



HAL
open science

Machine Learning Supported Interactive Visualization of Hybrid 3D and 2D Data for the Example of Plant Cell Lineage Specification

Jiayi Hong

► **To cite this version:**

Jiayi Hong. Machine Learning Supported Interactive Visualization of Hybrid 3D and 2D Data for the Example of Plant Cell Lineage Specification. Human-Computer Interaction [cs.HC]. Université Paris-Saclay, 2023. English. NNT : 2023UPASG006 . tel-04016701

HAL Id: tel-04016701

<https://theses.hal.science/tel-04016701>

Submitted on 6 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Machine Learning Supported Interactive Visualization of Hybrid 3D and 2D Data for the Example of Plant Cell Lineage Specification

*Visualisation interactive, soutenue par l'apprentissage
automatique, de données hybrides 3D et 2D; l'exemple
de la spécification du lignage cellulaire en
biologie végétale*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580 : Sciences et technologies de l'information et de la
communication (STIC)

Spécialité de doctorat : Informatique

Graduate School : Informatique et Sciences du Numérique. Référent : Faculté des
sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire interdisciplinaire des sciences
du numérique (Université Paris-Saclay, CNRS, Inria)**, sous la direction de **Tobias
ISENBERG**, Directeur de recherche et la co-direction de **Alain TRUBUIL**, Ingénieur de
recherche.

Thèse soutenue à Paris-Saclay, le 14 Février 2023, par

Jiayi HONG

Composition du jury

Membres du jury avec voix délibérative

Caroline Appert Directrice de recherche, Université Paris-Saclay	Présidente
Barbora Kozlíková Professeure Agrégé, Masaryk University	Rapporteuse & Examinatrice
Miriah Meyer Professeure, Linköping University	Rapporteuse & Examinatrice
Anastasia Bezerianos Professeure, Université Paris-Saclay	Examinatrice
Johanna Beyer Chargée de recherche, Harvard University	Examinatrice

SYNTHÈSE

Mots-clés : Données 3D, visualisation scientifique, visualisation abstraite 2D, interaction/HCI, apprentissage automatique

Les technologies graphiques actuelles permettent de visualiser efficacement des données scientifiques spatiales dans un environnement 3D afin que les utilisateurs puissent observer clairement leurs formes et positions 3D. Parallèlement, les visualisations abstraites 2D peuvent résumer des informations, montrer des données supplémentaires et contrôler une vue 3D. Combiner ces deux vues en une seule interface peut aider les utilisateurs à entreprendre des tâches complexes bien qu'il y ait un manque de recommandations sur l'intégration. En général, les experts analysent de grands volumes de données pour des travaux complexes, comme la reconstruction du lignage cellulaire d'embryons en biologie. Ici, le travail d'annotation peut être fastidieux, et les algorithmes d'apprentissage automatique peuvent l'alléger, en fournissant des prédictions initiales aux experts. Dans le cas du lignage cellulaire, ces prédictions contiennent toutefois des informations hiérarchiques et multicouches qu'il est essentiel de visualiser de manière séquentielle ou progressive afin d'accompagner les experts. De plus, les représentations 2D et 3D ainsi que les prédictions d'apprentissage automatique doivent être connectées visuellement et interactivement.

Dans cette thèse, nous prenons comme exemple le problème du lignage cellulaire des embryons de plantes pour concevoir et étudier un système de visualisation qui combine des vues 3D et 2D ainsi que des visualisations pour l'apprentissage automatique. Les cellules d'un embryon de plante sont jointives et constituent un ensemble dense d'objets 3D. Tout d'abord, nous évaluons trois techniques de sélection et nous montrons que la combinaison de la Sélection

par Explosion et de la Sélection par Liste est efficace pour désigner et observer les cellules d'un embryon. Ces techniques sont généralisables à d'autres données 3D similaires. Ensuite, nous décrivons un système hybride de visualisations et d'interaction que nous avons conçu afin que les biologistes puissent examiner les cellules d'un embryon et enregistrer l'histoire du développement dans l'arbre de lignage hiérarchique. Nous gérons la construction de la hiérarchie dans deux directions, à la fois en construisant l'historique de l'arbre de haut en bas en utilisant la sélection lasso, et de bas en haut selon l'approche usuelle des biologistes. Nous avons par ailleurs proposé un modèle de réseau neuronal pour fournir aux biologistes des prédictions initiales sur les filiations. Nous avons évalué le système avec des biologistes et concluons que les représentations 3D et 2D facilitent les prises de décisions et que l'outil enrichit la visualisation des embryons. Cependant, la performance du modèle d'apprentissage n'était pas idéale. Aussi, pour faciliter le processus de lignage et améliorer les performances du système, nous avons entraîné cinq modèles de classification différents en montrant leurs prédictions et leurs incertitudes. Notre étude montre que les représentations des classifieurs sont faciles à comprendre et que ce nouvel outil peut améliorer significativement les prises de décision.

Nous résumons nos conclusions comme suit. Tout d'abord, dans le contexte du lignage cellulaire, il est essentiel d'adopter de nouvelles techniques d'interaction, plus élaborées que les techniques usuelles. Deuxièmement, combiner les représentations 3D avec les visualisations abstraites 2D est bénéfique dans le domaine scientifique; nous l'avons montré pour la biologie et le lignage cellulaire. Troisièmement, les visualisations peuvent aider à la compréhension des prédictions des modèles d'apprentissage sur lesquelles sont basées des décisions. Tous ces résultats peuvent inspirer de futurs travaux visant à combiner, de manière transparente, diverses représentations dans une seule interface, ceci afin d'aider les experts à prendre de meilleures décisions.

PUBLICATION

Full papers - Accepted

1. **Jiayi Hong**, Alain Trubuil, and Tobias Isenberg. "LineageD : An Interactive Visual System for Plant Cell Lineage Assignments based on Correctable Machine Learning". In Computer Graphics Forum 41.3 (EuroVis 2022), pp. 195–207. DOI : 10.1111/cgf.14533.
2. **Jiayi Hong**, Ferran Argelaguet, Alain Trubuil, and Tobias Isenberg. "Design and Evaluation of Three Selection Techniques for Tightly Packed 3D Objects in Cell Lineage Specification in Botany". In Proceedings of Graphics Interface. Mississauga, ON, Canada : Canadian Human-Computer Communications Society, 2021, pp. 213–223. ISBN : 978-0-9947868-6-9. DOI : 10.20380/GI2021.33.

Full papers - Submitted and under review

1. **Jiayi Hong**, Ross Maciejewski, Alain Trubuil, and Tobias Isenberg. "Comparative Visualization of Machine Learning Predictions to Improve Human-AI Teaming on the Example of Cell Lineage". IEEE Transactions on Visualization and Computer Graphics.

Poster Communications

1. **Jiayi Hong**, Ferran Argelaguet, Alain Trubuil, and Tobias Isenberg. "An Interactive System for Analyzing Plant Embryo Cell Division". In Posters at the IEEE Conference on Visualization (IEEE VIS 2020).
2. Ebrar A. D. Santos, **Jiayi Hong**, and Tobias Isenberg. "A Design Space for Linked 2D and 3D Visual Representations". In Posters at the IEEE Conference on Visualization (IEEE VIS 2022).

ACKNOWLEDGMENTS

Three years ago, I started my PhD almost the same time as Covid-19 started to spread. Everyone had a tough time more or less during the global pandemics, but people around still helped me so much. I cannot make it without the support from them, including my colleagues, my collaborators, my friends, and my families.

First of all, I would really like to thank my supervisor Tobias Isenberg. He is the best supervisor, mentor, and educator I could ever imagine. He is the first one who told me that my ideas matter even when I was a novice researcher who just started PhD. He patiently taught me what is visualization, how to do research, how to prepare myself, and how to be independent in my academic career. Even when I was disappointed with myself, he always encouraged me to cheer up and continue with my work. I learned so much from him during the past three years, not only in research, but also in being a good person. I think I can learn from him for a lifetime. Alain Trubuil is a patient and nice co-supervisor who I am also really grateful for. He explained me biological knowledge I need to know, introduced biologists to me, and insisted to extend his career after he retried for attending my defense. I also want to express gratitude for jury members' valuable feedback.

My thesis used the example of plant cell embryos, and I collaborated with biologists. For that, I really want to thank dedicated biologists, Philippe Andrey, Jean-Christophe Palauqui, and Elise Laruelle, to provide the training datasets, answer my questions, offer ideas, and give feedback about my work. I am also so thankful to have all the participants to help me evaluate and improve my work. Though I cannot list their names here, I have to express my deepest gratitude to them for their time and kindly help. Thanks should also go to my collaborator Ferran Argelaguet and Ross Maciejewski for their helpful suggestions. I would

also mention Steff Laurent, the engineer from Inria, without whom I would have suffered more about the web deployment.

I felt so lucky to have my PhD within Aviz group. It is such a friendly team with full of wisdom. I would specially thank Petra Isenberg who gave me many valuable suggestions on my work and on my life. Jean-Daniel Fekete inspired me with fascinating ideas and helped me a lot to deeply experience French life. Also, I would much appreciated other Aviz members and visitors who inspired me and made my PhD years wonderful, including Pierre Dragicevic, Frédéric Vernier, Nivan Ferreira, Catherine Plaisant, Raimund Dachsel, Steve Haroz, Luiz Morais, Gaëlle Richer, Paola Valdivia, Xiyao Wang, Natkamon Tovanich, Florent Cabric, Mickaël Sereno, Tingying He, Alaul Islam, Alexis Pister, Lijie Yao, Yuheng Feng, Katerina Batziakoudi, Fairouz Grioui, Sarkis Halladjian, Ebrar A. D. Santos, and Federica Bucchieri.

I am also so grateful to have my family support me as they always do. Without their support, I cannot freely do whatever I want. They are my strongest backing. I also sincerely give credit to my friends. No matter where they were, they always helped me when I needed.

TABLE DES MATIÈRES


1	Introduction	23
1.1	Interactive Visualization	26
1.2	Combination of 3D and 2D Representations	28
1.3	Machine Learning	28
1.4	Thesis Statement	29
1.5	Thesis Overview	30
2	Background	33
2.1	Plant Embryo Datasets	33
2.1.1	Embryos	34
2.1.2	Cell Lineage	35
2.2	Cell Lineage Problem	36
2.2.1	Traditional workflow	36
2.3	Considerations	39
3	Systematic review	43
3.1	Visual System in Biology	43
3.2	Hybrid System Combining 3D and 2D Representations	46
3.3	Visualizations for Machine Learning	50
3.3.1	Understanding	52
3.3.2	Machine learning for cell lineage	52
3.4	Summary	53
4	Exploration of 3D selection techniques for plant embryos	55
4.1	Introduction	56
4.2	Related Work	58

4.2.1	Discovery and Access Techniques	58
4.2.2	Interaction Techniques for Dense Datasets	60
4.2.3	Cell Visualization	61
4.3	Study Design	61
4.3.1	Interaction Techniques	61
4.3.2	Tasks	63
4.3.3	Datasets	64
4.3.4	Interface	65
4.3.5	Measures	66
4.3.6	Participants	67
4.3.7	Procedure	67
4.4	Results	68
4.4.1	Completion Time	69
4.4.2	Accuracy	72
4.4.3	Clicking Ratio	73
4.4.4	Techniques Used in Combination	74
4.4.5	Task Strategies	75
4.4.6	Subjective Feedback	76
4.5	Discussion	78
4.5.1	Performance Differences	78
4.5.2	Subjective Ratings	79
4.5.3	Implications	79
4.5.4	Limitations	81
4.6	Conclusion	81
5	LineageD : Interactive Visual System for Plant Cell Lineage Assignments Based on Machine Learning	83
5.1	Introduction	84
5.2	Related Work	86
5.2.1	Visualization of and with Hierarchies	86
5.2.2	Biology-related Visualization	87
5.2.3	Machine Learning in Visualization	88
5.3	Background and Design Considerations	88
5.4	System Design	91
5.4.1	Front-end Interface	91
5.4.2	Server-side Back-end	94
5.4.3	Interaction Design	96
5.4.4	Implementation Details	98
5.5	Empirical Case Study Evaluation	100
5.5.1	Participants	100
5.5.2	Procedure	101

5.5.3	Study results	101
5.6	Discussion	104
5.7	Limitations and Future Work	106
5.8	Conclusion	107
6	Comparative Visualization of Machine Learning Predictions to Improve Human-AI Teaming on the Example of Cell Lineage	109
6.1	Introduction	110
6.2	Related Work	112
6.2.1	Visualization of Hierarchical Information in Biology	113
6.2.2	Cell Lineage Reconstruction	113
6.2.3	Visualization of ML Model Output	114
6.3	System Design	115
6.3.1	Design Considerations	115
6.3.2	ML Features, Model Training, and Prediction	116
6.3.3	Visualization Design	120
6.3.4	Further Interaction Design and Decision Making	123
6.4	Implementation Details	124
6.4.1	Datasets	124
6.4.2	Models	125
6.4.3	Visualizations	126
6.5	Case Study	126
6.5.1	Datasets	127
6.5.2	Participants	127
6.5.3	Procedure	127
6.5.4	Study Results	128
6.6	Discussion	132
6.6.1	Various Interaction Types with ML Visualizations	132
6.6.2	Customized ML Interaction Design	133
6.6.3	Visualization for Human-AI Teaming	134
6.7	Limitations	135
6.8	Conclusion	136
7	Discussion and Conclusion	139
7.1	Interacting with densely-packed objects	139
7.2	Designing for Lineage	141
7.3	Design for Domain Experts	143
7.4	Combination Design for 3D and 2D Representations	145
7.5	Visualization for Machine Learning	148
7.6	Future Work	151
7.7	Summary	152

TABLE DES FIGURES

1.1	The illustration of three possible types of cell division : arithmetic increase, geometric increase, and combination of arithmetic and geometric increase. Figure from the book [126]. Image © Jones & Bartlett Learning, used with permission.	24
1.2	The illustration of the hierarchy tree representing the cell division and the corresponding possible slices from the microscope. Each row represents one generation. Image © Alain Trubuil, used with permission.	26
1.3	The interface with two 3D selection techniques (Explosion Selection on the left side and List Selection on the right side) we developed.	27
1.4	The design of combining 3D and 2D representations. Selecting cells from either the abstract view or the 3D environment, the corresponding cell will be highlighted in the other view.	29
2.1	The illustration [94] demonstrates the developing history of an Arabidopsis plant. Image © Springer Nature, used with permission.	34
2.2	The sequential up-down slices from the same 16-cell plant embryo. Contrast enhanced to make the ID colors visible.	35
2.3	The typical cell lineage of the plant embryo. The naming rules here follow their traditional habits. As an embryo divides, the hierarchy tree is built from the top down, while when biologists are reconstructing the tree, it is built from the bottom up.	36

