust in each model. Moreover, we allow the biologists to select new features to train new or updated models online. Finally, we conducted an observational study with six biologists, and our results show that participants understand and appreciate the visualization design of multiple machine learning models, and they can check and correct the predictions effectively. Our contributions include: (1) a visualization system to assist biologists in effectively establishing cell lineage, (2) an exploration of how to use visualization techniques within human-AI teaming approaches, applied to the example of cell lineage specification, to better make use of AI approaches in which a single ML model is too limited, for example, due to too little training data or due to model limitations, and (3) an evaluation with domain experts in cell biology on using machine learning and visualization-assisted human-AI teaming for solving their scientific problems.

2 Related Work

We work toward improving the interaction of domain experts with multiple machine learning models rather than training better models. Here we summarize the relevant literature on the visualization of hierarchical information in the field of biology and the visualization of machine learning model output.

2.1 Visualization of Hierarchical Information in Biology

Various visual representations exist for general hierarchies and specifically for cell lineage. General approaches focus on building different kinds of hierarchical representations [37], [38]. Besides showing the necessary hierarchical information, biologists usually need to add additional information to the current tree design, e. g., time [11], relative object sizes [16], etc. In our scenario, we also need to add model predictions for each cell in the tree and visually track the manual adjustments. Previous work dealt with similar requirements, and, in the past, researchers have created various dedicated visual representations for bioinformatics data. Eisen et al. [11], e. g., designed a representation with colored nodes to represent DNA data that relies on a basic cluster dendrogram with additional temporal data, which resulted in a representation akin to a heatmap. Based on this design, the Hierarchical Clustering Explorer [39] added scattergrams to visualize DNA gene samples clustering under different conditions and provides an overview of different clustering results and detailed information with linked views. These examples inspired us to augment a traditional hierarchy view with other representation techniques like bar charts.

Along with 2D representations, researchers have also investigated approaches to visualize biological data using 3D views. Arena3D [29], e. g., presents network data both on a 2D graph and in 3D space. This combination helps users solve the overlapping and intersection problems inherent to 3D views when the entities reach the thousands. Unlike this and other similar work, in our application the detailed predictions for individual cells—usually nodes or points in the diagram—are as important as the overview of the results. In this case, we need to ensure that biologists can easily navigate the tree, see the prediction overview, and track the manual assignments. We visualize cell history in a 2D tree based on icicle plots and add representations directly to the tree.

There is also similar work in which researchers visualized the cell lineage to enhance the biologists' understanding. In Salvador et al.'s [34] CeLaVi tool, for example, the assigned cell lineage is visualized in a hierarchical tree, and the cells' positions are indicated with a circle in a 3D environment. The two views are closely connected, so biologists can target cells from the tree or the 3D view. In addition, CeLaVi allows researchers to highlight a specific gene and visualize the overall gene expression pattern in a heatmap. Though it supports the efficient analysis of the cell lineage, the tool does not support the building or the adjustment of the hierarchical tree, which is imported from a static file. Also, the 3D view only contains the cells' positions, without the cell shapes or shared surfaces. The lineage hierarchy itself, traditionally, is mostly being established manually with tools such as OsiriX [32], TrackMate [46], Fiji ImageJ [35] with the TreeJ plugin, but these tools typically rely on 2D slices.

Because the hierarchy results from numerous cell divisions and the position decides each cell division orientation and angle of mitotic spindle [19], we also investigated automatic algorithms to predict the hierarchy. Researchers have explored diverse machine learning models to predict the division planes [22], [23]. These prediction methods, in the past, have been based on microscopy slices [23]; however, the ML requires specifically prepared training data and the organization of results is difficult. Therefore, we work with the 3D model instead of the image data and use extracted numerical information for the training of our ML models.

2.2 Visualization of ML Model Output

ML prediction processes can be assisted by visualization. Researchers have enhanced, e. g., the interpretability of ML models [5], [33], with problems ranging from clustering [18] to classification [45]. Visual support can assist practitioners in understanding where, why, and how ML models make predictions [33]. Such work needs to show the steps of model generation as well as the actual prediction, including data preparation [43], model training [20], [21], [45], results evaluation [9], and model comparison [18].

In our case, however, the biologists are most interested in the final results of the model predictions—the final cell lineage hierarchy—rather than the ML model generation. We thus concentrate on the visualization of the model output. Prior work in visualizing model output has focused on illustrating the results and comparing predictions. For evaluating ML model results, designers use different representations depending on the prediction type, such as stacked bar charts to represent counts of data points in a cluster [18], scatter plots to show classification results [17], and histograms for visualizing perplexity [30]. Inspired by these representations, we used a variety of representations to show diverse details, such as using bar charts to show the overall predictions and semi-donut charts to represent model predictions for an individual cell.

Another field that can potentially help biologists better engage with machine learning models is interactive machine learning. It uses people's feedback to re-train the ML model to get better results [42] or to pick the optimal model among multiple models [43]—approaches that we do not adopt. EMA proposed by Cashman et al. [7], for instance, is a process that asks people to explore different models and select their preferred ones. Also, Gil et al. [13] proposed human-guided machine learning (HGML) to encourage domain experts to fully use relevant knowledge in getting a high-quality model, even if they have no experience in ML. The work done by Sugawara et al. [44] is also relevant as it uses deep learning to deal with image datasets to annotate, train, and predict cell tracking. These examples provided us with good examples of how to help biologists understand how models perform and enhance their overall experience: we need to visualize the properties and performance of the different models. However, in our work, instead of generating the best model, which is impossible due to the limited manual lineages available as training data and the fact that experts occasionally disagree on what is the correct lineage, we focus on enabling the experts to fully control the results.

We also focus our system design on enhancing trust between the biologists and the machine learning models. Previous work has shown that interaction can be critical in establishing people's trust in machine learning models. Dietvorst et al. [10] showed that people would likely trust models more if the system allows them to disagree with some predictions. Such trust could establish people's beliefs and their willingness to use the system and complete specific tasks [40]. When interacting with multiple models, examining various predictions and deciding which model to trust are also types of interaction to control model predictions. Thus, biologists need to interact with the prediction results, especially when they have no experience in dealing with machine learning models. As previously stated, producing a single ideal model for cell lineage classification is impossible. To overcome this issue, we produce multiple machine-learning models for biologists to compare and decide on. Such comparisons of ML prediction results are different from traditional comparative visualization [27], [47], which is to understand how datasets are similar or different. We visualize and compare predictions from five different models for one specific pair to help the biologists understand the model performance of each model better and make decisions based on overall predictions.

3 System Design

Based on this background, we set out to support the biologists in their work with a system that uses visualization to allow them to interpret the ML predictions for their cell lineage tasks. In this Section, we detail the design process of the LineageD+ tool.

3.1 Design Considerations

We summarized the biologists' feedback towards our previous tool LineageD [16], and discussed with a biologist in monthly meetings. Based on the results, we set a number of design goals for the new system. First, we wanted to take the biologists' **traditional work process** into account and support them in getting fully involved. As we saw in our study of LineageD, biologists are keen on their familiar components and interactions, even though they can be less efficient. We decided to keep the typical workflow and, at the same time, introduce advanced techniques to improve efficiency such as ML. Biologists usually do not have experience in ML, and they rarely care about the details of such models. Instead, they are concerned about how much a model can help them with making the assignments. Even though they do not care about the details behind the models, enhancing their control over the models, such as selecting customized features, could help us to increase their involvement and establish usage confidence. As such, we need to provide intuitive and efficient interactions to detect wrongly-paired cells and correct them if needed.

With the goal of better integration of ML in mind, we then reflected on the feedback from LineageD. The lack of a sufficient amount of training data prevents us from following the ideal approach, which would be to establish an optimal ML model as we did with LineageD. Furthermore, even experts can have difficulties deciding the right sister for a specific cell, and any given manual lineage solution is thus not necessarily unique. While more research into ML support for cell lineage may produce such an optimal model in the future, our focus is on training **multiple models** such that diverse models can cover most pairing cases. Also, biologists can compare their predictions and decide on which model to use in a given assignment. Our visualization and interaction design should help biologists to compare the different ML models and make informed decisions. With diverse models, our follow-up consideration was that we need to **guide** the biologists **on the variety** of ML predictions at a local level for each proposed pairing. Showing them one hierarchical tree per ML model would likely overwhelm them. Instead, it makes more sense to show them a single hierarchy (level) based on the most likely prediction, and then to show them the model disagreement for each cell or cell pairing. This local presentation of differences in the ML predictions should augment the global comparison of the different models.

Besides the ML prediction visualization, we need to visualize the uncertainty. The reason is that the cell lineage is deducted backward in time from a single stage of the plant embryo's development, which makes the process **inherently uncertain**. As just noted, even experts sometimes come up with various hierarchies for the same dataset that differ in details. The manually-assigned embryos we received from the biologists for generating the training datasets range from 2-cell embryos to 256-cell embryos, covering a wide range in the development history. Overall we had

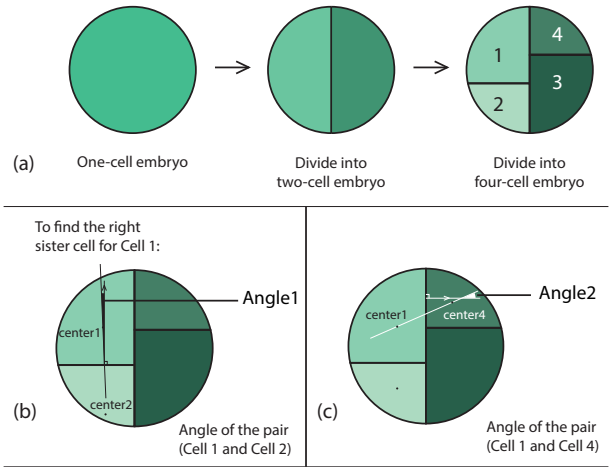


Fig. 2. Illustration of (a) the first two stages of cell division and (b, c) the angles we used in our ML training to identify likely sisters (illustrated in 2D for clarity). Here, *angle2* is bigger than *angle1*, which means that *cell2* is more likely to be a sister of *cell1*, compared to *cell4*.

access to only 93 datasets in total and this limited amount of samples adds to the uncertainty in the data. So the application of ML will inherently also have uncertainty, and it is essential to report this information. For example, one model may predict one pairing, while another model predicts another pairing. Thus, the user may have several options for selecting cell lineage, or even none at all. Our goal is to showcase the models' confidence in their predictions to allow biologists to understand the uncertainty.

3.2 ML Features, Model Training, and Prediction

To determine the features to use for the ML, we asked our collaborator about which properties biologists check when they decide on which cells are sisters. We also referred to established, universal division rules [2], [31]. Finally, we took each pairing in the manually assigned datasets as a record to be able to produce our training data. Based on these considerations, we extracted 8 properties with 12 features in total from the cell pairings as follows:

(1) the *normalized distance* between two neighboring cells, computed from their respective centers (mean vertex positions),
(2) the *angle* between the line that connects both centers and the weighted average normal of the shared surface,
(3) the *number of direct neighbors* of each of the two cells (i. e., two features as the value for the paired cell is independent),
(4) the *ratio of the volumes* between the two paired cells,
(5) the *ratio of the surface area* between the two paired cells,
(6) the *ratio of the area of the shared surface* to the surface of each cell (2 features),
(7) the *layer count* from the surface of the embryo in which each of the two cells are located (2 features), and
(8) the *generation of a cell in the division process* along with the *total cell count* in this generation (2 features).

For the *first property* we normalize all distances between adjacent cells in a generation to the interval [0, 1], 0 being the minimum distance and 1 being the maximum. This property encodes how close two sister cells are as compared with other adjacent pairs in the same generation. With the *second feature* we encode the orientation of the shared surface with respect to the centers of both sister cells, and pairings with low angles are more likely to be sisters than those with higher angles—as confirmed by our collaborator. As we show schematically in Figure 2, we compute the lines that connect both centers of a potential pairing and then compute the angle of this line to the weighted average normal of the shared surface (Figure 2(b)). For the *seventh property* we first classify all surface cells as layer 1 and others as internal cells. We then ignore all surface cells and iterate the algorithm, marking the next layer as layer 2, etc. With this layer marking we capture the two different types of cell divisions, periclinal and anticlinal. In periclinal division, cells divide into two in a row, while cells with anticlinal division divide into two cell columns. Younger embryos usually divide periclinally, while anticlinal division often produces cells with new functionality. We capture this aspect through the layer property, along with the *eighth property*, the cell generation.

TABLE 1
The overall accuracy rate and recall value for the five models using the cross-validation approach.

model	random forest	SVM	KNN	neural network	Bayesian
accuracy rate	94.24	94.23	93.42	93.30	92.07
recall	44.20	60.96	52.37	67.44	84.22

We used these 8 properties to pre-train our ML models using all 12 features. In addition, we allow the biologists to customize the feature selection and to train new models with only a subset of features if they desire. In LineageD+ we support 5 different machine learning techniques: support-vector-machines (SVM), random forest [4], k-nearest neighbors (KNN; we used $k = 5$), artificial neural network, and Gaussian naïve Bayes. Based on the datasets, we picked supervised machine learning classifiers which are applicable to train and predict online. In this way, users can directly re-train the model locally without setting up the corresponding environment. Models from other families, such as XGBoost [8], could also potentially provide different results. We excluded these, however, because either they require external package installation (e. g., `sdk`), making the website less accessible, or they do not support online real-time training.

To train our models, we treat the cell lineage prediction problem as a classification problem. This means that the ML should predict, for a given potential pairing, whether these neighboring cells are, in fact, sisters or not. We use our manually specified lineage datasets and extract, for each generation in this data, all potential cell pairings of neighboring cells in a given generation and classify them either as a correct sister pairing or as an incorrect pairing. This way, we capture both positive and negative targets from our input data. In total, we had 47,132 records with 43,392 negative targets and 3,740 positive ones. We had a lot more negative targets than positive targets as each cell has many direct neighbors but only a single correct sister. To solve this issue of our highly imbalanced training data, we use randomly over-sampling and under-sampling [24]. Finally, to analyze the stability of the models, we used a k-fold cross-validation approach ($k = 50$) and derived each model's accuracy rate and recall value, as we show in Table 1.

For our actual prediction of a new hierarchy level,[2] we first use all five models independently to predict all possible sister cells for any given cell. For each positive prediction, we then weight the prediction by the corresponding *model accuracy*. We also weight it with a customizable *model weight* to allow biologists to control the influence of each model (1 by default). For each proposed pair, we

2. We only predict one level at a time as we rely on user feedback for correction, and a wrong prediction invalidates any further hierarchy levels [16].

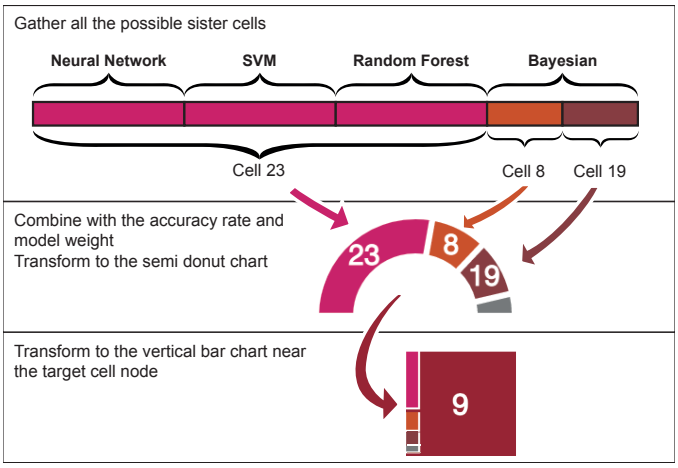


Fig. 3. The connection of visualization design in the different information levels. Here, we use Cell 9 as an example of the target cell. At the top, we place all the predicted sister cells from each model horizontally. Then, for each proposed sister cell, we calculate the proportion combined with the model weights and accuracy rates. We transform it into a semi-donut chart, with the grey area representing the uncertainty, to show how likely a proposed sister can be the right sister cell for the target cell in the middle. We also transform it into a vertical bar chart to give biologists a quick overview—without the need to unfold a detailed view—of the general model confidence of a supposed pair as shown at the bottom.

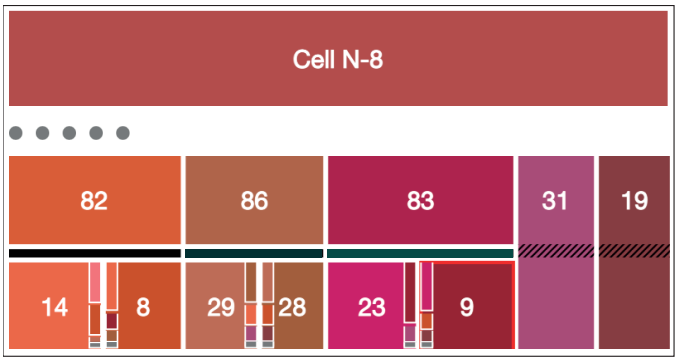


Fig. 4. The visual representations biologists see after the models predicted a level. An embryo can be divided into sub-embryos, each of which includes all the children cells that were divided from this sub-embryo. Experts can distinguish and decide on these sub-embryos from their experience and knowledge. Here we take the sub-embryo N-8 as an example. All the cells (14, 8, 29, etc.) below were divided from this sub-embryo. For this sub-embryo, we predicted the new level with the sub-embryo constraints and sorted the predictions based on the overall confidence of models from left to right. As the image shows, the left-most pair has multiple possible sisters for both affected cells. The position of a cell in the sub-embryo and the color of the bar indicate the overall certainty of prediction. The parent cell 82 for two children cells 14 and 8, for example, is the least certain prediction as it is placed left-most in the sub-embryo N-8 (i. e., needs to be processed first), and the bar below the parent is almost black, whereas higher certainties would be indicated with relatively right positions and greener colors (in the example above, the certainty is still fairly low so the amount of green is also fairly low; Figure 11 shows an example with higher certainty and greener marks). For cell 83, the ML models are more confident about assigning cell 23 and cell 9 as sisters. Also, from the figure, we observe that the ML models predict cell 19 and cell 31 to have no sister cell in the current level.

thus get the average prediction across the five models. We then sort all pairs by their probabilities and filter those with a probability value lower than 0.4. This means that we only consider those pairs on which at least two models agree. There are cases, however, when cell A has the most likely sister cell B, while for cell B the most likely sister cell is another cell C. In these situations we sum up the probabilities of these two pairs separately for each child cell and then pick the pair with a higher probability. In this way every cell can be marked with a comparatively most likely sister.