- 2.4 The traditional system TreeJ (a plugin to Fiji ImageJ) that the experts traditionally use to specify the cell lineage : (a) screenshots of the interface. 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 A and D) only serves as an additional view and is also not used for interaction, and (b) 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. 37
- 2.5 The illustration of the recorded format. The upper part is the array biologists would save in a text file, and below is the corresponding tree in 2D. 38
- 2.6 Embryos slices got from the microscope. From left to right, embryos consist of 16, 32, and 64 cells respectively. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission. 38
- 2.7 Traditional trees to record the embryos' developing history : (a) the hierarchy tree biologists wrote on the paper, (b) the tree used in TreeJ. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission. 39
- 2.8 The screenshot of Fiji ImageJ [162] deals with the slices of a 16-cell embryo. The lower slider bar enables users to target a specific picture among the whole stack of slices. The figure here illustrates the segmented embryo. 40
- 3.1 The screenshot of MorphoNet [110]. Image permissions by Leggio et al. [110],  CC-BY 4.0. 44
- 3.2 The illustration of typical types of trees. The left image is an example of a node-link diagram. The middle image is a treemap, a typical instance of space-filling diagrams. The right one is the Elastic Hierarchy developed by Zhao et al. [218]. Image © 2005 IEEE. 46
- 3.3 The screenshot of 3D-Cardiomics [140]. The heart model is expanded in the view, and two 2D panels add information to the 3D representations. Users can hover over specific parts to see the details and use the buttons on the panel to target a specific gene. . . 47
- 3.4 The illustration of the hybrid 2D and 3D to visualize trajectory attribute data [195]. Image © 2012 IEEE. 48

3.5	The illustration of Abstractocyte [138]. From left to right, the neurons and astrocytes are abstracted in different levels. Image © 2017 IEEE.	49
3.6	The screenshot of CeLaVi [159] using the default Ciona dataset. The hierarchy tree is placed on the left side, while the 3D view of cells is on the right side.	50
3.7	The design space we came up of combining the 3D and abstract 2D representations [160].	51
3.8	The workflow proposed in explAIner [176]. As we can see from the figure, there are multiple explainers being applicable in the process, and visual explanations can be added in between. Image © 2019 IEEE.	51
4.1	Our original system : (a) the main interface with three selection techniques and (b) the hierarchy tree for 8-cell embryo. The tree contains the 3D interactive models of corresponding cells.	56
4.2	Plant embryo dataset with 201 cells (87 “occluded” cells) : (a) a segmented cross section from confocal microscopy, (b) the 3D model, and (c) a part of the desired cell lineage tree—the botanists’ goal to be able to study the embryo’ development. The shared area of any specific cell indicate the shared area between itself with the parent cell. If a cell has no parent cell, like the cell 413 in this case, the shared area is the surface area.	57
4.3	Three main interaction targets for the techniques compared in the study : (a) List, (b) 3D Explosion, and (c) Combination selection. Cells to be assigned are marked in orange, and selected cells are red. In all three cases the 3D view was visible to the participants.	62
4.4	The focused view of a target cell and the associated number shown near the neighbor cell’s surface (red cell is the target cell and yellow cell is the neighbor cell with its associated number).	63
4.5	Study interface (combination selection shown).	65
4.6	Completion time (absolute mean time) for different numbers of cell neighbors in seconds : (a) overall time, (b) <i>List</i> selection, (c) <i>Explosion</i> selection, and (d) <i>Combination</i> selection.	69
4.7	Completion time (absolute mean time) in seconds (<i>List</i> in yellow and <i>Explosion</i> in red) : (a) the overall results, (b) selection of occluded cells, and (c) selection of surface cells.	70
4.8	Pair-wise differences for completion time : (a) the ratio overall, (b) the ratio for occluded cells, and (c) the ratio for surface cells.	70

4.9	Completion time (absolute mean time) in seconds with two steps (<i>List</i> in yellow and <i>Explosion</i> in red) : (a) the target cell discovery, and (b) neighborhood traversal.	71
4.10	Pair-wise differences for completion time in two steps : the ratios for (a) discovery and (b) traversal.	71
4.11	Accuracy rate (<i>List</i> in yellow and <i>Explosion</i> in red) : (a) overall, (b) selection of occluded cells, and (c) selection of surface cells.	72
4.12	Pair-wise differences for accuracy : (a) the ratio overall, (b) the ratio for occluded cells, and (c) the ratio for surface cells.	72
4.13	Clicking Ratio (<i>List</i> in yellow and <i>Explosion</i> in red) : (a) overall, (b) selection of occluded cells, and (c) selection of surface cells.	73
4.14	Pair-wise differences for clicking ratio : (a) the ratio overall and (b) the ratio for occluded cells, and (c) the ratio for surface cells.	74
4.15	Clicking proportions of <i>List</i> / <i>List</i> + <i>Explosion</i> in the <i>Combination</i> task : (a) overall and (b) by neighbor count (for discovery + traversal; x represents the numbers of the cell neighbors, and y represents the clicking proportions).	75
4.16	Accumulated participant preference ranks. Note that we allowed participants to rank two techniques as their first choice and then counted none as the second, resulting in ranks 1, 1, and 3.	77
5.1	An illustration of the traditional way of biologists doing cell lineage based on slices.	85
5.2	A 2D illustration of biologists' cell lineage workflow.	89
5.3	Screenshot of LineageD, for the example of assigning a 256-cell embryo (in progress). The core elements are the Main 3D View of the embryo, here exploded and with the target cell highlighted, and the Hierarchy Tree of the lineage, which is interactively established by the biologists. The Target and Sister View shows the relative position of the target cell within the whole embryo and its proposed sister. The Thumbnail View of the tree provides the context of the full hierarchy. Finally, the Operation Panel supports further control of the tool.	92
5.4	An illustration of our reasoning for how we designed the lineage hierarchy tree representation, based on the traditional hand-written or (ASCII) text-based records used by the biologists.	93
5.5	Several interaction techniques in LineageD.	95
5.6	Illustration of the typical interaction workflow of LineageD.	99

- 6.1 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). (c) The model results of the target cell (cell 64) and the interested proposed sister cell (cell 37) being sisters. 112
- 6.2 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 $cell4$ 118
- 6.3 For each quarter (here we take the quarter Cell N-8 as an example), we predicted the new level with the quarter 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 subtree 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 subtree 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 a greenish color. For the cell 83, the ML models are 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. 120
- 6.4 The details of model predictions for one cell. Here we take cell 9 as the target cell for example. 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. 121

- 6.5 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. 122
- 6.6 When hovering over a specific proposed sister cell in the semi-donut chart (as we did in Figure 6.4, 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 6.5, 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. 123
- 6.7 The connection of visualization design in the different information levels. The gray part is the uncertainty where models can make mistakes about the target cell (cell 9 in this example). 124
- 6.8 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 in pink. In this figure we highlighted the similar cell with a red box. 125
- 6.9 The workflow of using LineageD to do cell lineage. The steps with white backgrounds are based on our previous work [84], and the steps in orange represents the new stages we added in LineageD. 126
- 6.10 The interaction process of checking the model predictions. Biologists can target one cell from either the 3D view or the 2D hierarchy 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 prediction and make an informed decision. 137
- 6.11 The illustration of the hovering effect. In this figure, users hover 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. 138
- 6.12 The illustration of users targeting one subtree. Here a user focuses on Cell N-11. The 3D view only presents the cells within this part, and in the hierarchical tree, we make the other cells transparent. 138

7.1	The screenshot of GeoTime [90] to combine both temporal and geospatial visualizations. Image © 2005 SAGE Publications, used with permission.	142
7.2	The illustration of balancing traditional workflow and new design with advanced technologies to solve a scientific problem. The new design usually involves balancing different techniques inside, e. g., visualization, interaction, and machine learning.	144
7.3	The illustrations of potential design with (a) cell names on 3D and (b) both the names and lines representing the relationship of two cells being sisters.	145
7.4	We duplicated this figure from Figure 3.7. In this poster [160], we made five categories to group literature which contains 3D and 2D representations.	146
7.5	In the screenshot of the interface developed by Eulzer et al.[64], we can see that the 2D diagram (part 2) is the flatten representation of 3D mitral valve (part 1). Image © 2020 IEEE.	147
7.6	In this example from the work by Eulzer et al. [63], a pack of 2D diagrams are attached to eight spinal discs, and when users rotate the 3D objects, the 2D representations would rotate accordingly. Image © 2020 IEEE.	148
7.7	Sedlmair et al. [167] proposed two axes (task clarity and information location) for identifying the usage scope of design study methodology. © 2012 IEEE.	149
7.8	Based on the Figure 7.7, we added an area between Algorithm Automation and Design Study Methodology. In that area, we recommend using human-AI teaming to combine the advantages of human thinking and AI automation.	150
7.9	The illustration of the peeling process to solve a scientific problem. The layers refer to diverse techniques (visualization, machine learning, and interaction) or combinations of these techniques.	153

LISTE DES TABLEAUX

- 6.1 The overall accuracy rate and recall value for the five models using the cross-validation approach. 119

INTRODUCTION

Visualization is the field of research that deals with creating computational visual representations to efficiently help with the tasks of users [143]. Today, it has become a common element of our daily life, such as weather data diagrams on our mobile phones and distribution cost charts from our banks. When used in the professional domain, like biology, visualizations can additionally help experts to gain insights from raw data [79]. Normally the domain-specific datasets are complicated, and it is challenging for experts to detect the patterns from such raw data. Since human beings are good at dealing with visual information [194], visualizations for raw datasets can help users to have a better understanding. For instance, in botany, the structures of membranes and cytokinesis could be too abstract for students to understand, while visualizations could help illustrators to visually better demonstrate such processes in the textbook [126]. As a result, it can potentially assist experts in effective decision making. Among all the domains, biology is a natural science exploring life which is a complex field not fully understood. Adopting the techniques to present living organs and their development process (e. g., the structure of organs and the hierarchical history) could largely improve the expressiveness and inspire viewers with new insights.

The specific problem we are addressing in this thesis is the **cell lineage** problem. Cells in biological organisms generally keep dividing in their development (see Figure 1.1), and such divisions could be symmetric or asymmetric depending on species, generations, and cells' positions. The asymmetric division is likely to produce new cell types that function differently [153]. Such development history is called cell lineage in biology. Understanding the cell lineage is essential because it could help botanists to (1) check whether an embryo develops normally or not, (2) specify the functions of different divided parts, (3) predict the fate of

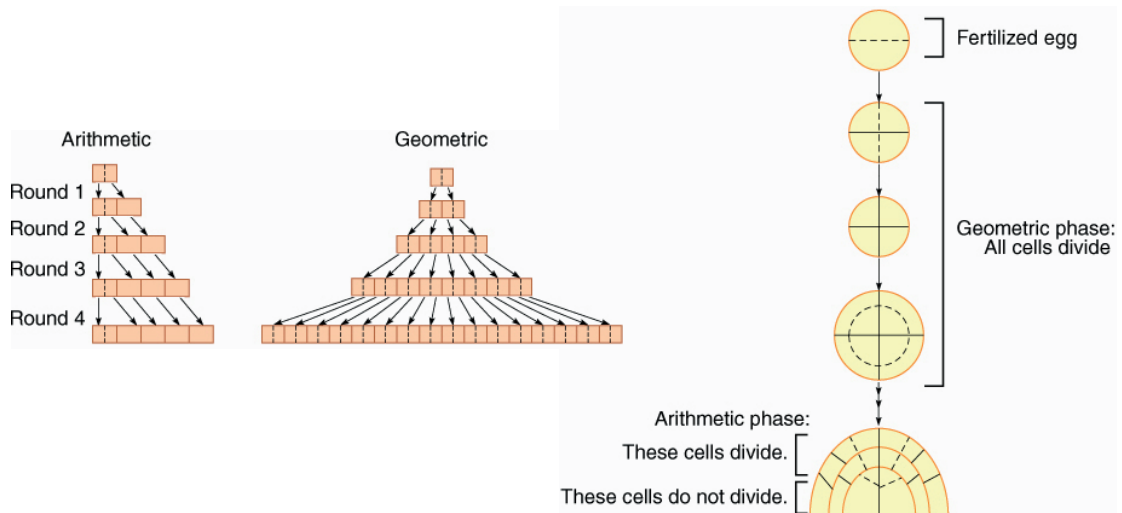


Figure 1.1 – The illustration of three possible types of cell division : arithmetic increase, geometric increase, and combination of arithmetic and geometric increase. Figure from the book [126]. Image © Jones & Bartlett Learning, used with permission.

certain cells in a pluripotent embryo at the early stage, and (4) effectively detect the potential reasons when one embryo gets abnormal [179]. Researchers have been investigating the cell lineage of diverse plant species since the 19th century, and it is still a challenging problem now because the workflow of establishing the cell lineage is quite time-consuming. Though computer technologies are developing fast, biologists seem to keep their traditional manual approach to making the assignments and building the corresponding tree (see Figure 1.2). The major drawbacks of this process are that, once the tree is set, it is hard to share and communicate the tree with others to dynamically and vividly show the division process. Meanwhile, biologists could even spend hours assigning for a single big embryo that contains about 256 cells, which means that machine learning algorithms have great potential to speed up the process and ease the workload of biologists. On the other hand, the number of manually assigned plant embryo datasets is limited, and the resulting lack of training datasets adds difficulties in getting an optimal model. In this case, visualizations are designed for better analysis. Thus, we collaborated with biologists to use visualizations, machine learning, and their related fields to help with this problem.

In this thesis, we collaborated with plant biologists from INRAE (French National Research Institute for Agriculture, Food, and Environment) to work on the plant embryo datasets, namely Alain Trubuil, Philippe Andrey, Jean-Christophe Palauqui, and Elise Laruelle. At the beginning of the thesis, we had several meetings with biologists to understand their problems. We observed how they used

the traditional tools to do cell lineage and their issues in finishing the assignments. They also provided us with initial ideas about the potential solutions. Then we discussed with them the feasibility of these solutions and whether there were better solutions. After we had agreements, we started to produce some sketches and prototypes for them to review. The collaboration follows such a process. Every time when we had some ideas and implemented some functions, we had meetings with biologists to get their feedback. We iteratively refined our solutions back and forth via such regular meetings with biologists and got the final design in the end.

As we found in our iterative design approach, to help biologists with the cell lineage process, a tool needs to provide biologists with detailed information for plant embryos (3D embryo data and temporal development data), basic assignments predictions to start with, and the ability to interactively target and check the predictions and adjust them. There are also additional requirements for the design. The first is about the display of cells in an embryo. Different from many other 3D datasets, plant cells are densely packed together without any space between each other. This leads to occlusion, which cannot be easily solved by existing 3D selection techniques and tools. The second is about biologists' traditional working habits. Biologists are so used to working with 2D diagrams that they are not familiar with 3D interactions. Thus, the interaction techniques connecting the 3D environment and the 2D tree should be smooth and intuitive. Lastly, there are currently tedious, repetitive steps in doing the cell lineage. A system should use advanced techniques to reduce redundant work and enhance working efficiency.

To conclude, in the field of cell lineage research, there are three major challenges : (1) the cells are **tightly packed** together in an embryo which makes the interaction with spatial data difficult, (2) the demand for **closely linking 2D hierarchical data and 3D** cells requires a new design for an interactive system, and (3) the manual process is time-consuming, and **machine learning algorithms** could help speed up the process but for which biologists have to be able to make informed decisions. Our thesis is structured to address these challenges, and we will explain them in detail below. Note that though we targeted the plant cell lineage problem in this thesis, the methodology and solutions can be used in other similar problems. For example, the design for the second point can be adopted in other systems with both 3D and 2D. We discuss this in detail in chapitre 7.