3.3 Visualization Design

Based on these design considerations and our ML setup, we created LineageD+ with a particular emphasis on the visualization of ML model parameters, on features to support mistake detection, and on interactive decision-making to improve the biologists' workflow.

3.3.1 Visualization of ML prediction data

The multi-model prediction process described in Section 3.2 allows us to produce a single (partial, bottom-up) hierarchy similar to our previous one-model approach [16]—which we show in an icicle plot. The color of each cell in the 3D view and the hierarchical tree is identical. We encode the cells' relative spatial positions with a color map based on the work of Ovsjanikov et al. [26], whose color encoding ensures that close 3D locations have perceptually similar colors, while distant objects are perceptually far in color. Our multi-model prediction, however, provides the users with more information about each pairing and the overall model parameters, so that we adjust the visuals accordingly.

We collected the model predictions from the five models, summarized them based on cell names, and connected different views for biologists to check (Figure 3). We introduce the details from bottom to top, following the biologists' interaction order. Figure 4 represents the view after the prediction. In this view, we sort the bar charts descending from top to bottom so viewers can focus more on the most likely predictions. In Figure 4, the vertically stacked bar chart of cell 9 has three parts and an additional gray area. This representation means that, for cell 9, the ML models proposed three potential sisters. The colors of the bar chart elements match the colors of the target cells, so we can see that the majority of the predictions go to cell 23, which actually was chosen as the most likely sister in the pairing. The gray area at the bottom of the bar chart represents the accumulated uncertainty of all of the ML models based on the accuracy rates and model weights. For cell 23, the ML models predicted three possible sisters, and the most possible sister cell is cell 9, thus confirming the match. We place the vertical bar on the side near the proposed sister (i. e., on the "inside" of each paring) to allow biologists to directly compare the individual predictions of any given pair. For those cells without predicted sisters, we add a diagonal line texture upon the nodes to indicate that the ML models consider them not to have divided at this stage (or are not confident enough to make a pairing).

While this overview can provide the biologists with a general sense of the model predictions, the possible alternative matches are not obvious. Also, the ML models occasionally do not predict the correct sister, as indicated by the gray mark based on our previously determined model accuracy rate. Thus, the biologists need to be able to see all adjacent cells and investigate them individually. For this purpose, we designed a half-donut pie chart (Figure 5), inspired by previous work on necklace maps [41], to show all predicted cell names with the proportions indicating the probabilities. In addition, we list all remaining direct neighbors of the selected cell (i. e., those not included in any prediction) in a line below the cell for the biologists' reference. What this representation still does not show is the individual model predictions, i.e., which sisters were predicted by which model. We thus created a pop-up view to show

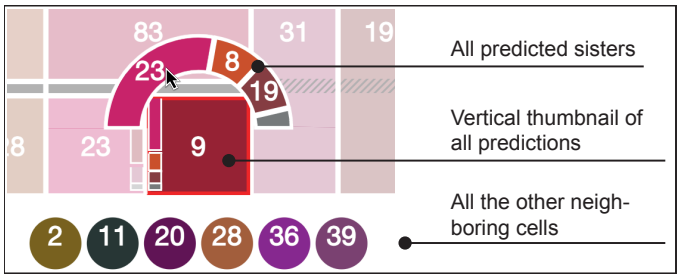


Fig. 5. The details of model predictions for one cell. Here we take cell 9 as the target cell for example. We make all the cells except this target cell with its proposed sister cells transparent. From the colors, we can also tell the relative distances between two cells. The percentage of each model depends on its accuracy rate and the customized model weight. The colors of cells in the semi-donut chart correspond to the cell colors as derived from their 3D positions. A small proportion of gray space indicates the uncertainty of model prediction for the cell 9.

Fig. 6. We show the overall details of predictions from all models when the user double-clicks a child cell in the icicle plot. Taking again the example of cell 9, this view shows us that Neural Network, SVM, and Random Forest predicted cell 23 as the only possible sister, while Bayesian proposed two cells, 8 and 19, to potentially be the sister. KNN, however, does not propose any potential sister for cell 9. This figure correlates with the top level of Figure 3.

the overall predictions from five models (Figure 6). To display which models proposed a specific sister cell, we added another pop-up view to the current interface (Figure 7), which we show when the biologists hover over one of the bars in the half-donut pie chart. In the example in Figure 7 which shows the popup upon hovering on the cell 9 curved bar in Figure 5, three models predict cell 23 as the sister cell of 9. For models which do not predict these two cells as sisters, we used diagonal stripes as the texture.

3.3.2 Support for detecting possible errors

It is also important to allow the biologists to quickly target potential errors in the predictions. We adopted two approaches for this purpose: (a) we sort the newly predicted cell pairs for each top-level sub-embryo based on the overall model certainty and (b) we use color highlighting in the 3D view to indicate cells that are similar to a given target cell. For (a), we show the cells with the least certainty on the left, with increasing certainty toward the right. We chose this assignment to first show the biologists the most likely mistakes that they need to address, and, as they move from left to right, they can stop once they feel that the remainder of the assignments is reliable. Of course, this can only be sorted within each of the top-level sub-embryos to not break this previously (manually) produced top-to-bottom assignment. For (b), we want to be able to highlight wrongly assigned cell pairs by assuming that if one pair is wrong then another proposed pair is likely also wrong and would share similar properties to the first wrong pair. This situation can be captured by the Tanimoto similarity [1], which we calculate for all possible pairs of mother cells of a given level.

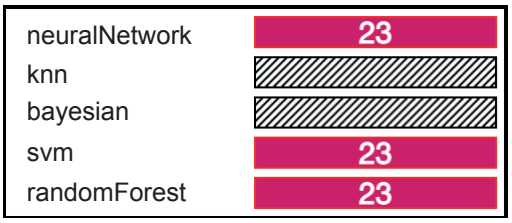


Fig. 7. When hovering over a specific proposed sister cell in the semi-donut chart (as we did in Figure 5, where the mouse hovers over the proposed sister 23), we show the detailed view on the chosen pairing as opposed to showing all predictions as before. Different from Figure 6, here we only show which models actually predicted these two cells as sisters. In this example, three models (Neural Network, SVM, and Random Forest) proposed cell 23 to be the sister of the target cell 9.

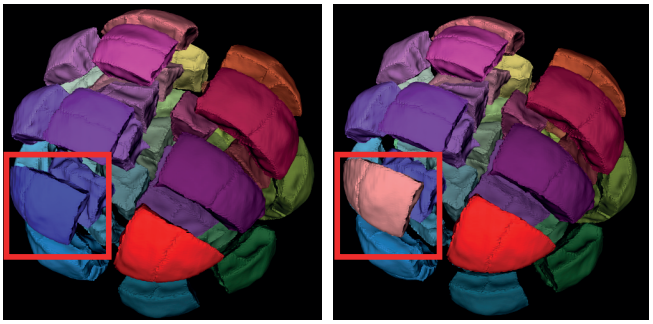


Fig. 8. This view shows the similar pairs with the target cell. The left image indicates the original embryo, and the right image demonstrates the color change of the similar cell in the detection mode. In these figures, the target cell is highlighted in red, and the similar cell is in pink. In this figure we highlighted the similar cell with a red box.

Then, as one target (mother) cell is highlighted (e. g., in the view in Figure 4), in the 3D view, we show those other mother cells similar to the target if its Tanimoto similarity is larger than 25% by encoding with a magenta color (Figure 8). The biologists can then review this limited range of possibly wrongly predicted pairs.