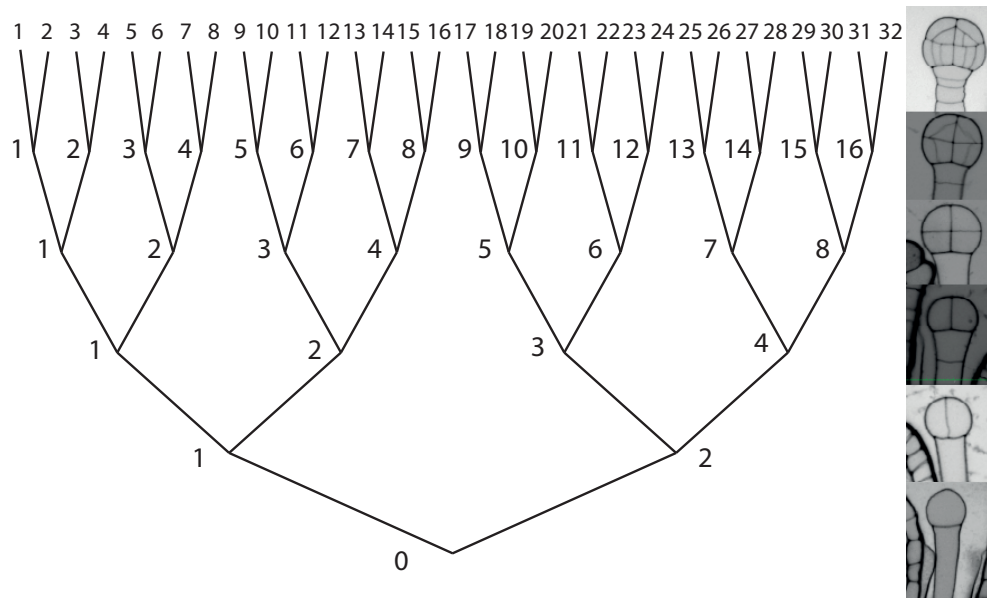


Figure 1.2 – The illustration of the hierarchy tree representing the cell division and the corresponding possible slices from the microscope. Each row represents one generation. Image © Alain Trubuil, used with permission.

1.1 INTERACTIVE VISUALIZATION

First, visualizing biological phenomena is complicated [70]. One reason is that the phenomena themselves are complex. The complexity lies in the multiplicity of life's processes, the diversity of molecular species, and the adaptability to variable environments. Another reason is that these datasets usually contain diverse data types (abstract and concrete, 3D and 2D, and temporal). To illustrate this kind of biological data well, we need to use visualizations to demonstrate various parts such as the biological structure and the development history in our case so that we can inspire researchers [54]. Therefore, for the first two challenges mentioned in section 1, adopting interactive visualization is a potential solution because it can help with the occlusion problem and provide visual representations to help people effectively complete the tasks [143]. Interactive visualization includes two components : interactivity and visualization. As the definition of Human-Computer Interaction (HCI) shows, interactivity is an art of methods design with which people can communicate with the computer system [49]. Two fundamental factors in the HCI field are functionality and usability [91]. Functionality indicates what functions the system could provide and what kind of problem could be solved with this tool, while usability is about how friendly and

efficient the system could be used. In our case, all the functions should serve for the task, which is cell assignments. Especially for a tightly packed embryo that has no space between two adjacent cells, the interaction techniques should effectively help users to solve the occlusion problem. Also, experts need intuitive interactions to check and edit the assignments. In short, interactivity is crucial for our system because biologists need to investigate the data and complete the tasks manually. In parallel, general embryo data and its traditional hierarchy tree are hard to refer to and recall. Visualizations could make full use of people's visual perception to recognize data shapes and changes and detect potential division patterns [26]. Therefore, combining visualizations with interactivity can largely increase the visualization's ability to express and reveal information. In our specific case, there should be both 3D and 2D visualizations so that the interactivity also needs to include the connection between these two parts. Thus, in the thesis, we target to provide interactive visualizations so that biologists can have a better overview of embryos in the 3D environment and the developing history in the hierarchy tree. Also, it could improve the efficiency of examining cells and assigning sisters to them.

For the first challenge in section 1, with the embryo visualized in 3D, we developed selection techniques (1D list, 3D explosion, and the combination of these two techniques) to target inside cells (see Figure 1.3). We also conducted a study to investigate the interactions. We present our work for solving the second challenge in the next section.

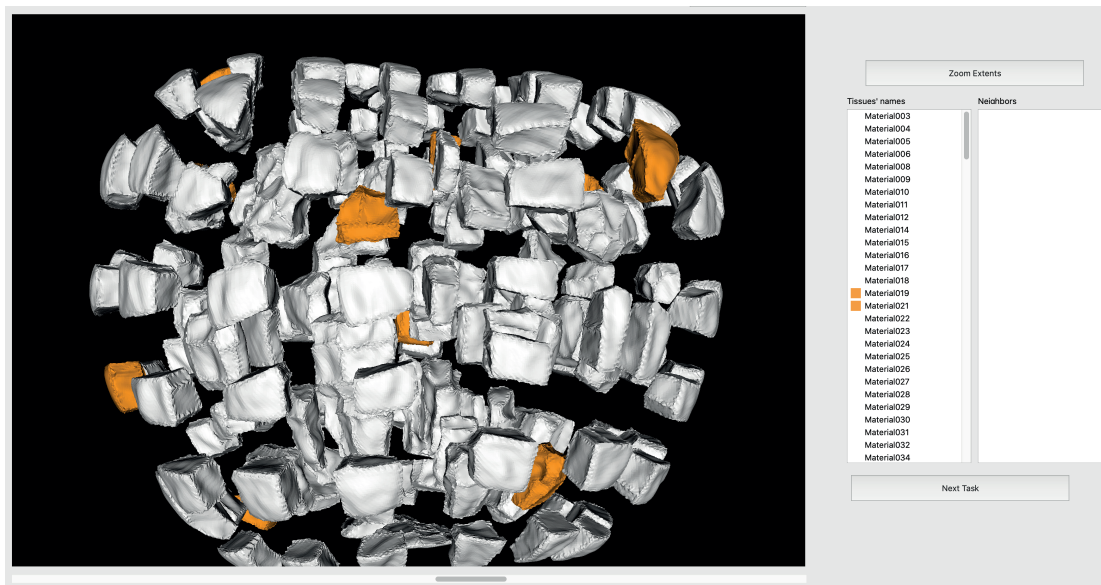


Figure 1.3 – The interface with two 3D selection techniques (Explosion Selection on the left side and List Selection on the right side) we developed.

1.2 COMBINATION OF 3D AND 2D REPRESENTATIONS

As computer graphics technology advances, people can better visualize spatial datasets nowadays. 3D representations can intuitively present real spatial information, such as 3D shapes and positions. Meanwhile, 3D environments usually require 3D operations such as orientating, which makes it difficult to examine numerical information. 2D abstract representations could provide additional data or help highlight and summarize the important parts, on the contrary. Combining these two views can provide biologists with detailed 3D information to make better decisions for assignments and provide an interactive hierarchy tree to record the embryo's development history. Such 2D and 3D combinations are commonly viewed in Virtual Reality [212, 214]. For example, in work by Hurter et al. [88], the authors used projected 2D interactions on controllers instead of using graphical interfaces. Also, in the biological field, Abstractocyte [138] used a 2D panel to control the transition between highly-detailed 3D views and abstract 2D views. The essence of the combinations is that the views are connected in diverse ways : displays, contents, and interactions. The display connections include placing them separately [65], replacing each other [212], and colliding (3D representations on 2D visualizations [217] and 2D on 3D [98, 214]). Contents are connected via the dataset and visualizations, while interactions are connected either via controlling 3D environments with 2D representations [65] or controlling 2D visualizations with 3D objects [56].

In our scenario, the 3D representations provide the basic information, and 2D abstract visualizations present additional temporal data as a record. The interactions with two views are closely connected, e. g., the encoded colors, cell selections, and generation switching (see Figure 1.4). We explain the details in chapitre 5.

1.3 MACHINE LEARNING

Though visualization systems can solve the first two challenges mentioned in section 1, the third challenge requires additional techniques. For biologists, assigning cells is time-consuming, especially for a large embryo with more than 200 cells. One reason is that traversing every cell within a big embryo takes time. A potential approach to save time is adopting machine learning algorithms to provide basic assignments for biologists to check. Other approaches include recognizing similar cells and predicting the quarters of an embryo. Machine learning (ML) works on algorithms to help biologists predict and solve the problem

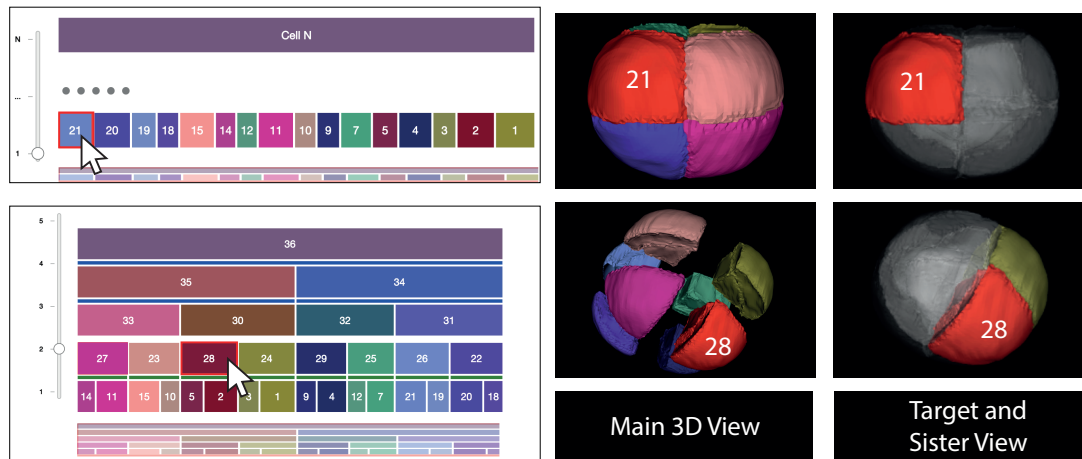


Figure 1.4 – The design of combining 3D and 2D representations. Selecting cells from either the abstract view or the 3D environment, the corresponding cell will be highlighted in the other view.

[136]. As we have mentioned in the previous two sections, biologists need to do the assignments manually based on 2D slices from the microscope to solve the cell lineage problem. This process could be quite time-consuming when embryos reach large sizes. Thus, it is essential to include machine learning models to provide basic predictions for biologists to start with. Since two adjacent cells are either sisters or non-sisters, we transferred the prediction problem into the binary classification problem. Multiple supervised algorithms could help, such as decision trees, support-vector-machine, single-layered / multilayered perceptrons, and statistical learning algorithms (e. g., naïve Bayes classifier) [101]. All these models have different pros and cons, and the choices of models highly depend on the problem itself. In the cell lineage case, there are lots of pairs to predict. Different models could have diverse results for each pair, so they should be included. We used visualizations to combine these different predictions and provide an overview with detailed views for biologists to have thorough information. We explained in detail about these in chapitre 6.

1.4 THESIS STATEMENT

In our thesis, we focus on the methods to help biologists to improve the efficiency of the cell lineage process. Thus, we assumed that the image processing part is completed through other means and worked on the segmented data instead of the raw image data from the microscope. In particular, we focus on de-

signing visualizations to present the model prediction results and the development history.

With these backgrounds, we investigate the following research questions :

- **RQ1** How to effectively check cells' 3D properties of an embryo for decision making?
- **RQ2** How to connect 3D representations with 2D abstract information?
- **RQ3** How to better visualize temporal information in 2D hierarchical structures?
- **RQ4** How to involve machine learning to help with the assignments and better visualize the predictions for biologists to refer to?

To conclude, our thesis contributes to : (1) an exploration of the 3D selection techniques in picking specific cells among the tightly-packed plant embryos, (2) a web-based tool, namely LineageD, designed for biologists to predict the cell assignments, visualize the cell development history, and interactively correct the predictions, (3) a new design of visualizations with corresponding interactions to provide an overview of multiple models' predictions, compare different models, and support customizing model combinations for reference.

1.5 THESIS OVERVIEW

This thesis consists of seven chapters. The summarized contents of each chapter are listed below :

- **Chapter 2 Background On Plant Cell Lineages** describes background information about the plant embryo and the cell lineage problem.
- **Chapter 3 Systematic Review** provides a thorough review of the related work of the visualizations in the biological field, 3D techniques for the occlusion problem, tree visualizations, and visualizations for machine learning.
- **Chapter 4 Exploration of 3D Selection Techniques for Plant Embryos** explores three different 3D selection techniques to pick cells from a tightly packed embryo. We conducted a study to simulate biologists' assigning decision-making process and evaluate these techniques with the general public.
- **Chapter 5 LineageD : Interactive Visual System for Plant Cell Lineage Assignments** presents an interactive web-based tool to help predict, visualize, and interactively adjust plant embryo cell lineages for biologists.

The system combines both 3D representations and 2D abstract representations and supports intuitive interactions for them along with the machine learning model. We also did a case study with biologists to evaluate whether the tool could help the assigning and inspire their insights.

- **Chapter 6 Comparative Visualization of Machine Learning Predictions to Improve Human-AI Teaming on the Example of Cell Lineage** presents the work of investigating multiple machine learning models' interaction and visualizations in helping with the cell lineage problem. We trained five ML models and used visualizations and interaction techniques to compare and understand these different models. We also conducted a study with six biologists to see whether and how the new design help with understanding and interacting with machine learning models.
- **Chapter 7 Discussion and Conclusion** discusses how our work could contribute to the visualization community and the potential future work which can inspire researchers. Finally, it provides an overview of our work.

BACKGROUND

In this chapter, we introduce an overview of the plant embryo we worked on, the lineage problem, and considerations to design.

2.1 PLANT EMBRYO DATASETS

For the research described in this thesis, we worked on the ovum cell of *Arabidopsis thaliana* plants (see Figure 2.1). In exploring the life of plants, biologists cannot investigate all species, and thus they choose model species to study. Such model species should generally be representative and easy to conduct experiments. *Arabidopsis thaliana* is one of the model species they are working on. After exploring this species, biologists could easily use the knowledge they got to study other species. Thus, we chose *Arabidopsis thaliana* plants and used all the embryo's hierarchical division histories and 3D information of that species. Also, there are multiple types of embryo growth, such as arithmetic and geometric increase. In an arithmetic increase, only one cell is dividing, while in a geometric increase, all cells are dividing. Geometric growth is common when the plant embryo is extremely young and small. Later, the growth can be a combination of both arithmetic and geometric growth [126]. We currently target the early stage of embryo development in consideration that young embryos are easier to find division patterns. Note that, initially, when people are talking about an embryo, they are indicating the **proper embryo**, but actually an embryo includes both the **proper embryo** (the upper part) and the **suspensor** (the lower part) as shown in Figure 2.2. The suspensor is the suspending part of an embryo,

and the function of these cells is to support the proper embryo.

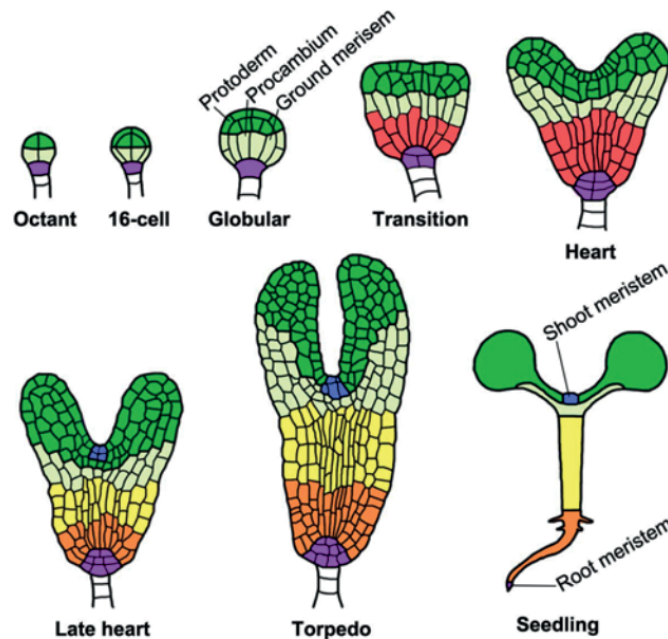


Figure 2.1 – The illustration [94] demonstrates the developing history of an Arabidopsis plant. Image © Springer Nature, used with permission.