3.4 Further Interaction Design and Decision Making

We based our new interaction design on our previous tool LineageD (see Figure 9) and added ML model interactions to help the

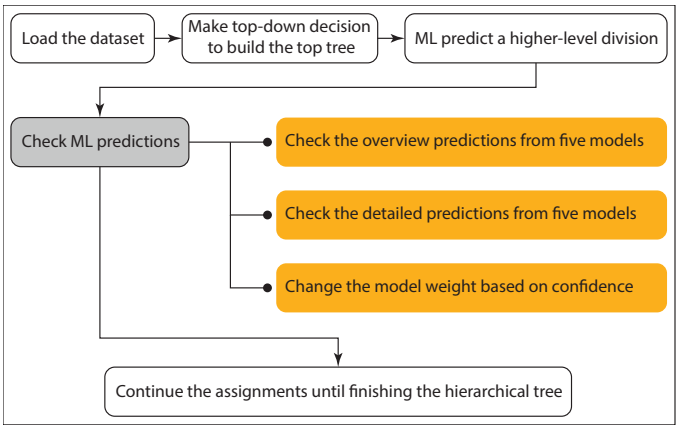


Fig. 9. The workflow of using LineageD+ to do cell lineage. The steps with white backgrounds are based on our previous work [16], and the steps in orange represents the new stages we added in LineageD+.

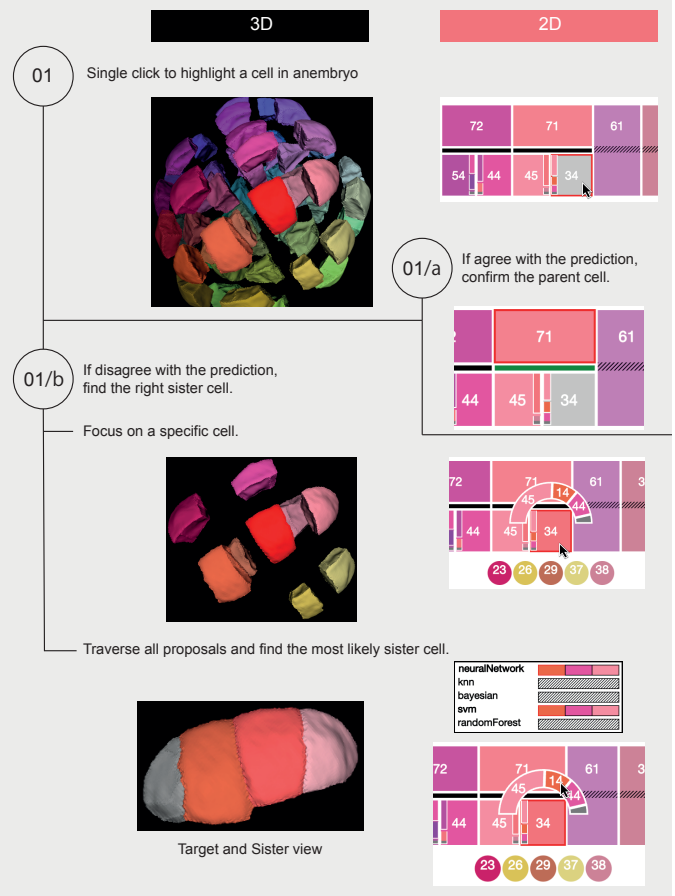


Fig. 10. The interaction process of checking the model predictions. Biologists can target one cell from either the 3D view or the 2D hierarchical tree. If they are happy with the prediction, they can confirm the pair, and the bar then turns green as in 01/a. If they disagree with the prediction, they can double-click to see the details of the prediction and adjust.

biologists to understand the different models' decisions and to make potential adjustments (as illustrated in Figure 10). Before making a new prediction, we now allow biologists to specify custom features based on their background and then to train the five different ML models with these. Once satisfied with the trained models, they can load their dataset and use LineageD's lasso selection which enables people to draw polygons to divide an embryo into subparts in a top-down fashion. In this way, they can build the top tree to constrain the ML later on and to create better predictions. Then, biologists can start to use the models to make predictions. In addition to the visualization of the different predictions by pair that we discussed in Section 3.3, we also provide the possibility to hover over a given ML model representation to highlight all those pairings that were predicted by this model (see Figure 11). To meet the requirements of some biologists who are used to doing the lineage assignment by sub-embryos (the top subtrees), we also allow them to double-click a specific top-level sub-embryo to only display the cells within this sub-embryo, without all the other cells in the 3D view and with the selected sub-embryo highlighted in the hierarchical tree (Figure 12). Finally, for each ML model, we also allow the biologists to change the weight as desired. Then, we adjust the corresponding visualizations and future predictions accordingly. We also enabled biologists to trace their progress using the horizontal bars between parent cells and their children cells.

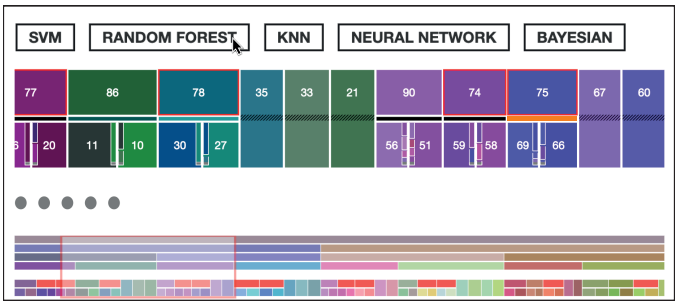


Fig. 11. The illustration of the hovering effect. In this figure, users hover over the model Random Forest. All the pairs predicted by this model would be highlighted with red strokes in the hierarchical tree. Also, the corresponding cell in the thumbnail would turn red too. The horizontal bars indicate the checking status of each pair. Pairs with orange bars, like cell 75, represent that the cells were corrected by biologists. Green bars, like the bar of cell 78, indicate that biologists agreed with the ML predictions. All pairs with other colors are not checked by biologists.

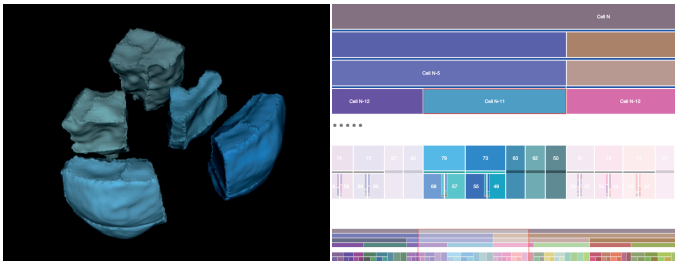


Fig. 12. The illustration of the function to enable users to target one sub-embryo. Here a user focuses on Cell N-11. Then the 3D view only displays the cells within this part, and in the hierarchical tree, we make the other cells transparent. In this way, biologists can easily focus on a sub-embryo to check predicted assignments.

The bars of confirmed pairs are marked green, while we mark the corrected ones in orange as shown in Figure 11.

4 Implementation Details

We now detail our implementation, including datasets, models, and technical realization of the visual representations in our tool.

4.1 Datasets

We received the original embryo datasets in `SURF` format from our biology collaborator, who had processed the data with the biological tool Avizo. This format includes the names of the cells in the embryo, the surface mesh that represents each cell, and the corresponding vertex locations. The cell names consist of a string and a unique number (e. g., `Cell001` or `Materials002`). We extract the unique ID in these names and use them as labels for the cells in LineageD+. We number any newly created parent cell by continuing this count, as suggested by the biologists. For any previously (manually) assigned embryo dataset, we also received another text file that records this information. From this file we extract the corresponding lineage hierarchy for the given embryo.

4.2 Models

We implemented the neural network model using `TensorFlow.js` (v. 3.8.0) to train and predict a *sequential model*, while for the other four models in our tool we used the library `ml.js`. Specifically, we use the packages `libsvm-js` (v. 0.2.1), `ml-knn` (v. 3.0.0),

`ml-naivebayes` (v. 4.0.0), and `ml-random-forest` (v. 2.1.0). We treat the extracted features of each pair as the input for all models. The output of these models is 1 or 0 with the default threshold, indicating which group this pair belongs to: sisters or non-sisters. The neural network model consists of two hidden layers with *ReLU* activation functions and one final output layer with *sigmoid*. We also used the `ml.js` library to get the accuracy with a k-fold (k=50) cross-validation. To detect similar pairs (i. e., to help the biologists to efficiently find error-prone pairs), we used *Tanimoto* similarity and distance methods with this library. For the default datasets, we pre-trained and saved the ML models for quick accessibility. We also support online training with feature sets customized by the biologists, yet a re-training with a custom feature selection requires computation in the order of hours to days.

4.3 Visualizations

We used `JavaScript` together with `Node.js` (v. 14.15.0) and `Express.js` (v. 4.17.1) to build the interface structure including a static back-end server. We implemented the charts that present the model properties and the hierarchical tree with `D3` [3]. For visualizing the embryos in 3D, we used `vtk.js` (v. 19.0.4) [36].