2.1.1 Embryos

To work on the cell lineage problem, biologists work on different parts sequentially. They name the external parts as the *epidermis layer* and inner parts as the *internal layer*, while the upper part is called as *apical part* and lower part is *basal part*. All these parts make the proper embryo. Biologists separate the embryo into these parts because they are more familiar with the cell shapes in these different parts. Also, traditionally, biologists use a staining technique [196] for imaging entire plant embryos with the confocal laser microscope.

Besides these naming habits, biologists typically deal with the images they get from the microscope via existing tools to segment embryos. The cells' ids are set in ascending order by tools, starting from 1, and the name has no meaning for a specific cell. Each mesh file represents one embryo, including cells' names, every cell's triangles, and the positions of the vertex of meshes. Since tools like Avizo help biologists with the segmentation and automatically build meshes from labeled images, they could make some errors, especially in the naming part. For instance, a cell with a unique number could be nonexistent. Also, tools cannot distinguish the suspensor from the proper embryo, and files

would include all the cells in the slices. These required double-checking when parsing the datasets. In this thesis, we do not focus on the segmentation part because we were more interested in the interaction and visualization aspects of the problem. Thus, we assumed that the segmentation part is solved by existing tools, and we extracted the cells' properties from the mesh files.



Figure 2.2 – The sequential up-down slices from the same 16-cell plant embryo. Contrast enhanced to make the ID colors visible.

2.1.2 Cell Lineage

The plant embryo cell lineage problem is a general problem of interest in biology. In this thesis, we worked only with *Arabidopsis* embryos. The design process, however, could also be potentially applied to exploring the lineages of other species because the basic division process is similar: cells divide during the development process so that the embryo grows and develops. In plant biology, one cell typically divides into two cells, and that initial cell is called **parent cell** for the two generated cells, which are **children cells** or **daughter cells**. These two cells could also be called **sister cells** for each other. We used these terms in the whole manuscript.

During this process, however, not all cells necessarily divide in the same time period as the majority. This means that the lineage hierarchy tree is not strictly a binary tree and thus adds difficulties to the reconstruction. To better understand a species' division patterns, biologists need to work on exploring and rebuilding such embryo development history. This process is a bottom-up approach to building the hierarchy. Since the basic cells from the slices are named ascendingly, when naming a new parent cell, biologists would choose the number of the biggest number of the existing cells adding one. Thus, the hierarchy tree of a 32-cell embryo could look like that in Figure 2.3.

There are several general rules that biologists use when they establish the cell lineage, such as Hertwig's rule [81] and Errera's rule [18]. Hertwig's rule, also known as 'long axis rule', indicates that cells divide along their longest axis. Errera's rule describes that "the cell plate, at the time of its formation, adopts the geometry that a soap film would take under the same conditions" [18, 62]. These

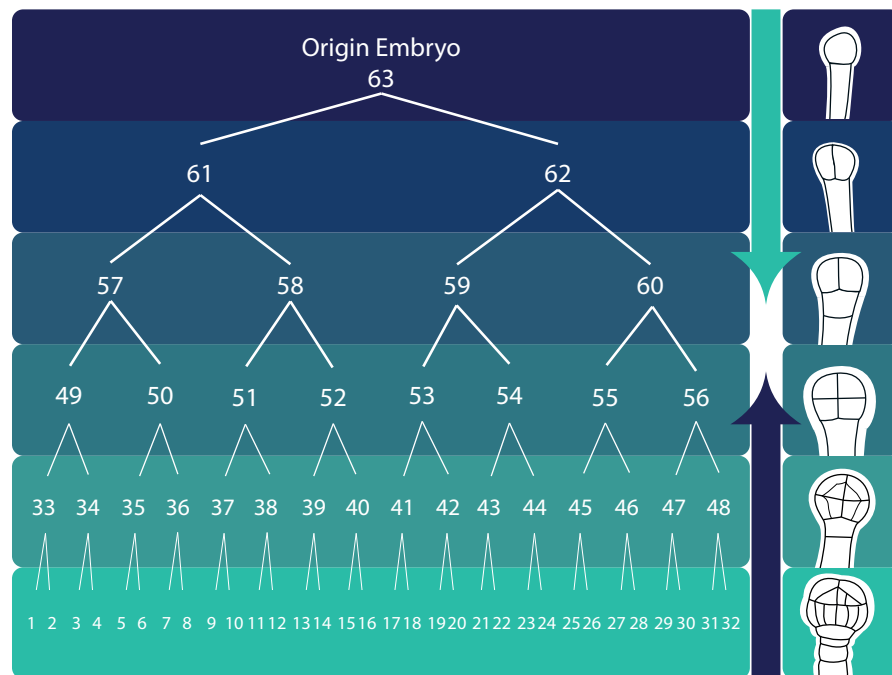


Figure 2.3 – The typical cell lineage of the plant embryo. The naming rules here follow their traditional habits. As an embryo divides, the hierarchy tree is built from the top down, while when biologists are reconstructing the tree, it is built from the bottom up.

rules, together with the biologists' experiences, provide general guidelines for them to check the cell shapes and volumes to make decisions.

2.2 CELL LINEAGE PROBLEM

2.2.1 Traditional workflow

For establishing the cell lineage, botanists will cut the slices of a sample and observe these slices from the microscope first, as shown in Figure 2.6. Based on our collaborators' description, with the slices they got from the microscope, they would import them into the existing tools, such as MorphoLibJ [112], Fiji ImageJ [162], MorphoGraphX [44], and OsiriX [157]. They use these tools to segment the embryo and, based on the segmented 3D meshes, they start to trace back the developing history of the embryo. They would observe and traverse all the cells in the current generation and observe their positions, shapes, and shared surfaces. With existing tools like TreeJ (see Figure 2.4), they need to single-click the target cell and the sister cell to make them a pair, and the tool then en-

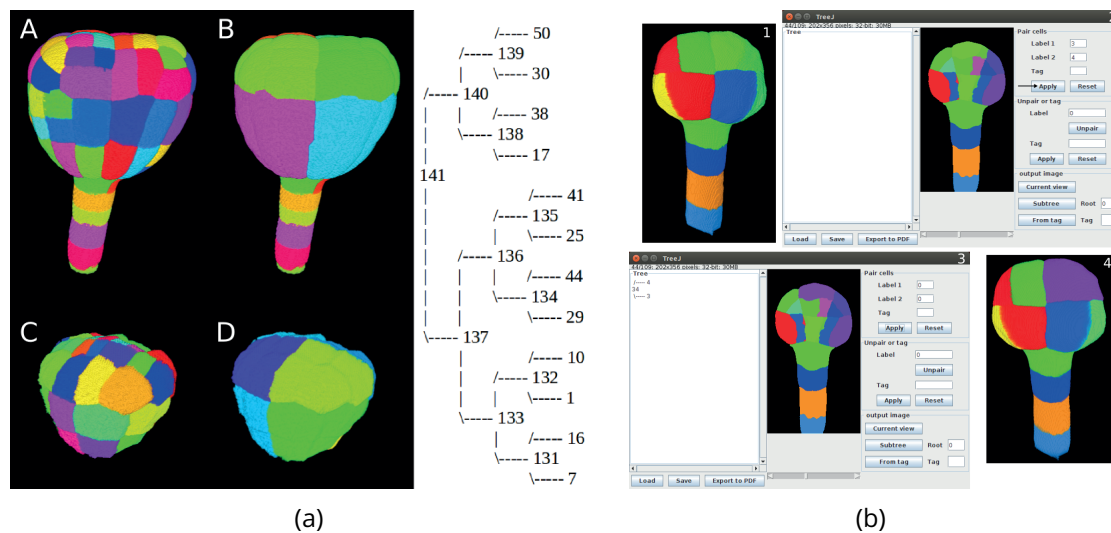


Figure 2.4 – The traditional system TreeJ (a plugin to Fiji ImageJ) that the experts traditionally use to specify the cell lineage : (a) screenshots of the interface. Users primarily interact in a 2D slice of the microscopy data, colored by the cell's assigned ID, to select a pair of cells to merge. The resulting tree is then represented in a textual form with numbers as cell labels, and it is no interactive. The 3D view (marked A and D) only serves as an additional view and is also not used for interaction, and (b) 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.

codes these two cells with the same color. Based on their knowledge and experience, they pick a most likely sister cell and mark these two cells as pairs and write down these pairs. Typically, tools do not support the hierarchy tree building, and biologists need to do it manually. The confidence they get after making such a decision varies among biologists. Still, they are typically confident in their choices, especially for small embryos. After assigning all the cells in a given generation, they would merge the proposed pairs and repeat the process for the new-merged cells in the temporally previous generation until only one cell remains. Though the cell shapes and positions could only be slightly different between generations, they do not affect the decision-making of biologists. The cell lineage process is basically a repetition of seeking sister cells. Normally, biologists record the ongoing hierarchy history in the text format without any interactions with the tools as in Figure 2.5, or they would write down the most likely pairs in the tree format on paper (see Figure 2.7). The text format is actually an array. The number in each position indicates the parent cell's id. For example, if the first number in an array is 30, then it means that cell 30 is the parent cell

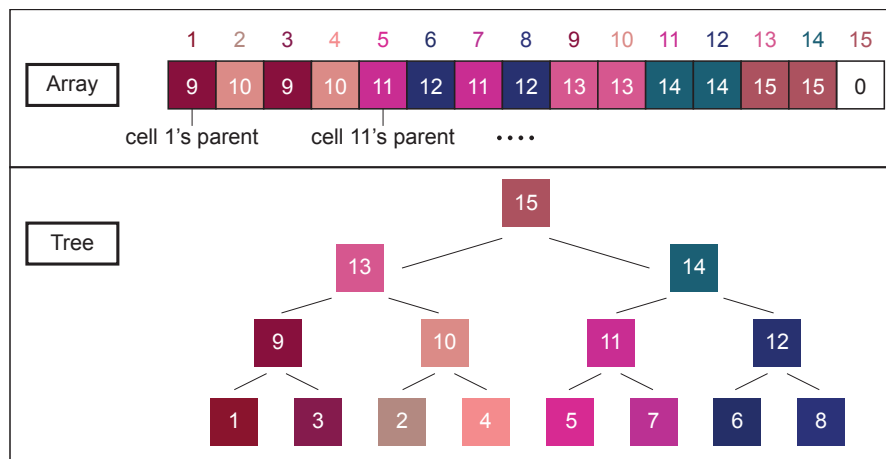


Figure 2.5 – The illustration of the recorded format. The upper part is the array biologists would save in a text file, and below is the corresponding tree in 2D.

of cell 1. These manual characters of the data recording make it hard to connect cells in the hierarchy tree with 3D objects. During this process, it is likely that different biologists build diverse trees. The only way to be certain about the hierarchy tree is to observe the time lapses of plant embryos [73] though the image quality is not good. However, it is not easy because it happened inside a small seed.

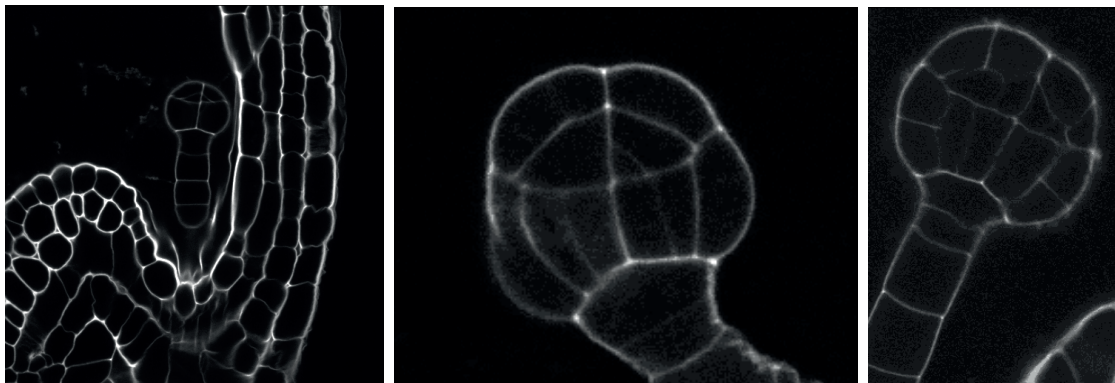


Figure 2.6 – Embryos slices got from the microscope. From left to right, embryos consist of 16, 32, and 64 cells respectively. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.

The current usage tools (e. g., OsiriX [157], Fiji ImageJ [162], and the TreeJ plugin for Fiji ImageJ) help the biologists with the cell checking and history rebuilding. They, however, were mainly designed for checking embryo shapes and segmenting embryos and mostly relied on 2D images for the biologists to make deci-

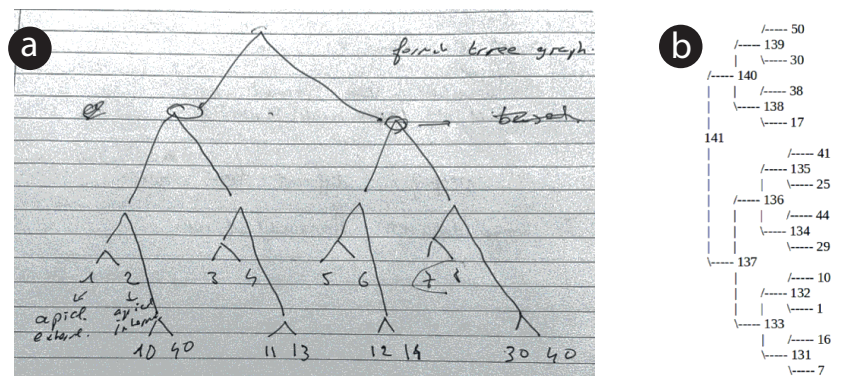


Figure 2.7 – Traditional trees to record the embryos’ developing history : (a) the hierarchy tree biologists wrote on the paper, (b) the tree used in TreeJ. Image © Elise Laruelle, Philippe Andrey, Jean-Christophe Palauqui, and Alain Trubuil, used with permission.

sions. For example, with Fiji ImageJ, biologists input slices, and the tool segments the embryo automatically with different colors (see Figure 2.8). After checking all the slices being segmented correctly, the biologists reconstruct the embryo model in mind to build the hierarchy tree. Though TreeJ was designed to solve the cell lineage problem, it also lacks 3D shapes for biologists to refer to. The history could only be built in a 2D format as the Figure 2.7 shows, which means that the hierarchy tree cannot be explored interactively. Even for those tools (e. g., Avizo) which provide 3D environments, it is difficult to examine the inner structure, including the shapes of inside cells, because plant cells (like the cells in most living organisms) are densely packed together in an embryo. Moreover, this manual assigning workflow is time-consuming and error-prone, especially for large embryos with hundreds of cells. In addition, these tools require users to download and install the implementations and, thus, have limited accessibility. Thus, developing an elegant and efficient tool for solving the cell lineage problem is challenging, urgent, and essential.

2.3 CONSIDERATIONS

With all these challenges and backgrounds, next we summarize our considerations regarding the specific biological scenario for the design.

The first consideration is that **the embryo has different parts**, e. g., apical and basal parts. Biologists have the knowledge of how an embryo could be divided into those parts, and this information largely helps with and influences their assigning decisions. That means biologists can build the hierarchy tree in

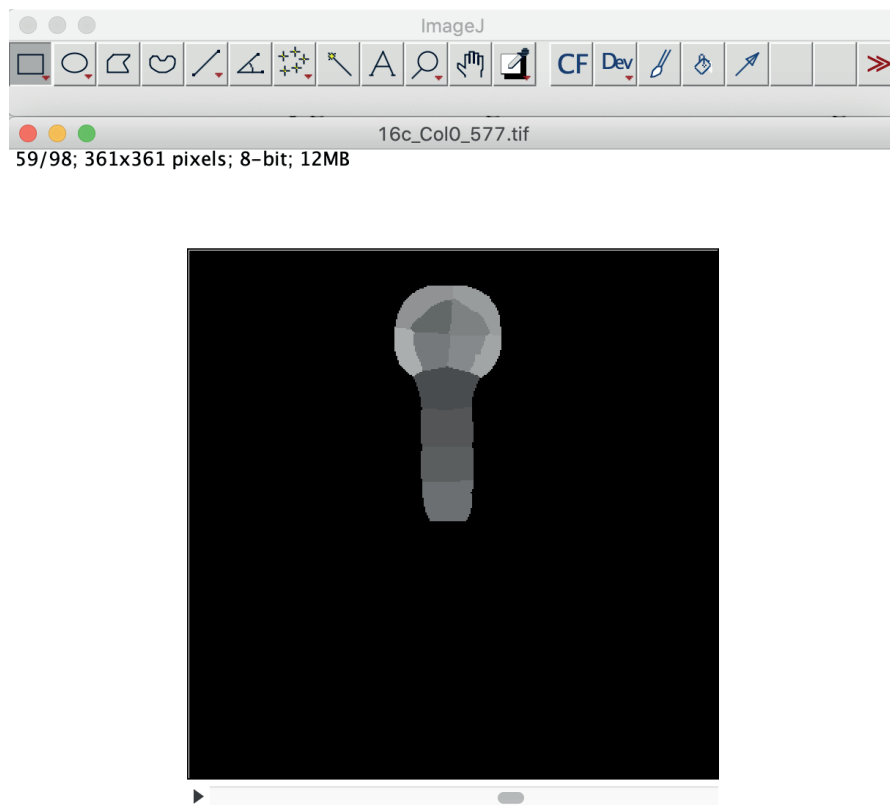


Figure 2.8 – The screenshot of Fiji ImageJ [162] deals with the slices of a 16-cell embryo. The lower slider bar enables users to target a specific picture among the whole stack of slices. The figure here illustrates the segmented embryo.

both a bottom-up and a top-down approach. In the system, we should support such grouping for biologists so that when machine learning algorithms are predicting the assignments, they can also provide better predictions. We achieve the grouping function and explain the details in section 5.3.

The second consideration is that **the raw datasets usually include the main embryos and the suspensors extracted from the slices**. We make use of this fact and help biologists detect the suspensors and hide them from the main embryo in chapitre 4. In the meantime, we support biologists manually marking these suspensors.

The third consideration is that **biologists have their preferences and habits**. For example, they used to encode random colors to different cells so that they could easily distinguish two adjacent cells even though these colors have no additional meanings. Also, they have their customized approach to doing the lineage, and we take this situation into consideration in chapitre 5.

The forth consideration is that the system should work for **general lineage problems** other than just plant embryo datasets. Biologists are also interested in the lineage of other tissues or organs. That is to say, the visualization and interaction design need to meet other datasets. We considered this and discussed our detailed visualization in chapitre 5 and chapitre 6. We also discussed the potential extension in chapitre 7.

SYSTEMATIC REVIEW

In this chapter, we present work that is related to our own research. Based on the tasks-by-data-types taxonomy proposed by Shneiderman [173], in our thesis, we include three-dimensional data (spatial information of plant embryos), temporal data (the approximate division periods), and tree data (the relationships between parent cells and children cells). Thus, the major challenge for us is to connect all these data types in a visualization system and to provide interaction techniques that allow users to smoothly manipulate them to finish the task. We, therefore, investigate different visual systems in biology. Then, we review the literature on interactively linking the 2D and 3D representations. Finally, for a better understanding of how machine learning algorithms can assist users solving the cell lineage problem, we survey related research projects about predicting object assignments in similar setups.

3.1 VISUAL SYSTEM IN BIOLOGY

There are numerous problems to work on in the biological field, ranging from species-related to multi-scale challenges, which can be somehow solved by visualization techniques. It is because these problems are usually abstract and complicated, where visualizations can assist in explorations. For instance, visualization has been proven to help biologists better understand life, and the process of organs [54]. Since problems in biology are usually specific, such as analyzing the gene sets [105], protein interaction [33], and biomolecular interaction networks [170, 25], researchers should create tools particularly targeting at

the problem. Thus, many different visual systems were created to solve those diverse problems. Among them, we are interested in two types of systems. One is to use enhanced visualization and interaction with 3D biological models to solve the problem. The other one is to visualize the biological data in 2D representations for better analysis. We are interested in these two types because, in the scenario of the cell lineage, biologists need 3D plant embryos to refer to and potentially require 2D diagrams for further hierarchy analysis.

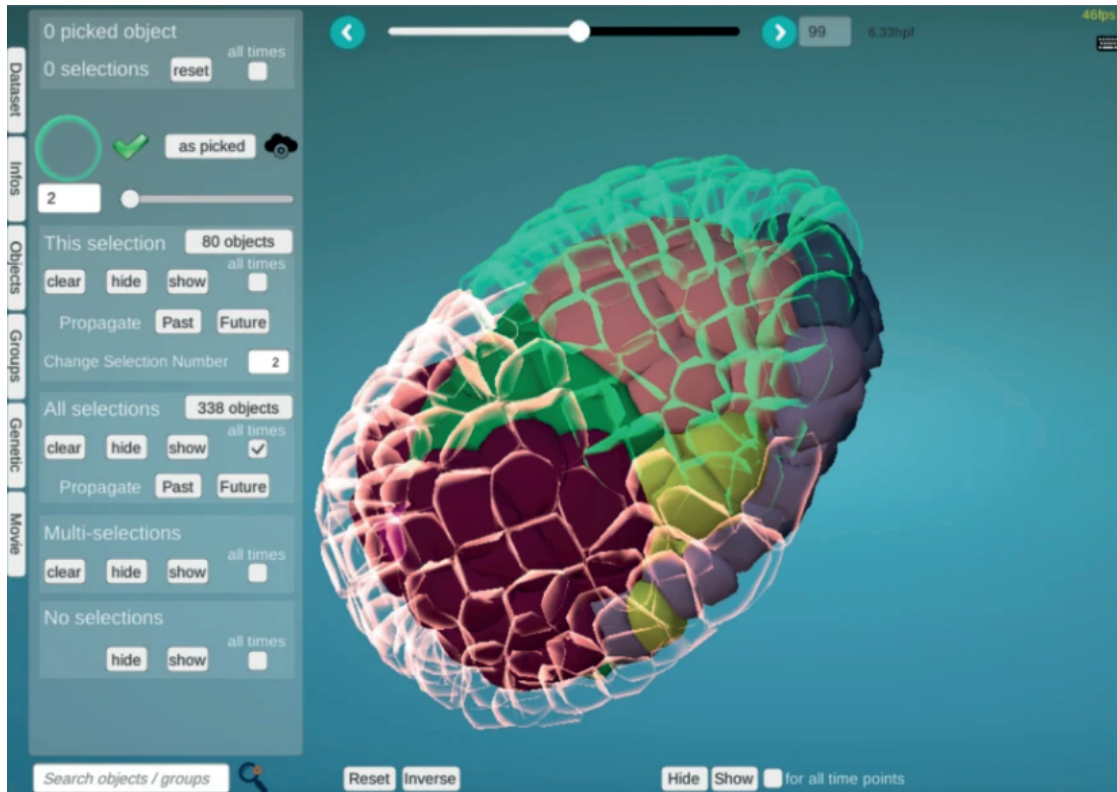


Figure 3.1 – The screenshot of MorphoNet [110]. Image permissions by Leggio et al. [110], [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/).

For helping with visualizing the 3D models, one interesting example is MorphoNet [110] which enables biologists to observe the multi-scale biological data (see Figure 3.1). It can visualize cells and organs. Other than that, it also allows biologists to select specific cells among the whole embryo and make changes to them, such as changing colors or hiding them. MorphoNet was developed in Unity 3D and embedded in a website. This installation-free system also inspired us to support an online tool with interactive 3D environments. The tool, however, has constraints in several ways. Firstly, even though it can make cells transparent, the cell occlusion problem is still challenging. It is difficult to get ac-

cess to the inner cells. Secondly, the 2D panel was designed to control the 3D objects but lacks sufficient summarized information for complicated decision-making. Lastly, the compatibility of the website with Unity is not that smooth, and the interaction has some delays for big embryos. Similarly, 3D-Cardiomics developed an interactive 3D heart for biologists to examine the gene and compare between different heart pieces [140]. It used the expand function to move parts separately to solve the occlusion problem. Unlike these visualization systems, our purpose is to see not only the 3D structures of plant embryos but also to make adjustments to these datasets and produce a dynamic hierarchy tree.

Besides this work, we also investigated other literature on 3D visualizations from different aspects. For example, for better rendering, Mindek et al. developed the visualization pipeline for cell science which enables to process and render the volumetric data sequentially and in parallel so that biologists could get the visualizations in different degrees of quality [134]. Researchers also investigated visualization approaches for other datasets such as multiscale molecular [131]. Such systems to analyze the molecular usually contain diverse techniques, like visual guide [32], distortion [59], extracting abstraction structures [68], and overview and details [38]. These strategies are also important for our system to support embryo examination.

In addition to the 3D-related techniques, these strategies usually require additional abstract data visualizations. For example, based on MorphoNet [110], researchers added a genome browser to visualize the gene order [40]. Also the visual representations can also be organized in the 3D environment, such as Arena3D [148]. It used the staggered layers in the 3D environment to visualize networks. Each layer contains diverse groups of information. Unlike traditional 2D images, the design uses a large space and avoids overlapping issues. Because we worked on the plant cell lineage problem, we specifically looked into the hierarchy visualizations. Wang et al. [204] summarized two categories of visualization techniques to visualize hierarchies : node-link and space-filling diagrams. Node-link diagrams focus more on the relationships between nodes [187]. In contrast, the space-filling diagrams can emphasize the relative widths or areas of nodes. Thus, they can provide a clear visual guide to the hierarchy. The combination of these two types, such as Elastic Hierarchies [218], saves space and emphasizes the inner relationships (see Figure 3.2). Researchers also investigated the color modes for the tree structures [193].

Plant embryos' division processes contain the hierarchical structure and the temporal information. As we have stated before, biologists can roughly group the cells with similar division time into one level. It means that the hierarchy has inherent temporal data, though the information is time periods instead of precise timestamps. Biologists, thus, usually create customized hierarchical trees

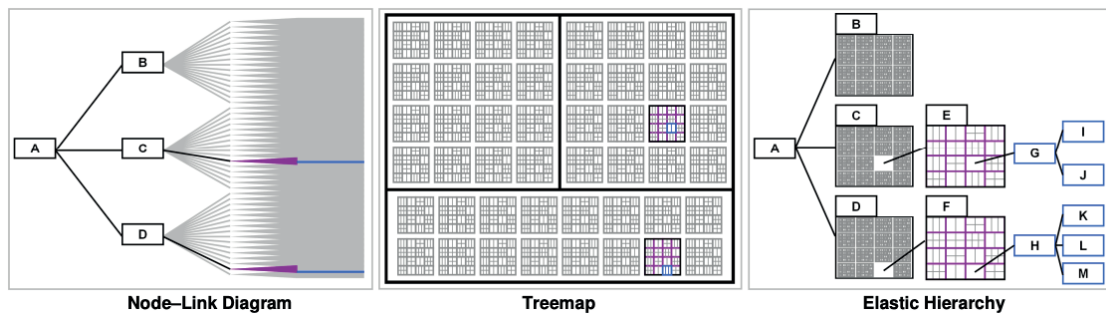


Figure 3.2 – The illustration of typical types of trees. The left image is an example of a node-link diagram. The middle image is a treemap, a typical instance of space-filling diagrams. The right one is the Elastic Hierarchy developed by Zhao et al. [218]. Image © 2005 IEEE.

with generation information emphasized (see Figure 2.7).

These projects mainly focused on better and more efficient rendering of biological objects and helped with the analysis. In some cases, however, biologists' decision-making should base on 3D biological models. For the cell lineage problem, biologists need to analyze the embryo cells' inter-relationships and build the hierarchy tree. These decisions are made after observing the cells' properties in the 3D environment. Thus, in the following section, we investigated the combination of design with 3D visualization and 2D abstract representations, which are not necessarily in the biology field.

3.2 HYBRID SYSTEM COMBINING 3D AND 2D REPRESENTATIONS

As shown in the previous section, 3D visualization normally presents necessary information about specific 3D objects, such as the embryonic data [110]. For controlling such complicated systems, designers usually add a 2D control panel with other 2D representations to show abstract or additional information. These two parts (the control panel and 2D representations) can be actually merged together to support users in better understanding the dataset and making decisions. In this way, besides getting additional knowledge, people can also control the 3D visualizations from such 2D representations. Other than separating two views, we can also add 3D models directly to the 2D diagrams or the other way around. Then within one view, we can visualize extra attribute values intuitively as well.

In this thesis, we are interested in the design of such combinations of 3D and

2D representations where 3D and 2D views are not simply control panels for the other view but also represent valuable data. We thus divided all the systems into three categories : 1. Two views are mainly adding/highlighting information for the other view without interaction connection ; 2. The interaction connection between two views is one-directional ; 3. The interaction connection between two views is bi-directional. The first category includes cases where one view encodes additional datasets for the other, or the 2D view is an abstract view of 3D objects. In this scenario, two views are mainly connected via the datasets or visual linking. The last two categories include cases where two views are interactively connected. These three categories actually demonstrate two questions : 1. Why should these tools combine 2D and 3D representations? 2. How could they let people understand and intuitively interact with multiple views? We will explain in detail as follows.

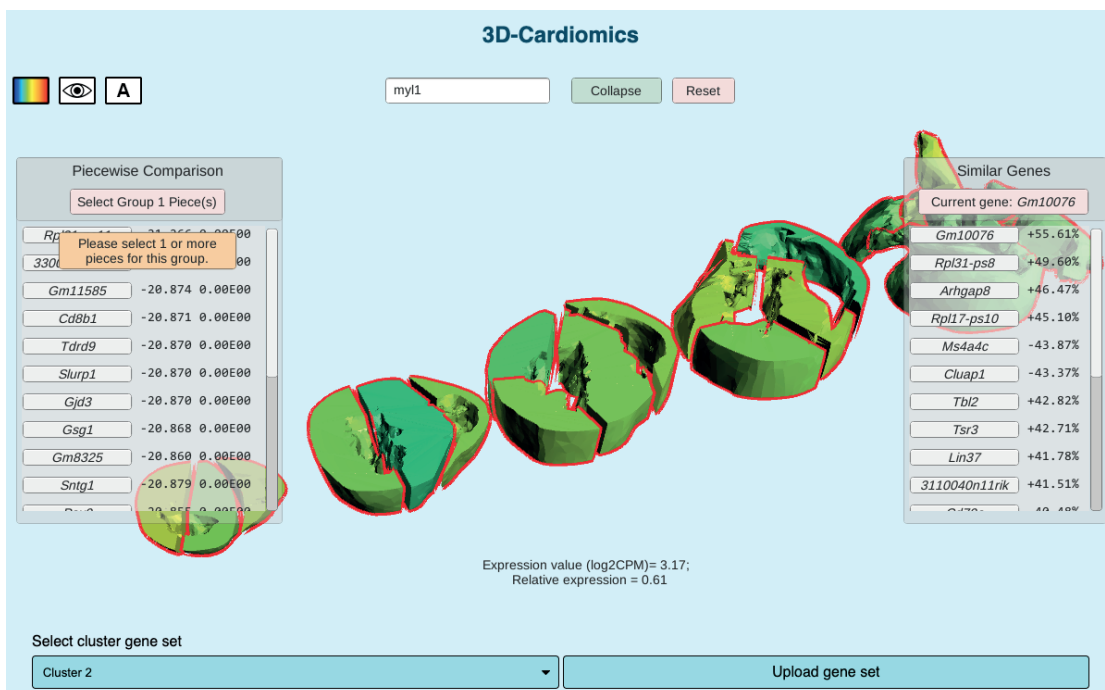


Figure 3.3 – The screenshot of 3D-Cardiomics [140]. The heart model is expanded in the view, and two 2D panels add information to the 3D representations. Users can hover over specific parts to see the details and use the buttons on the panel to target a specific gene.