5 Evaluation Study with Experts

To evaluate whether our system can help biologists with better assigning cells using multiple machine learning algorithms, we conducted a case study evaluation with six biologists with specific expertise in cell lineage. We chose qualitative experiments because they can provide a more holistic view of all the factors that would influence visualization and people's useage [6], [28]. We extended our existing ethics application for our work on cell lineage (AVIS № 2021-46) to get permission for this human-subjects evaluation from our institution's (Inria) ethical review board. We also pre-registered the experiment online (`osf.io/2f6uc`).

5.1 Datasets

To allow our participants to get familiar with the LineageD+ interface, we used a 16-cell embryo in the training session. For the actual case study, we used a larger dataset with 64 cells. As we had found in our previous work, biologists build the hierarchical tree using both a bottom-up and a top-down approach. Since the purpose of this study is to evaluate the visualizations of machine learning algorithms (i. e., in the bottom-up approach), we pre-set the top part of the hierarchical tree in advance so that biologists could focus on interactions with the bottom-up predictions.

5.2 Participants

We recruited 6 biologists (3 females and 3 males), aged 32–61 years (mean: 49.5 years), via social networking and mailing lists. All of them have been conducting research in plant cell lineage or a related field for 3 to 20 years (mean: 11 years, sd: 6.957 years, median: 12.5 years). We anonymized their personal data with a participant number (P1–P6). Their specific research focus included bio-image analysis and modernization (P1), plant gene expression (P2), bio-mathematics (P3), cytology image analysis (P4), cell division (P5), and cytogenetics (P6). Three of the participants (P1, P2, and P3) had created cell lineage datasets (daily, once a week, and several times a year), while the others work on general cell lineage problems as opposed to establishing the lineage themselves. P1 conducted the experiment remotely via videoconferencing, while all others participated in person. The in-person attendees used their preferred working PCs or laptops (some with separate large screens) in a meeting room. P1, P2, and P3 had worked with our previous tool LineageD before, while the others had not. With respect to experience in machine learning, P1, P2, and P6 do not use ML in their professional work, while P4 and P5 have some experience in using deep learning for segmentation, and P3 is familiar with convolutional neural networks (CNNs) for image processing.

5.3 Procedure

The study consisted of three parts: a training session, an observational study, and a post-study interview. Before the study began, we distributed a consent form and a background and demographic data collection form for participants to complete. We also asked them four questions about their previous experience in cell lineage and machine learning. For the offline studies, we let participants read and sign an image and voice recording consent form to allow us to take pictures and/or audio recordings. We started the study only after receiving formal consent. Our participants were not paid but received free access to our online tool as compensation.

In the training session, we gave a brief introduction about the purpose and the process of the study. For the remote participant, we asked them to open the website on Chrome using the credentials we provided and to follow us during this training. The training session for this online participant took longer because the remote communication added difficulty for the expert to understand some of the functions of the tool. After the initial introduction was complete, we showed them the procedure of doing the cell lineage for the 16-cell embryo. They could interrupt us for questions at any time. The overall training part took about 15 minutes for the on-site sessions, and about 20 minutes for the remote session.

After the participants felt that they were familiar with the interface, we introduced them to the task of assigning the 64-cell embryo. For the remote participant, we asked them to share the screen with us and talk about what they were doing and thinking during the assignment process. We also recorded the screen and audio with the participants' permission for later analysis. For the in-person sessions, we observed the participants' actions and took notes on how they operated the system. Once they finished the assignments, we conducted a post-study interview using a pre-designed question list (see Appendix A), asking them about their experience with the interface. We also asked them to fill out a System Usability Scale (SUS) [14] to assess their experience.

5.4 Study Results

The study process took approximately 60–120 minutes, depending on how fast each expert could get familiar with the tool (ranging from 10 to 15 minutes). The duration was also subject to their available time slots. All participants finished the assignments of at least one generation, and could quickly get familiar with the system under our guidance—especially in the in-person cases. Five participants identified key difficulties in assigning cell pairs for the 64-cell embryo, and reported uncertainty about their decisions. They noted that there were cases where one cell had two possible sisters. In this case, they would trust the machine learning first and come back to correct the assignments later if necessary. The other participant had no problem with the assignments because they knew the embryo quite well.

Workflow. We observed how participants operated the system during the observational portion of the study. Four participants

did the assignments using similar steps as we observed from our previous study with LineageD, which means that they did not refer to the detailed predictions of each cell during the assignments and instead made decisions on their own. Detailed predictions refer to information such as which model provided a specific prediction and the prediction distribution of a specific cell (as shown in Figure 6 and Figure 5). For them, they already had the assignment ideas when targeting a given cell. When the machine learning algorithm gave incorrect lineage assignments, participants corrected them without checking other proposals from the ML. Thus, in short, the experts compared the ML proposal with their own ideas to make the corrections. Another participant only checked the details when she was not sure about a sister. The final participant, in contrast, checked all the potential sister cells proposed by machine learning and then made a decision. When checking the current pair, whether it was right or wrong, she would always traverse the proposed sister cells to make sure the other options were wrong. Unexpectedly, the biologists targeted the children cells instead of the parent cell to find the wrong pairs. This led to their heavy reliance on the target and sister view of the tool.

In addition, all experts assigned the lineage based on the pre-set top-down sub-embryos. They would finish the assignments of all cells in one sub-embryo and then move on to the next one. Four of them used our sub-embryo view (Figure 12) to only focus on a specific sub-embryo, while the other two went through the cells in the tree order but without engaging the sub-embryo view.

Visual Representations. All participants appreciated the visual information design for the hierarchy and machine learning. P1 and P5 noted that they fully understand the interface. P3 felt that all elements are useful and especially favored the tree visualization design because it links with the 3D view and is easy to understand. P2, P3, and P6 stated that it is really important to have the 3D view instead of the traditional 2D slices so that biologists can see the embryo to better understand it. P4 also suggested exporting the view in 3D so that she can show her assignments to others. P6 expressed her appreciation of the tool development because, for older biologists like her, the tool is easy to use and "cannot be better for her." The only concern she had is that the red highlight coloring can be unfriendly for people with color impairments.

P3 and P4 also provided feedback for future work, noting that they missed a view of only the target cell that would have allowed them to carefully check the shapes of individual cells and their children. P5 was not familiar with the 3D interaction techniques and needed some time to get used to them. Another potential improvement proposed by P3 was to distinguish the level in the tree from the division generation of a given cell. In every round, the ML models predict one level, but this level does not necessarily correspond to the true generation because, when the data is being captured, the embryo can be in the process of division. In this case, some cells in the embryo are in a different generation than the others, and the prediction currently does not consider this situation.

Machine Learning Design. Before starting our observational study, we asked all participants about their expectations with respect to machine learning. P1 and P3 envisioned the possibilities of two cells being sisters so that they can better make decisions. P2 hoped that the ML models can do all the work automatically and only leave the checking job for him. P2 and P5 also hoped the models would be applicable to other datasets (i. e., other species) and tools. P6 assumed that ML can help predict both the past and the future of embryos. It means that, hopefully, models can predict not only the hierarchical tree, but also the future fate of the cells.

After our study, we asked the participants what they think of the machine learning support in LineageD+. Though four of them reported that they did not have enough time to fully experience the ML in terms of using ML for other more datasets, all participants appreciated the prediction results. The three biologists who had used LineageD before said that the performance and experience of LineageD+ are much better than the previous version. All experts also thought the visualizations of machine learning are readable and easy to understand. P6 reported that interacting with machine learning made her feel like she was discussing with the computer in making assignment decisions. She started to look at one proposed pair and targeted one of the children cell. For her, the machine learning was proposing other options in the semi-donut chart and she would almost "talk" to the model about whether it is wrong or if it makes sense. She felt she did not have to think much but just traversed the proposals from machine learning.

As for improving the machine learning itself and how it can be deployed, though P1 and P2 did not refer to the detailed predictions of each cell from ML, they were curious about how models worked and why models gave such predictions. P2 and P6 were also interested in knowing why models can come up with a specific wrong pair. P3 argued that he cannot decide the model weights in the very beginning. It required much time to interact with the system so that users can choose and decide the preferred model weights. He and P2 would love to see the machine learning do the weight adjusting job for them. P4 was concerned that she would potentially be influenced by the proposed predictions from machine learning. Though there are vague situations where the assignment for one cell has multiple solutions, the ML would pick one solution automatically for the biologists, and they would need to manually try out the other possibilities. P2, P3, and P4 would love to have the models updated based on their corrections and then predict future pairs using the updated model. Another wish of P2 and P6 was that the ML models should predict the two directions, top-down and bottom-up, rather than only bottom-up. Also, P2 thought it would be better to report the feature weights used by the models and to enable users to change these weights as well. Since P1 and P3 were used to assigning embryos section by section, they would have loved to have the machine learning predict all assignments within a sub-embryo, instead of only a single level.