The purpose of combining the 3D with 2D representations is mainly to use one view to enhance the expressiveness of the other view via adding additional information or highlighting some part of the data. In this case, 2D visualization can help present the extracted data from the 3D models, and 3D representa-

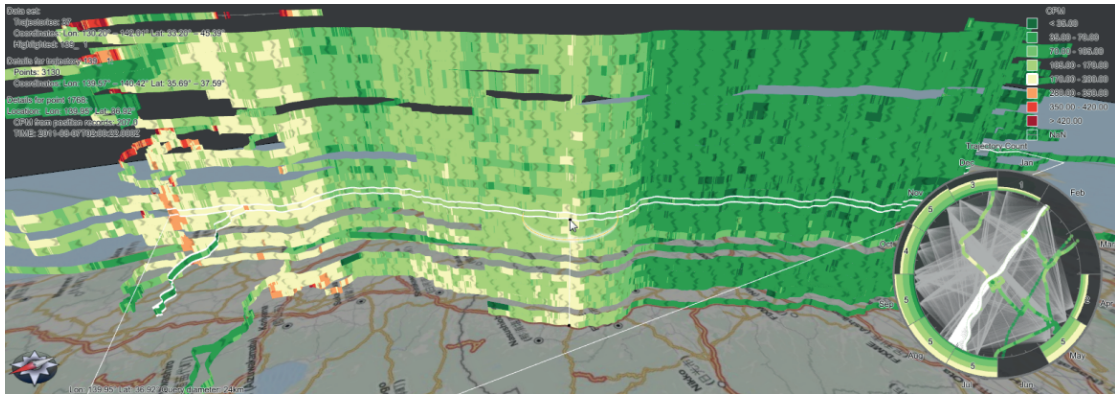


Figure 3.4 – The illustration of the hybrid 2D and 3D to visualize trajectory attribute data [195]. Image © 2012 IEEE.

tions can encode other datasets for 2D diagrams. For example, Semmo et al. [168] transformed the geo-virtual city into an abstract diagram and combined these two seamlessly so that people can effectively observe the city. Similarly, in the paper [106], researchers use the abstracted 2D view to represent the network traffic structure for users to find patterns efficiently. The forming of 2D diagrams depends on the 3D view, and no additional information is included in these cases. Also, with 2D abstract representations, a system could add additional information to help with the expressiveness. A typical example is 3D-Cardiomics [140]. With this tool, when targeting a specific gene, all the similar genes will be listed in a 2D panel with additional information like the gene names and the false discovery rate (FDR). Meanwhile, targeting the gene name in the 2D panel, the 3D models will be highlighted accordingly (see Figure 3.3). This tool is interesting because the researchers enable to make interaction of 2D and 3D representations closely connected, which largely increases the interfaces' usability. For designing 3D visualizations on 2D diagrams to represent extra data, we have examples where researchers build 3D visualizations on a 2D map [195] as the Figure 3.4 shows to visualize the trajectory. In their design, the temporal data is encoded with the ordering of bands, and the speed is represented by different colors. It provides an intuitive overview with a dynamic query system to support the exploration. Another example is the vessel wall visualizations from Meuschke et al. [130]. They used a flatted 2D map to represent the aneurysm surface with color indicating the wall thickness and a 3D bar chart to show or emphasize one attribute, such as the deformation. Different from the 2D diagrams on 3D visualizations, the 2D representation here serves as a background, and the close placement of 2D and 3D visualizations add to the sense of engagement.

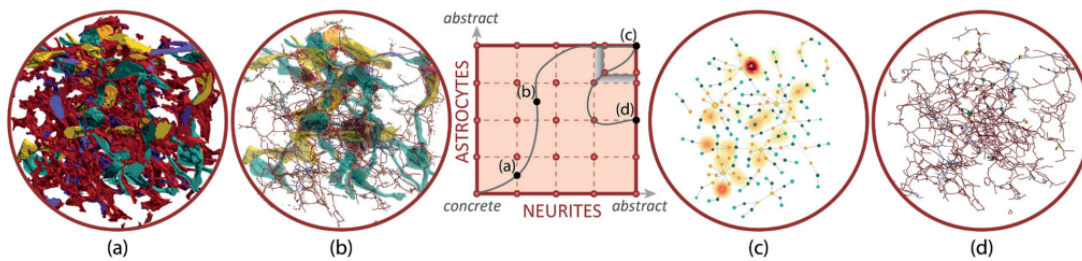


Figure 3.5 - The illustration of Abstractocyte [138]. From left to right, the neurons and astrocytes are abstracted in different levels. Image © 2017 IEEE.

After exploring the reasons why previous tools chose to combine 3D and 2D representations, we investigated how they are linked together. 2D and 3D representations are both usually important in decision-making and task completion. For systems with two views that are able to finish the task independently, i. e., the datasets are shared in 3D and 2D, researchers are likely to put these two views in the same place, so people can link them visually. For example, in the biological field, Abstractocyte enables biologists to control the transition between the 3D meshes and abstract node-linked diagrams of neurites [138] (see Figure 3.5). The 2D and 3D visualizations replace each other in the main view and are connected via the datasets. Furthermore, researchers also explore electronic transitions of the molecules in two dimensions for better analysis [125]. When both views can only collaborate together to finish the tasks, researchers tend to put these two parts side by side, e. g., [138, 159]. In these cases, two views are linked because operations in one view can get feedback from both views. For instance, CeLaVi is one of the most related works to visualize the hierarchy tree combined with 3D simulated molecules [159]. The hierarchy tree is visualized on the left with a node-linked vertical diagram. Each node represents a cell in a specific generation. The generation information is displayed below the tree. On the right, every cell is simulated with one sphere in 3D, and its position is the center of the cell. Two views are interactively connected as shown in Figure 3.6. Interacting with one of the views will highlight nodes/cells in the other view. Though biologists could easily target the branches and correlate them to the cells in the 3D environment on the right, the hierarchical tree is static and has no predictions.

Our co-supervised master's student Ebrar A. D. Santos worked on exploring the design space of such combinations and published a poster about it [160]. We went through the papers of 2012-2022 from IEEE Vis, EuroVis, and TVCG, and extracted 97 related work. Then, we came up with five dimensions to group these designs, attempting to answer questions of why, where, and how to combine as illustrated in Figure 3.7. The semantic relationship demonstrates "why" to compare these two kinds of visualizations. The display medium shows "where" to

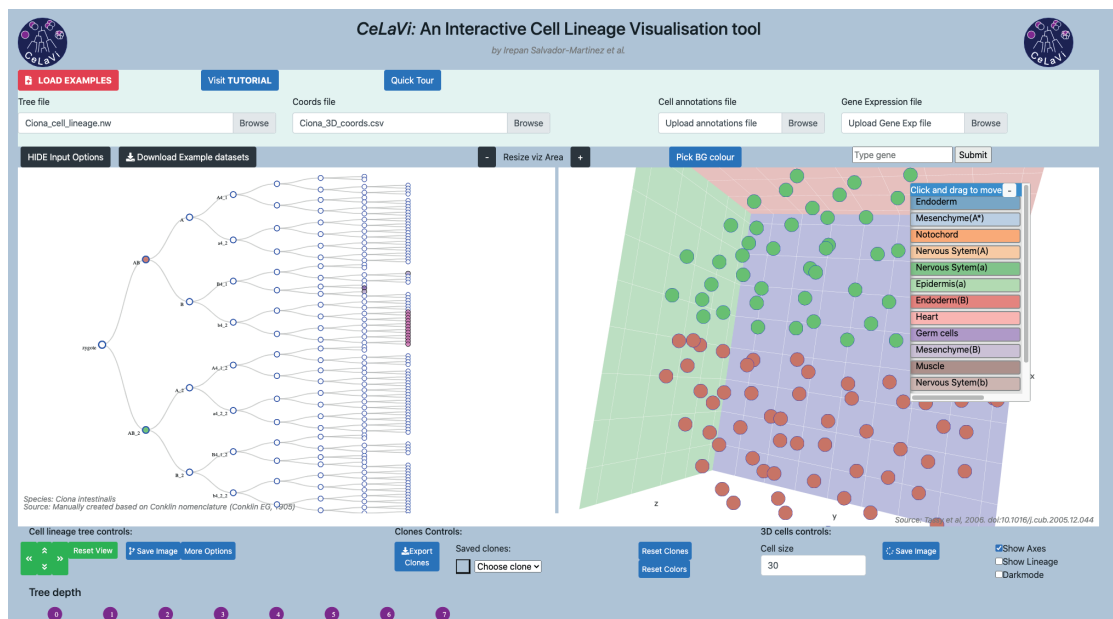


Figure 3.6 – The screenshot of CeLaVi [159] using the default Ciona dataset. The hierarchy tree is placed on the left side, while the 3D view of cells is on the right side.

use these two views, and lastly, the layout, linking relationship, and control panel are designed to illustrate the "how" question. In our thesis, the 3D embryo shapes and positions are important for biologists to make a decision on which two cells can potentially be sister cells. At the same time, the machine learning predictions are visualized in a 2D tree allowing people to check. Thus, based on the literature, we chose to place these two parts side by side. Also, because the datasets of our 3D models and the hierarchy tree are different, we decided to interactively connect these two views to enable controlling the 3D views with the 2D abstract tree and support the other way around.

3.3 VISUALIZATIONS FOR MACHINE LEARNING

Machine Learning (ML) has become increasingly important in solving diverse problems. In our case of plant cell lineage, ML can also assist in predicting the lineage so that biologists do not have to assign every cell with a proper sister cell totally by themselves. However, people may not fully understand the direct output of ML predictions, and the lack of knowledge of ML models themselves could also discourage people from trusting models. Visualization, in this case, can help people to generate, evaluate, and understand various ML models [158].

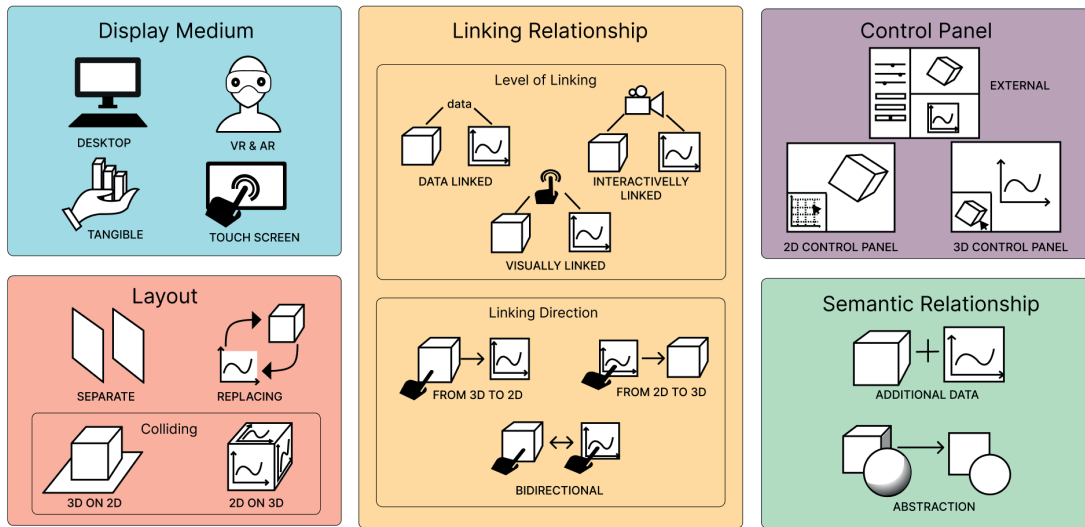


Figure 3.7 – The design space we came up of combining the 3D and abstract 2D representations [160].

For example, in our case, adding visualization to explain the prediction results can assist biologists in making better decisions. In the work [119], the authors specified the literature on visualization for ML into three categories : understanding, diagnosis, and refining the ML models. To combine visualizations with the machine learning in the workflow, explAIner [176] proposed the pipeline as in Figure 3.8. With this, researchers can improve the interactions between users and ML models. Such combinations of visualizations and ML can be used in diverse fields, such as biology [15] and medicine and health care [200]. In this section, we will briefly discuss the related work in the following directions to see how visualizations can assist ML, especially in the results and how our system can learn from these examples.

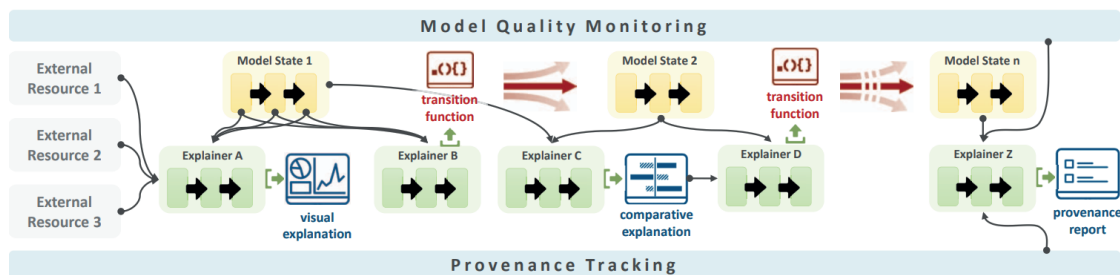


Figure 3.8 – The workflow proposed in explAIner [176]. As we can see from the figure, there are multiple explainers being applicable in the process, and visual explanations can be added in between. Image © 2019 IEEE.

3.3.1 Understanding

Visualizations help enhance users' understanding of machine learning in various ways, including the model generation process and model performance. The former part mainly aims to create an optimal model, while the latter focuses on the predicted results. In terms of the model generation process, Wongsu-phasawat et al. developed a system to visualize Tensorflow's architecture [208]. As for the model performance, ActiVis designs visualizations to unify instance and subset-based neuron activation and classification results and support easy comparisons and mistakes reasoning [89]. Another example is Seq2Seq-Vis [182], which visually represents the seq2seq model predictions and enables viewers to find the potential issues.

For our cell lineage prediction problem, we transformed it into the classification problem, and many visualizations were created to understand these classifiers better. We investigated interactive analysis visual approaches in the classification field. For example, based on the clustered branch organization [192], Paiva et al. proposed a visual classification approach to assist in analyzing the similarities between classification results [147]. These examples indicate that the visualizations need to be designed specifically for the requirement, and in our case, we should combine the visualizations for ML results with the hierarchy tree.