Interaction Design. We asked participants about their interaction experience both with the general interface and with the ML models. For the general system, P1, P2, and P3 favored the interaction connection between the 3D view and the 2D view. P1 said that in traditional tools, such as the TreeJ plugin to Fiji ImageJ, it is hard to find the exact cell with the 2D slices, but LineageD+ perfectly solved this problem for her. P2 and P4 were impressed by the sub-embryo focusing function. It enabled them to clearly make decisions within the range constraints and thus reduced potential errors. In addition, P2 appreciated that he could adjust the ML model weights and that the detailed information about the ML predictions was not presented all at once but instead in layers that could interactively be revealed. Also, P3 and P4 appreciated the explosion and peeling function to solve the occlusion problem. P4 and P6 liked the interactions for comparing different potential sisters in the target and sister view. Interestingly, P6 appreciated the use of mouse clicks for making re-assignments, yet both P1 and P3 reported that they were confused by the single- and double-click actions in the tool. Though they could understand the use of these clicks after we explained the differences to them (single click to target cells and double click to assign cells), P6 would have liked a

more detailed manual for people without technical support. P1, as the only remote participant, expressed her concern as to whether or not she could finish the assignments without the experimenter's help. In addition, P4 and P5 felt it was inconvenient to have the buttons below the hierarchical tree, and P4 would have preferred the tool to have a right-click menu for the tree interaction.

Potential Influence. In the post-study interview, we asked participants how fast their construction can be and how confident they felt about the results after sufficient training. Overall, all participants thought that LineageD+ can help save time, even though they would check every proposed pair. Also, based on P1's feedback, the speed of using LineageD+ partially depends on how familiar a biologist is with the analyzed embryo. When the biologist had never done any assignments for the specific embryo before, the machine learning predictions can be a large help for the thinking and decision-making process. Yet, the effect may not be so obvious for familiar embryos because biologists need to go through every cell anyway. P4 and P6 would love to try out other datasets to confirm the assumed improvement of assignment efficiency. Reporting on their confidence in working with LineageD+, P4, P5, and P6 thought they would feel very confident because they can easily understand and use the tool to check every step. P3 thought that LineageD+ offers the same confidence as if he would manually establish the lineage. P5 assumed that his confidence in the results can reach about 90%. The other participants would feel equally confident with the results from the traditional tool because they would basically do the same assignments.

In addition, we asked the biologists whether they thought the tool would change the traditional approach they used in the assignment. All of them believed that LineageD+ can possibly change the strategies they used. P1 and P2 assigned the cell lineage sub-embryo by sub-embryo, and the ML predictions build the tree level by level. They were used to the sub-embryo-based assignment because they wanted to move the embryo slices as little as possible in their traditional tools, but in LineageD+ biologists knew where they are thanks to the 3D visualizations and they thus may switch to a level-by-level assignment with some training. P4 specifically emphasized this notion because LineageD+ provided clear visualizations of individual cells as well as how two sister cells can be combined. P6 noted that tools for biologists should be free and easily accessible and that LineageD+ meets this requirement.

We also calculated the System Usability Scale rating as 77.67/100 (sd: 15.77) on average, which is higher than the average SUS score (68) [14]. Yet it can still be improved via all the aspects mentioned by the participants as we reported. Among all participants, only one gave a score lower than the average SUS score, who was the remote attendant—a possible reason being that the networking and communication affected her experience. Compared with LineageD's SUS score of 68, the improved score could be due to LineageD+'s improved model performance and interaction experience. Another possible reason is that five participants conducted the study in person (compared to LineageD) so that, if they encountered problems, they could get timely help.

6 Discussion

Based on the study results and biologists' feedback, we summarize the following points as takeaways from our work.

6.1 Various Interaction Types with ML Visualizations

First, we observed that experts use machine learning and the visualization of its details depending on their level of expertise, their familiarity with a given dataset or task, their background, and the perceived performance of the ML models. For example, when a biologist was familiar with the species of the embryo, they would concentrate on checking the ML's assignments and largely only validate the predictions—provided that the model would perform well. Only if the model made apparent mistakes they would explore alternative assignments and quickly settle on an alternative cell pair based on their experience. Biologists with less experience with the embryo's species, in contrast, would rely more heavily on the visualizations of the ML predictions to compare alternatives and ultimately make decisions. Looking at the specific backgrounds of the biologists, those who usually do not create cell lineage datasets but focus more on other questions in the context of the general problem (e. g., segmentations) were more likely to explore the other functions, including the machine learning prediction details. The participants who work on creating cell lineage would primarily use the lineage assignment or confirmation functionality and would not check the details of the ML visualization. Despite this diverse expertise and range of experience, all biologists assigned cells for at least one level using either approach thanks to our ML visualization and interaction design. Therefore, our staggered way of presenting an increasing amount of detail about the ML predictions is effective at supporting this range of potential users and their needs, and this method can potentially be applied to other domains where experts have a diverse level of knowledge about ML approaches.

During the design and evaluation process, we also found that LineageD+ allowed biologists to interact with the ML predictions in a "communicable" way. One participant (P6) even treated the ML proposals as if the machine learning was suggesting the potential sister cells to her in real-time, and she "explained" to the ML models (i. e., to us in the think-aloud protocol) the reasons why a prediction was reasonable or not. Though she spent more time traversing the proposals, such a workflow was appealing to her and may be similar for other biologists with less experience and knowledge in constructing cell lineage. As she described it in the study, the ML served a similar role as a colleague with whom she would discuss the assignment, and for such a "narrative interaction" an initial assignment proposal (as provided by the ML) and the human's control over the final decision—human-AI teaming—is needed, and this approach could be highly beneficial beyond the specific application we have studied in this work.

6.2 Customized ML Interaction Design

Another interesting point we observed is that the experts generally use the ML models to improve their immediate efficiency, instead of spending time on other tasks with long-term rewards. In our case, e. g., the main purpose is to get the cells assigned. We observed that the biologists tried to complete the study as soon as possible, rather than checking the detailed prediction from ML models—even though the information could have helped them to decide on proper model weights and thus benefited them in the future. To help experts with such situations, when designing a system we need to record their behavior and develop additional ML models to automatically support those tasks—such as adjusting the model weights or the weights of the individual features for future training.

In addition, based on the varied expectations and preferences of the biologists, we saw that we need to support ML predictions by both levels and sections, and potentially even top-down predictions. This means that we really need customized ML support, instead of a single generic black box which cannot easily meet everyone's

requirements. Customized machine learning behaviors and their corresponding visualization—as we provided it in our approach—can be helpful (yet further developments need to add the suggested additional functionality). This ML customization and visualization can also be extended to other scenarios where different target users have diverse working habits. Corresponding interactive visualization can then help users to customize their model use, in addition to providing assistance in interpreting the ML. For instance, visualization can help users to better understand their actual needs to then allow them to pick a suitable model.

6.3 Visualization for and as part of Human-AI Teaming

Our case study thus showed us that visualization can play an essential role in a user's interaction with ML models. In this context we do not see the AI components as a superior authority but instead, as a collaborator with whom one can and should interact. Our visualizations of the details of the ML model predictions thus serve as a mechanism to support this human-AI teaming [25], to ultimately come to the best possible result in a manageable amount of time. This concept certainly does not apply to all applications of machine learning or AI, but it can be useful in those cases where the outcome is crucial, the manual checking is essential and feasible, or where it is likely that the models can make mistakes due to the complex nature of the given problem. It supports the ideas proposed by Heer [15] that we need to design automation based on the nature of the task, and humans need to be the initiators of critical decisions.

In our case, the specific lineage hierarchy that results from the processing with LineageD+ is important because the biologists need to further analyze it. Yet even the biologists themselves are occasionally unsure about some cells' assignments. Moreover, our training datasets come from embryos manually assigned by biologists, and their number is limited (only 93 embryos at this point). Under such circumstances, a single ML model is likely to make mistakes. Though most correctly proposed sisters were predicted by the majority of models, there are a few cases where only one model gave accurate predictions. Although multiple ML models can partially make up for this problem, the biologists would feel more confident if they checked all cell pairs of any new dataset to ensure that they are correctly assigned, as they stated in the study. Consequently, in our application, the biologists are interacting with the machine learning models to come to a final conclusion, as opposed to simply accepting an ML-provided result. The human experts, ultimately, have gained a lot of experience in their education that we may not be able to capture with ML models even for larger training datasets, and in this case, the experts still have the choice to overrule the model-suggested assignments. Such a teaming approach as allowing humans to fully control the essential results can potentially be used for other system designs so that domain experts can produce reliable and satisfying results.