3.3.2 Machine learning for cell lineage

Machine learning can help the biology field in various ways, such as cell segmentation and cell clustering [202]. For the cell lineage problem, which is a top-down building approach to the hierarchical tree, researchers investigated different ML models to predict cells' division and their potential fates. For example, researchers used ML algorithms combined with cell lineage tracking to disclose the coordination in the organ regeneration with the sample of superficial neuro-masts [201]. The model they created can help identify the features for determining the cell fates. Similarly, in this work of McDole [127], the authors developed a supervised ML model to detect the dividing activities. As for tracking back the development history of an embryo which is a bottom-up hierarchy-building process, we transformed the cell lineage problem into a classification problem, so ML models need to decide, for each cell, which are potential sister cells. Other researchers treated the cell lineage prediction problem differently. For example, Louveaux et al. explored the methods to predict the cell division plane [120] following the biological rule for the division [154]. The rule is that for plant cells that are going to divide, the cell walls will be built in the orientation along maximal tension.

As we can see from all these examples, it is essential to use visualizations in explaining machine learning to people in terms of the ML prediction process and output so that people, even without enough knowledge about the dataset, can get an overview of the information and better make decisions. For people who have limited knowledge of ML, we may need to focus more on providing them with understandable prediction results and supportive interaction techniques to interact with these results, especially for complicated problems where ML has limited ability due to the limited training datasets. As in our scenario, based on biologists' feedback, they care more about the results of model prediction than the models themselves. We then mainly used visualizations to summarize the prediction results from multiple models. We also supported Human-AI teaming to enable biologists to control the final decision-making.

3.4 SUMMARY

To conclude, we have gone through previous work in diverse visual systems in biology, visualizations design combining 2D and 3D representations, and machine learning models for the cell lineage problem. With an overview of them, though numerous research work is done in these three directions, none of these projects systematically focus on combining them to provide an interactive tool. We, thus, are targeting to support biologists with visual systems to better deal with the cell lineage problem.

EXPLORATION OF 3D SELECTION TECHNIQUES FOR PLANT EMBRYOS

4

Our first project in my thesis is to try to answer **RQ1** : How to effectively examine the 3D properties of embryo cells for decision making? The cells in plant embryos are densely packed with virtually no space between them, which means that traditional interactive exploration techniques do not apply in this scenario. We, thus, investigated three established alternative selection techniques, namely 1D List Selection, 2D Cross Section Selection, and 3D Explosion Selection as shown in Figure 4.1. In our pilot study, we found that the Cross Section Selection approach is hard for biologists to control. Thus, excluding this 2D selection technique, we investigated the other two methods in more detail for their suitability for densely packed cell environments.

In this chapter, we describe the controlled user study in which we investigated and compared two selection techniques in the scenario of cell lineage. We also divide the cell interaction into two parts : discovery and traversal. We compare List Selection, 3D Explosion Selection, and a combination of both techniques in these two aspects. Our results indicate that the combination was most preferred. List selection has advantages for traversing cells, while we did not find differences for surface cells. Our participants appreciated the combination because it supports discovering 3D objects with the 3D explosion technique while using the lists to traverse 3D cells.

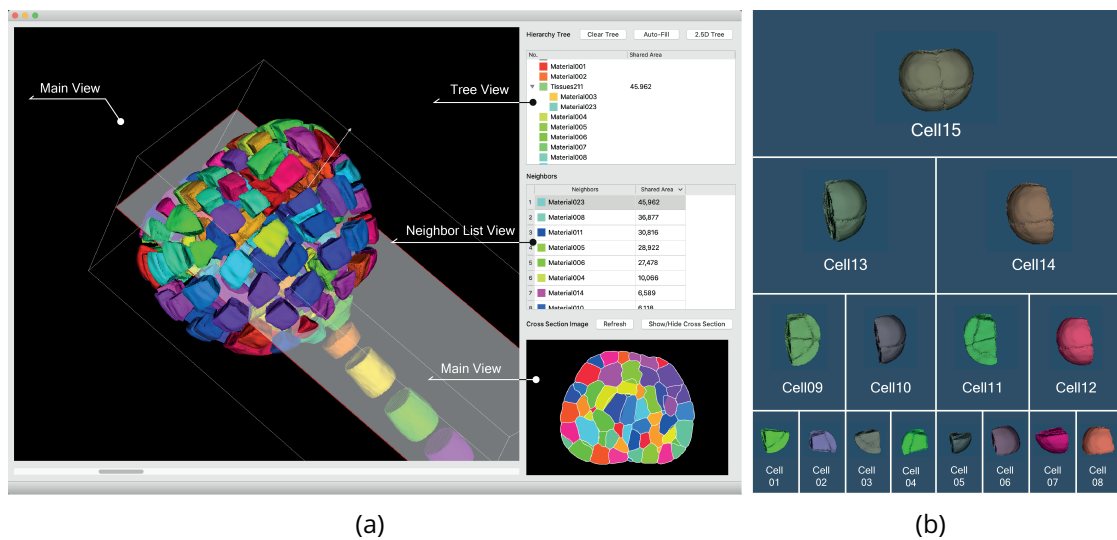


Figure 4.1 – Our original system : (a) the main interface with three selection techniques and (b) the hierarchy tree for 8-cell embryo. The tree contains the 3D interactive models of corresponding cells.

4.1 INTRODUCTION

Selection as an interaction technique is fundamental for data analysis and visualization [207]. In 3D space, selection requires users to find and point out one or more 3D objects (or subspaces), and a sizable amount of research has been carried out on different 3D selection techniques [6, 13, 17, 5, 59]. Among them, ray-casting [155, 135, 5] and ray-pointing [10, 5, 145] for object selection as well as lasso techniques [216, 215] for point clouds or volumetric data are common techniques. These existing techniques come to a limit, however, when data objects are tightly packed and no space exists whatsoever between adjacent data objects so that internal structures are inaccessible.

Such selection problems in dense environments arise in many scientific domains where researchers deal with data that originates from sampling properties in 3D space. We are motivated, in particular, by botany where cells are densely packed in captured data, virtually without any room between them and half or more of them being enclosed [59] such as in a confocal microscopy dataset of a plant embryo's cellular structure (Figure 4.2). With such data, botanists explore the development of plant embryos based on their cellular structure. Using a segmented dataset, they reconstruct the history of the embryo's cellular development [142]. This process requires them to select each cell, one by one, ex-

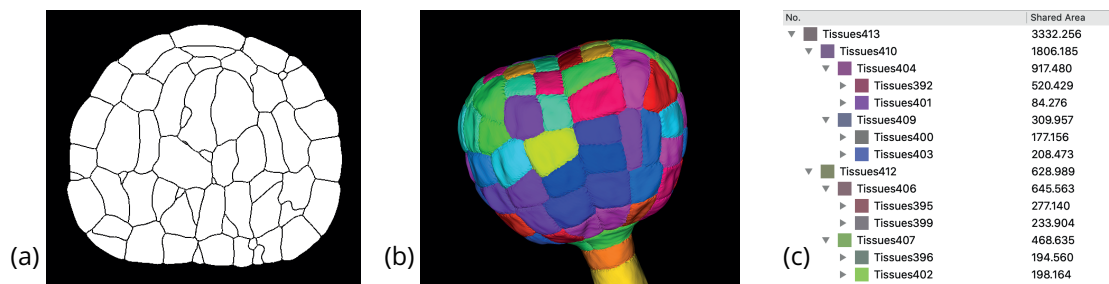


Figure 4.2 – Plant embryo dataset with 201 cells (87 “occluded” cells) : (a) a segmented cross section from confocal microscopy, (b) the 3D model, and (c) a part of the desired cell lineage tree—the botanists’ goal to be able to study the embryo’ development. The shared area of any specific cell indicate the shared area between itself with the parent cell. If a cell has no parent cell, like the cell 413 in this case, the shared area is the surface area.

mine its immediate neighborhood, select each potential candidate in the neighborhood to check the shared surface and relative position, and then decide on a likely sister cell that originated from the same parent as the target cell. This process is continued for all cells, and potentially previous assignments are revised if needed. The cells are naturally tightly packed, so we ask the question of how to effectively select 3D objects in such spaces, in particular for realistic datasets with 200 cells or more.

Currently, botanists use several tools to study cell division, but none of them provides efficient selection interaction techniques for 3D objects in dense packed environments; they are unable, e. g., to filter cells in a view for better selecting or to support marking based on 3D data rather than just 2D (TIFF) images. Researchers currently manually mark the cells, starting by targeting cells for which it is easiest to find the respective sisters. From the set of 2D images, they then identify all neighbors and examine their shapes and that of the surface the two cells share. Based on their past experience, they then decide on the most likely sister for the target cell.

We thus worked with them to understand their needs, to investigate intuitive selection techniques, and to support them to interactively derive the cell division tree. To better investigate the effectiveness of the needed selection techniques in this specific dense packed data scenario, we divided the cell selection into two parts : discovery and traversal. Discovery means to find a specific cell to assign within the whole embryo, while traversal refers to picking a specific range of cells in order. With this definition, we can describe the cell division process as repeatedly discovering target cells and traversing their complete set of neighbors. We then evaluated three selection techniques : list selection (*List*), explosion selec-

tion (*Explosion*), and a combination of both (*Combination*). *List* provides traditional lists to indirectly select cells, while *Explosion* displays an explosion view of the embryo and allows to directly select cells. *Combination* supports both techniques in one interface. We were also interested in how efficient these techniques are when selecting cells in different positions (on the surface and being enclosed). We thus designed an experiment to compare the techniques and the two cell positions. We measured task completion times, assignment accuracy and clicking ratios (clicking times for each neighbor). We also gathered subjective feedback from our participants such as their interaction strategies and preference.

Our results show most participants favored the *Combination* technique : they preferred to control the cell distance, often discovering targets in the 3D view, and then using the lists to traverse the neighbors. *List* performed better than *Explosion* when assigning occluded cells, while there was no clear performance difference between these two techniques for the cells on the surface. With our results on the techniques' performance and people's feedback about interaction, we derived suggestions for future 3D selection technique design and discuss current limitations. In summary, we contribute :

- a controlled experiment to study selection of dense 3D datasets with traditional input devices, whose results shed light on the performance of three selection techniques, for two cell positions (on the surface or occluded),
- an analysis of participants' preferred strategies for *List*, *Explosion* and *Combination* as well as the involved two steps (discovery and traversal) of cell selection, and
- a discussion of selection techniques for dense 3D environments.

4.2 RELATED WORK

The actual tasks we employed in our work on selection techniques focus on object discovery and traversal, rather than simple picking. Below we thus first review related work about discovery and accessing techniques for 3D objects. We then discuss general interaction techniques besides selection for dense datasets, especially for desktop-based interaction. We end this section with a small survey of cell visualization applications—our application domain.

4.2.1 Discovery and Access Techniques

3D discovery is essential for finding the target cells among numerous cells. It needs to be able to deal with occlusion, yet should maintain the spatial relationship of an object and its context [59]. Elmqvist and Tsigas [59] summarized

a range of techniques to discover objects from densely datasets in virtual environments. They identified five design patterns : multiple viewports, virtual X-ray tools, tour planners, volumetric probes, and projection distorters. One of our approaches (explosion selection) falls into the last of these categories, while our list selection seems to be a separate category as it uses an abstract representation of the elements.

Though there were ways in dealing with the occlusion problem, the direct interactions including discovering are limited and to completely solve the occlusion, usually multiple techniques would be used [6]. To ease discovery, researchers have also used object highlighting or dimming the remainder of the objects. In the past, space distortion [69, 67, 60] and distinguishing the objects in a region [178] have been extensively studied for object highlighting, while object deaccentuation has been achieved with transparency [47, 57, 60] and selective object hiding [60]. These techniques, however, have not been fully tested for discovering a large number of objects such as in our case because the such datasets have high needs for orientation and an extreme lack of visual cues. Here, our application has an advantage : it is guaranteed that the sister cell, at any hierarchy level, is next to its sibling.

Multiple techniques have also been studied for precise accessing [59], and the spacial occlusion cases are most relevant for us. In 3D environments and, especially, VR, researchers have investigated using dedicated 3D selection tools to address the occlusion issue [6]. The most common techniques are ray-casting [107, 135, 121], ray-pointing [145], bubble cursor [121, 36], sphere-casting refined by QUAD-menu (SQUAD) [100] and virtual hand [152, 151]. Among these four, ray-casting and SQUAD were claimed suitable for dense objects [30] and numerous of studies have explored ways to improve these two techniques. For example, JDCAD [115] allowed people to use the cone selection to freely create the selection volume, which avoided the drawback of ray-casting that uses additional 1D input to select 3D objects. Grossman et al. [74] proposed a ray cursor that provided all the intersected targets and allowed users to choose. Later, Baloup et al. [10] developed RayCursor to automatically highlight the closest target and support manually switching the selection of intersected objects. As for the SQUAD, to offset the cumbersome steps in accessing dense objects, Cashion et al. [30] added a dimension called Expand to enable the sphere to zoom. Furthermore, to help accurately select an object users see, researchers have explored advanced access techniques that could calculate which object users would possibly select. For example, Haan et al.'s [42] IntenSelect technique dynamically calculated a score for objects inside a set volume and allowed people choose from the objects with the highest scores. Similarly, Smart Ray [74] continuously calculated and updated object weights to help users to determine which object to

select when multiple targets were intersected. All these techniques are efficient in discovering and accessing objects in sparse datasets, yet are not suitable for the highly dense environments with no space between possible selection targets. Moreover, in practical scenarios people are typically aware of which target to select, while in our cell division application the biologists make the decision by referring to the shared surface between the two cells and thus have to traverse a number of potential targets to assess their suitability. Also, the learning effects of new techniques could be high.

4.2.2 Interaction Techniques for Dense Datasets

In virtual 3D cell manipulations, biologists need to precisely select objects from dense sets, without knowing which objects may need to be selected. Previous studies [141] have demonstrated that users tended to stick with the familiar mouse interaction. In addition, past work [199, 14] has shown that low-DoF input devices such as mouse and keyboard can easily achieve such tasks with high accuracy. These supported our decision to study cell division with familiar input devices. Nonetheless, in virtual 3D environments—especially in VR—discovering an enclosed object can consume more time [6], even though the selecting is easier due to better depth perception in stereoscopy. In our dense embryo cells scenario we thus relied on a traditional projected-3D environment with mouse and keyboard input to accommodate our domain’s need for high selection accuracy.

Researchers have also explored various methods for mouse and keyboard input to manipulate the objects. For example, Houde [87] raised the idea of creating a handle box outside the 3D object and, similarly, modern 3D modeling applications such as Blender and Rhino allow users to individually transform the 3D objects with mouse and keyboard. Applications also provide layers for organizing the objects and selecting multiple items from a list. Even though in some controlled environments the object layout can be rearranged to avoid occlusion [177], in our case the cells’ spatial relationship must not be changed to provide our users with a faithful representation.

Past work on selection in dense datasets has focused on structure-aware approaches (e. g., [45, 46, 58, 215, 216]). Unlike particle or volumetric data which contains huge amounts of points or a sampled data grid without explicit borders, our embryo cell data has dedicated cells that could be picked—yet are tightly packed to each other such that many are not accessible for traditional picking. Lasso-based selection is also not appropriate because we do not need to enclose regions but need to match two dedicated objects as sister cells. We thus instead require interaction techniques that preserve the respective positioning at least locally and allow us to access all cells in an efficient and effective way.