7 Limitations

Naturally, our approach is not without limitations. To start, LineageD+ was built based on LineageD, and some limitations of the latter still apply in our extended approach. First, we chose to apply the ML models to predict the assignments level-by-level. Although some biologists preferred to have the sector-by-sector ML predictions, we designed this because the potential corrections of the current level would invalidate a complete predicted sector. Yet it would be interesting to compare biologists' experience and feedback towards these two different ML appliances. Second, we do not update the ML models based on the interaction patterns of the biologists. While this would technically be possible, the benefit would likely be limited because of the rather small set of training data. Without such an interactive updating process the interaction with the ML is not really a "discussion" as was implied by P6 in our study. Thus, we would be interested in combining our approach with techniques from explainable AI that would allow us to create an environment in which the ML could participate more in a "discussion." Third, in both systems, biologists typically check all cells—even those with the proposed assignments—because they feel more confident after checking every pair. As a result, the lineage process still takes time, especially for large embryos. We would also be interested to include precision in our system to visualize the predictions and help the biologists with their decision-making. Finally, we used a limited set of training datasets (93 embryos) from biologists. A larger number of manually assigned embryos can potentially improve the model performance.

Our specific human-AI teaming approach in LineageD+ and our evaluation have limitations as well. Based on the biologists' traditional workflow, we introduced ML algorithms and improved and designed the usage of LineageD+. This new way of establishing the cell lineage for an embryo dataset is challenging for biologists because they are often not familiar with the use of 3D visualizations and interactions, the use of ML in general or specific models, and also our specific interaction design. They need time to understand the representations of the ML predictions and get trained to do the assignments with such a system. Also, we used only a certain set of features all with equal weight and included only 5 ML models in LineageD+, again with the default weight of 1. Introducing other features with diverse weights may improve the ML models, and including more models can possibly increase the overall precision for biologists. Meanwhile, developing additional ML models to automatically change the feature and model weights could potentially ease the biologists' mental workload. In our evaluation study, we recruited "only" 6 experts in the specific field (plant embryo lineage) to validate our system design, and half of them had used LineageD before so they might have had biases. Other minor usability issues noted by biologists are that, first, the raw data in the first stage may not be clean, because the snapshot may include cells of different generations. In this case, the ML models need to detect the differences. Second, there is a format constraint for the uploaded data file. Though it can be obtained via morphonet packages, we used the mesh data from a paid tool instead of a more generally-used tool. It adds difficulty to the generalization. Moreover, we directly used the segmented data, while such manual cell segmentation is often time-consuming. In addition, we dealt with data including cell boundary labels, and the tool cannot be applied to data without such boundary labels.

8 Conclusion

Traditionally, ML models are frequently used to take over otherwise tedious tasks and can assist people in finishing complicated work. Often, we do not even change anything about the way the ML does its work but treat it as a black-box tool. Sometimes, however, people also check the decision-making process, examine the results, and give feedback to models to improve them. In such scenarios, visualization can serve as a bridge to connect people with ML. In such a process, however, the ML models often dominate the decision and the other parties' (people and visualization) work

or are used to improve the ML's performance. In our project, in contrast, we did not aim to improve the ML models and accepted that any given ML model will have limitations, in particular in cases when the training data is inherently sparse or inconsistent. We then treated the ML as a tool that helps people make decisions but that no longer is the final authority. We then demonstrated that visualization, in this relationship, can empower people to find a balance between their own experience and ML models' proposals and thus engage with the ML as if it was another collaborator. As people have a diverse sets of needs, such human-AI teaming allows them to decide to what degree the ML models should be involved in the decision-making process. More experienced users may avoid the ML model intervention, while people with less experience may rely more on the ML predictions to a larger degree. In this case, the procedure to complete tasks is somewhat of a collaboration work, where the ML models and their visualization are ideally supportive partners by providing the desired information and explaining it clearly. To achieve this goal, in LineageD+ we used five ML models and, in the future, would like to explore other techniques such as Explainable AI and Interactive ML to allow people to understand and use the ML collaborator more effectively. Although we worked in the biological field and in other fields the visualization and needed interaction design may be different, we are confident that our concept of using visualization to support human-AI teaming applies to other domains in a similar way.

Acknowledgments

We thank all participants in our study for their valuable feedback and the AVIZ team for their general comments. The work was partially supported by Inria's Naviscope project and was completed while the first author was a PhD student with Inria and Univ. Paris-Saclay.

Supplemental Material Pointers

Our software is available at `github.com/JiayiHong/LineageD_Plus`. Our pre-registration can be found at `osf.io/2f6uc`. We also provide the images and study materials from this paper, as noted below, at `osf.io/dcek9`.

Images/figures license and copyright

References

[1] D. Bajusz, A. Rácz, and K. Héberger, "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" *J Cheminf*, vol. 7, no. 1, pp. 20:1–20:13, 2015. doi: 10.1186/s13321-015-0069-3

[2] S. Besson and J. Dumais, "Universal rule for the symmetric division of plant cells," *PNAS*, vol. 108, no. 15, pp. 6294–6299, 2011. doi: 10.1073/pnas.1011866108

[3] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-driven documents," *IEEE Trans Vis Comput Graph*, vol. 17, no. 12, pp. 2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[4] L. Breiman, "Random forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324

[5] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J Artif Intell Res*, vol. 70, pp. 245–317, 2021. doi: 10.1613/jair.1.12228

[6] S. Carpendale, "Evaluating information visualizations," in *Information Visualization: Human-Centered Issues and Perspectives*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds. Berlin: Springer, 2008, ch. 2, pp. 19–45. doi: 10.1007/978-3-540-70956-5_2

[7] D. Cashman, S. R. Humayoun, F. Heimerl, K. Park, S. Das, J. Thompson, B. Saket, A. Mosca, J. Stasko, A. Endert, M. Gleicher, and R. Chang, "A user-based visual analytics workflow for exploratory model analysis," *Comput Graph Forum*, vol. 38, no. 3, pp. 185–199, 2019. doi: 10.1111/cgf.13681

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. KDD*. New York: ACM, 2016, pp. 785–794. doi: 10.1145/2939672.2939785

[9] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Proc. VAST*. Los Alamitos: IEEE CS, 2010, pp. 27–34. doi: 10.1109/VAST.2010.5652443

[10] B. J. Dietvorst, J. P. Simmons, and C. Massey, "Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them," *Manage Sci*, vol. 64, no. 3, pp. 1155–1170, 2018. doi: 10.1287/mnsc.2016.2643

[11] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *PNAS*, vol. 95, no. 25, pp. 14 863–14 868, 1998. doi: 10.1073/pnas.95.25.14863

[12] M. R. Endsley, N. Cooke, N. McNeese, A. Bisantz, L. Militello, and E. Roth, "Special issue on human-AI teaming and special issue on AI in healthcare," *J Cognit Eng Decis Making*, vol. 16, no. 4, pp. 179–181, 2022. doi: 10.1177/15553434221133288

[13] Y. Gil, J. Honaker, S. Gupta, Y. Ma, V. D'Orazio, D. Garijo, S. Gadewar, Q. Yang, and N. Jahanshad, "Towards human-guided machine learning," in *Proc. IUI*. New York: ACM, 2019, pp. 614–624. doi: 10.1145/3301275.3302324

[14] B. D. Harper and K. L. Norman, "Improving user satisfaction: The questionnaire for user interaction satisfaction version 5.5," in *Proc. Mid-Atlantic Human Factors Conference*, 1993, pp. 224–228.

[15] J. Heer, "Agency plus automation: Designing artificial intelligence into interactive systems," *PNAS*, vol. 116, no. 6, pp. 1844–1850, 2019. doi: 10.1073/pnas.1807184115

[16] J. Hong, A. Trubuil, and T. Isenberg, "LineageD: An interactive visual system for plant cell lineage assignments based on correctable machine learning," *Comput Graph Forum*, vol. 41, no. 3, pp. 195–207, 2022. doi: 10.1111/cgf.14533

[17] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, "ActiVis: Visual exploration of industry-scale deep neural network models," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 88–97, 2018. doi: 10.1109/TVCG.2017.2744718

[18] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. De Filippi, W. F. Stewart, and A. Perer, "Clustervision: Visual supervision of unsupervised clustering," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 142–151, 2018. doi: 10.1109/TVCG.2017.2745085

[19] B. Leggio, J. Laussu, E. Faure, P. Lemaire, and C. Godin, "Multiscale mechanical model for cell division orientation in developing biological systems," bioRxiv preprint 785337, 2020. doi: 10.1101/785337

[20] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, "Analyzing the training processes of deep generative models," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 77–87, 2018. doi: 10.1109/TVCG.2017.2744938

[21] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans Vis Comput Graph*, vol. 23, no. 1, pp. 91–100, 2017. doi: 10.1109/TVCG.2016.2598831

[22] P. Martinez, L. A. Allsman, K. A. Brakke, C. Hoyt, J. Hayes, H. Liang, W. Neher, Y. Rui, A. M. Roberts, A. Moradifam, B. Goldstein, C. T. Anderson, and C. G. Rasmussen, "Predicting division planes of three-dimensional cells by soap-film minimization," *Plant Cell*, vol. 30, no. 10, pp. 2255–2266, 2018. doi: 10.1105/tpc.18.00401

[23] N. Minc and M. Piel, "Predicting division plane position and orientation," *Trends Cell Biol*, vol. 22, no. 4, pp. 193–200, 2012. doi: 10.1016/j.tcb.2012.01.003

[24] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: overview study and experimental results," in *Proc. ICICS*. Los Alamitos: IEEE CS, 2020, pp. 243–248. doi: 10.1109/ICICS49469.2020.239556

[25] National Academies of Sciences, Engineering, and Medicine, *Human-AI Teaming: State-of-the-Art and Research Needs*. Washington, DC: The National Academies Press, 2022. doi: 10.17226/26355

[26] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Trans Graph*, vol. 31, no. 4, pp. 30:1–30:11, 2012. doi: 10.1145/2185520.2185526

[27] H.-G. Pagendarm and F. H. Post, "Comparative visualization: Approaches and examples," in *Proc. EGViSC*. Wien: Springer, 1994, pp. 95–108, url: elib.dlr.de/37146.

[28] M. Q. Patton, *Qualitative Research & Evaluation Methods: Integrating Theory and Practice*, 4th ed. Los Angeles: Sage, 2014.

[29] G. A. Pavlopoulos, S. I. O'Donoghue, V. P. Satagopam, T. G. Soldatos, E. Pafilis, and R. Schneider, "Arena3D: Visualization of biological networks in 3D," *BMC Syst Biol*, vol. 2, no. 1, pp. 104:1–104:7, 2008. doi: 10.1186/1752-0509-2-104

[30] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, "DeepEyes: Progressive visual analytics for designing deep neural networks," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 98–108, 2018. doi: 10.1109/TVCG.2017.2744358

[31] A. Pierre, J. Sallé, M. Wühr, and N. Minc, "Generic theoretical models to predict division patterns of cleaving embryos," *Dev Cell*, vol. 39, no. 6, pp. 667–682, 2016. doi: 10.1016/j.devcel.2016.11.018

[32] A. Rosset, L. Spadola, and O. Ratib, "OsiriX: An open-source software for navigating in multidimensional DICOM images," *J Digital Imaging*, vol. 17, no. 3, pp. 205–216, Sep. 2004. doi: 10.1007/s10278-004-1014-6

[33] D. Sacha, M. Kraus, D. A. Keim, and M. Chen, "VIS4ML: An ontology for visual analytics assisted machine learning," *IEEE Trans Vis Comput Graph*, vol. 25, no. 1, pp. 385–395, 2019. doi: 10.1109/TVCG.2018.2864838

[34] I. Salvador-Martínez, M. Grillo, M. Averof, and M. J. Telford, "CeLaVi: an interactive cell lineage visualization tool," *Nucleic Acids Res*, vol. 49, no. W1, pp. W80–W85, 2021. doi: 10.1093/nar/gkab325

[35] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji – An open-source platform for biological-image analysis," *Nat Methods*, vol. 9, no. 7, pp. 676–682, 2012. doi: 10.1038/nmeth.2019

[36] W. Schroeder, K. Martin, and W. Lorensen, "The design and implementation of an object-oriented toolkit for 3D graphics and visualization," in *Proc. Visualization*. Los Alamitos: IEEE CS, 1996, pp. 93–100. doi: 10.1109/VISUAL.1996.567752

[37] H. Schulz, "Treevis.net: A tree visualization reference," *IEEE Comput Graph Appl*, vol. 31, no. 6, pp. 11–15, 2011. doi: 10.1109/MCG.2011.103

[38] H. Schulz, S. Hadlak, and H. Schumann, "The design space of implicit hierarchy visualization: A survey," *IEEE Trans Vis Comput Graph*, vol. 17, no. 4, pp. 393–411, 2011. doi: 10.1109/TVCG.2010.79

[39] J. Seo and B. Shneiderman, "Interactively exploring hierarchical clustering results," *Comput*, vol. 35, no. 7, pp. 80–86, 2002. doi: 10.1109/MC.2002.1016905

[40] K. Siau and W. Wang, "Building trust in artificial intelligence, machine learning, and robotics," *Cutter Bus Technol J*, vol. 31, no. 2, pp. 47–53, 2018.

[41] B. Speckmann and K. Verbeek, "Necklace maps," *IEEE Trans Vis Comput Graph*, vol. 16, no. 6, pp. 881–889, 2010. doi: 10.1109/TVCG.2010.180

[42] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady, "explAIner: A visual analytics framework for interactive and explainable machine learning," *IEEE Trans Vis Comput Graph*, vol. 26, no. 1, pp. 1064–1074, 2020. doi: 10.1109/TVCG.2019.2934629

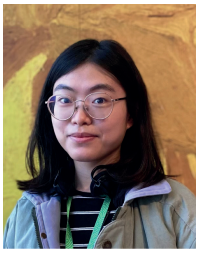
[43] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 667–676, 2018. doi: 10.1109/TVCG.2017.2744158

[44] K. Sugawara, Ç. Çevrim, and M. Averof, "Tracking cell lineages in 3D by incremental deep learning," *eLife*, vol. 11, pp. e69 380:1–e69 380:19, 2022. doi: 10.7554/eLife.69380

[45] G. K. Tam, V. Kothari, and M. Chen, "An analysis of machine-and human-analytics in classification," *IEEE Trans Vis Comput Graph*, vol. 23, no. 1, pp. 71–80, 2017. doi: 10.1109/TVCG.2016.2598829

[46] J.-Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri, "TrackMate: An open and extensible platform for single-particle tracking," *Methods*, vol. 115, pp. 80–90, 2017. doi: 10.1016/j.ymeth.2016.09.016

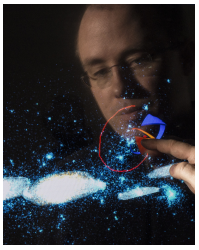
[47] J. Woodring and H.-W. Shen, "Multi-variate, time varying, and comparative visualization with contextual cues," *IEEE Trans Vis Comput Graph*, vol. 12, no. 5, pp. 909–916, 2006. doi: 10.1109/TVCG.2006.164



Jiayi Hong is a postdoctoral fellow at Arizona State University in the School of Computing and Augmented Intelligence. She has received her PhD from the Université Paris Saclay, Inria, France, where also most of the work of this article was done. Previously, she received her Master's degree at the University of Bristol, UK, and her Bachelor's degree at Zhejiang University, China. Her work focuses on exploring the combination of 3D and 2D representations and visualizations for machine learning.

Ross Maciejewski is a professor at Arizona State University in the School of Computing and Augmented Intelligence and Director of the Center for Accelerating Operational Efficiency—a Department of Homeland Security Center of Excellence. His primary research interests are in the areas of visualization and explainable AI.

Alain Trubuil is a senior research scientist emeritus at InraE, France. His research interests include image processing and analysis, modelling of cellular process, scientific visualization. He is particularly interested in the analysis of information from 3D and 3D+Time data obtained from the observation of cellular biological process using advanced microscopies.



Tobias Isenberg is a senior research scientist at Inria, France. Previously he held positions as post-doctoral fellow at the University of Calgary, Canada, and as assistant professor at the University of Groningen, the Netherlands. His research interests include scientific visualization, illustrative and non-photorealistic rendering, and interactive visualization techniques. He is particularly interested in interactive visualization environments for 3D spatial data.

APPENDIX

A: QUESTION LIST

Before the study:

- Do you use ML in your professional work? Could you please explain in details?
- Is this the first time you have used LineageD?
- What expectations do you have for the use of machine learning in the context of establishing the cell lineage?
- How often do you create cell lineage datasets? (Once a year, once a month, once a week, several times a week, daily or more often, or I do not create cell lineage datasets but I work on this general problem)

After the study:

- How do you think this approach/tool compares to the traditional ImageJ tool/your previous tools (explain what) in doing the cell lineage?
- In the LineageD+, how do you generally feel about the visual representations that we use?
- What do you generally think of the ML support in LinageD+?
- Do our visual representations of the ML models support your decision process? Or are they confusing? Explain please.
- How do you feel about the interaction with the different ML models that we provide? Explain please.
- What specific parts/elements do you like or dislike?
- What interaction or visualization do you miss?
- Do you think LineageD+ changes how you approach the assignment process (strategies you used)? If yes, in what way? If not, why not?
- Would you change something about how the ML is being employed?
- After sufficient training, how fast do you think it would be for you to construct the cell lineage, compared to your traditional approach?
- Again after sufficient training, how confident do you think you feel about the result that achieved establishing the lineage with the new tool, compared to your traditional approach?