4.2.3 Cell Visualization

Cell data visualization has been found to be useful in helping biologists get knowledge about cell development. Various academic tools (e.g., OsiriX [157], Fiji ImageJ [162], OpenWorm [186], and Icy [34]) and commercial software (e.g., Avizo, Imaris) provide advanced live-imaging techniques and computational approaches to allow users to clearly observe and interact with their data. The interaction in these tools, however, remains simple: mouse-clicking the cells on the surface of an embryo provides the users with access to specific variables and actions. For example, MorphoNet [110] uses Unity to visualize diverse types of cell data on a website, allowing users to visually explore cells. They left-click to target a cell, and can rotate and zoom using specific keyboard combinations. This interacting process is smooth for a few cells, while it gets slow and tedious for large datasets (i.e., with > 100 cells). Though the software can hide and show cells, it only provides access to the current outside of the embryo. No single tool among the mentioned software is applicable to the cell division annotation, so we worked to develop and study dedicated selection techniques for the entire embryo.

4.3 STUDY DESIGN

To understand how people can best select objects in densely packed 3D settings—in our application domain to discover target cells and traverse their neighbors—and, ultimately, to process large datasets using these interaction techniques, we designed the experiment as described below. We pre-registered this study (osf.io/cewhn/) including the study design and the data analysis methods (supplementary materials at osf.io/yze5n/), and it was also approved by our institution's (Université Paris-Saclay) ethical review board.

4.3.1 Interaction Techniques

We chose all the techniques based on previous related work and implementations biologists are using now. From our decisions to focus on desktop settings, an obvious interaction technique to select from a set of segmented objects is to use a list widget (see 4.3(a)). Participants could discover the target cells from the list only. It has the advantage of mapping the objects distributed in 3D space into a 1D dimension, for a given order in the set. Naturally, there is no such mapping that preserves the objects' original 3D location, but in our use case researchers need to access all of the cells from the set eventually. Moreo-

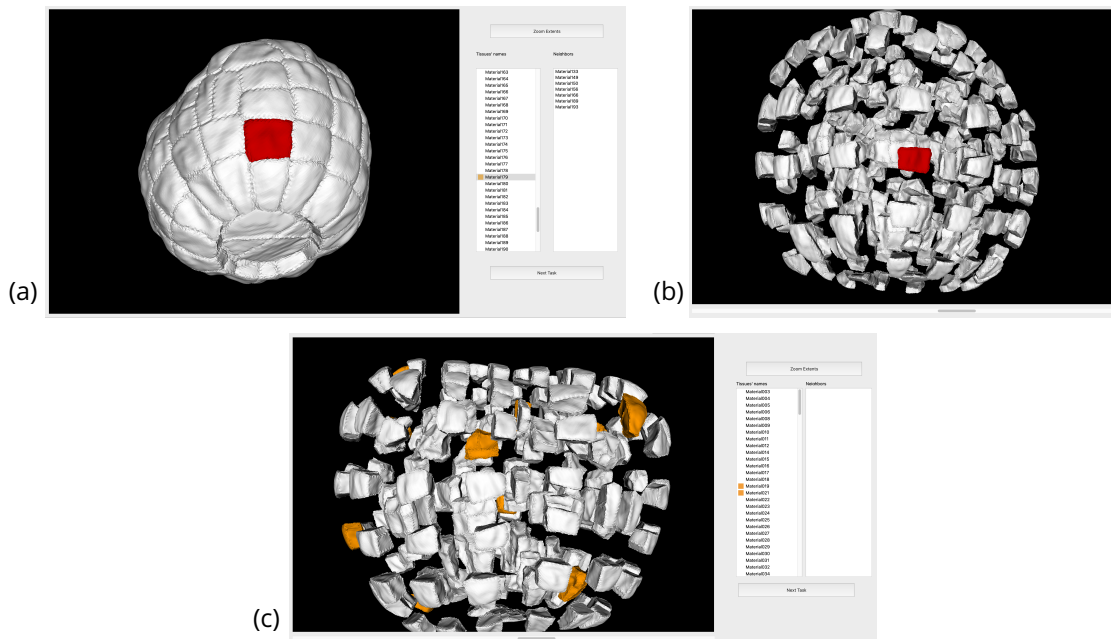


Figure 4.3 – Three main interaction targets for the techniques compared in the study : (a) List, (b) 3D Explosion, and (c) Combination selection. Cells to be assigned are marked in orange, and selected cells are red. In all three cases the 3D view was visible to the participants.

ver, this interaction also lends itself easily to the task of marking the cell division history, as we can algorithmically extract the potential sister cells of a selected target from the segmented dataset and show them in another list widget. For each item in the list, we only show a name because, in the real scenario, biologists refer to such names. In addition, we did not include additional data since they evaluate the shapes and neighborhoods of cells in the 3D view rather than making decisions based on numeric cell property values such as a shared surface area.

Nonetheless, the 3D location and 3D shape of the respective cells do play a role, both for the initial target selection (as researchers tend to solve the easy cases first) and for the decision on the sister cell (by inspecting the geometry of the shared surface). We thus were also interested in the performance of selection techniques directly in the projected 3D view. We solved the inherent object density and occlusion issues by employing 3D explosion techniques [114, 191]. Using this approach we created additional space between the cell objects, both for the initial selection of a target cell in the embryo (e. g., 4.3(b)), the examination and, ultimately, selection of the sister cells for this target (e. g., Figure 4.4).

Another fundamental approach to exploring the inside of 3D objects or vo-

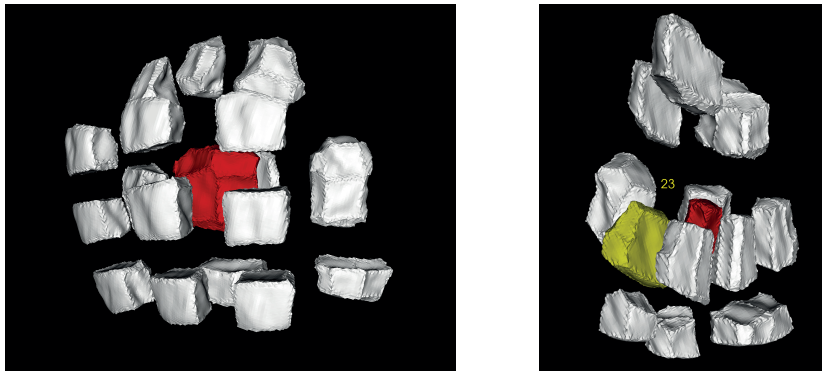


Figure 4.4 – The focused view of a target cell and the associated number shown near the neighbor cell’s surface (red cell is the target cell and yellow cell is the neighbor cell with its associated number).

lumetric datasets in visualization is the use of cutting planes (e. g., [82]). We also explored this technique as a basis for exploration and selection as it conceptually relates to the slices of the confocal microscopy approach in our application domain. With this technique, researchers would be able to move and orient a cutting plane freely in 3D space, and then we would show the intersected cells in an unprojected slice view where they could be clicked for selection. Pilot tests showed, however, that this approach was not promising because it was difficult to reason from the intersected cells to their correct 3D shape and correct selections took a long time, so we did not further pursue this technique in our experiment.

Instead, we also merged the first two techniques into a *Combination* technique in which participants had the choice between using *List* and *Explosion* selection. Moreover, in all techniques, including in the *List* selection, we showed the 3D projection of the embryo or a target cell’s direct environment as our collaborating biologists always make the decision of which two cells are sisters based on numerous properties, such as the shape and size of their interface (i. e., the shared surface between the two cells). We thus also used an explosion representation for the *List* selection technique, to guarantee that our participants can observe the shared surface. In the *Explosion* and *Combination* techniques, however, we allow users to freely adjust the explosion degree and to control the amount of space they need for navigating in 3D space.

4.3.2 Tasks

With these interaction techniques we aimed to support the practical task of deriving the cell lineage for an entire embryo. We thus modeled the tasks in

our experiment based on the approach our collaborating experts (three plant biologists, all with more than 20 years of professional experience) take to derive the cell division history as outlined in section 4.1, using the tools described in section 4.2. We followed the same process in our experiment : participants were first asked to select a non-marked target cell from the embryo. We then showed them this cell's immediate neighborhood in the focused view (Figure 4.4, both as a 3D view and, in case of *List* and *Combination* techniques, as a list), and then asked them to select the correct cell based on which cell is most likely the sister of the target.

This approach would naturally limit us to participants with years of experience in plant biology cell lineage analysis and the cell division scenario only. To circumvent these restrictions, we implemented a proxy for the biologists' experience : As we show a target cell's neighborhood, we asked participants to select each potential neighbor, after which we showed a pre-defined "likelihood" (an Integer $\in [1, 99]$) of being the correct sister cell. We chose this number randomly and independent of the specific situation because we were interested in general feedback on selection in dense environments with non-expert participants. We displayed this number in the 3D environment hidden from the current view to force participants to use 3D navigation (i. e., rotation) to reveal the number—this interaction mimicking the 3D evaluation of the interface between two cells that the biologists would do. Participants would then need to find the cell with the highest number to make a correct selection. In addition, this highest number was not necessarily 99, so that participants would have to examine each potential neighbor at least once.

4.3.3 Datasets

We used a real embryo data provided by our collaborators, which contained 201 cells. We chose this single dataset as a representative research entity because its size is realistic and other plant embryo datasets would contain similar cell shapes and arrangements. Experimental time limits, however, meant that participants could not assign sisters for all cells, we thus created three sets of target cells for them to mark, each with 10 cells. We were interested in the influence of the cell position (surface vs. occluded), so we created all three sets with 5 cells on the embryo's surface and 5 cells that were enclosed by other cells. To reduce learning effects, the three sets did not share a same cell, nor did they share any of the respective neighbors. Each set plus its 1-neighborhood (i. e., direct neighbors) was thus completely distinct from the other sets, plus their respective 1-neighborhoods, which guaranteed that any past assignment (even if done incorrectly) would not affect any future marking. Otherwise, if two target cells would have shared a potential neighbor, then participants marking this

neighbor as a sister of either target would mean that the other target would lose a sister candidate.

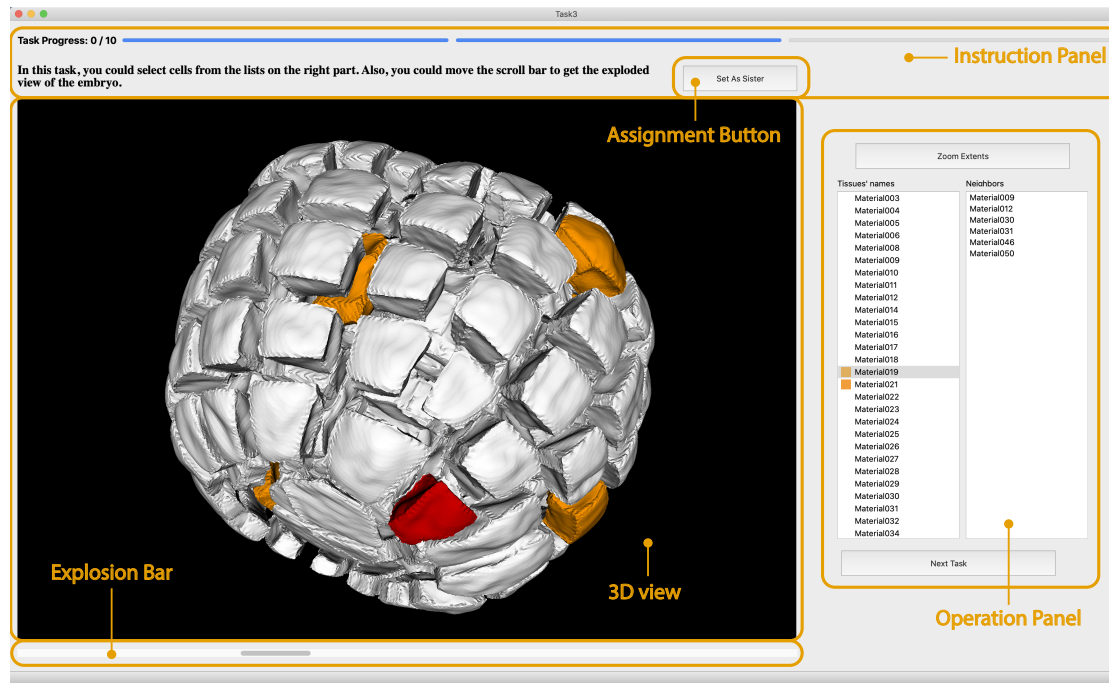


Figure 4.5 – Study interface (combination selection shown).

4.3.4 Interface

In three conditions, the interfaces contained three main parts : instruction panel, 3D view and operation panel (see Figure 4.5). The operation panel in all techniques contains two buttons. One could be used to auto relocate the whole embryo to the center of the 3D view, in case participants got lost, and another one enabled participants to jump to the next task. In *List* and *Combination*, this panel included a global list of all cells in the left list view and a focused neighbors list, showing only the direct neighbors of a selected target cell. We scaled the interface to completely fill the screen size of participants' computers, with the ratio of each part's size to the interface size being fixed. In the instruction panel, we displayed the study progress state and a brief introduction of the interaction in the task. We placed the 3D view on the left, while we showed the operation panel on the right. We designed the relative to indicate that 3D view was the main reference, and such that it was approximately square. Below the 3D view, we placed a horizontal bar widget to allow participants to control the explosion distance between the cells. We placed the button to mark two cells as sisters

on the top and in the center, somewhat in the middle between 3D view and operation panel such that the distances to travel to the button from 3D view or lists were about the same. We also allowed participants to assign cells by pressing the space in the keyboard to further reduce the impact of the actual marking action on completion time.

For indicating cells from the sets to be marked, we highlighted them in the list via orange icons for *List* and rendered the cells' 3D shapes in orange in the 3D view for *Explosion*. In *Combination* mode, we used both forms of highlighting. When participants clicked on a cell either in the 3D view or the lists, we also showed the corresponding item in the lists and the cell in 3D view would in red (for target cells) or yellow (for neighbor cells) in the 3D view or highlighted in the list as shown in Figure 4.5. Finally, we modeled the interaction in the 3D view after commercial 3D modeling software like Rhinoceros or Blender. Participants could hold the right mouse button to rotate, scroll the wheel to scale, and hold the wheel to pan. To distinguish rotating from clicking, the left button of mouse in the 3D view could be used to click and double click the cell.

4.3.5 Measures

We assigned a unique participating number to every participant and recorded all data based on this number to guarantee participant anonymity. For all trials, we recorded total completion times, accuracy, every action participants did, and tracked the real-time position of the camera. We started the timer when the program had loaded the visualization for each trial and stopped once the participant triggered the signal of assigning the cell sister (button click or keyboard press). We asked participants to activate the assignment once they found the sister. After choosing the sister for the target, these two cells would disappear in the 3D view and the corresponding items in the lists would also be disabled. We then instructed participants to continue with the next assignment and we restarted the timer. We measured the total trial completion time and accuracy by calculating the ratio of correct assignments in all assignments. Aside from completion time and accuracy, we also recorded the cell selection ratio (clicking times divided by the neighbor count) to better understand the efficiency of different techniques. A more efficient selection technique was likely to have lower clicking ratio, one that is closer to 1. After participants finished all tasks, the examiner conducted a post-study semi-structured interview, focusing specifically on the following questions : **Q1**—Sort the three techniques by preference ; **Q2**—What strategies did you use in doing three tasks? and **Q3**—Do you have any other comments on the interaction?

4.3.6 Participants

As our goal was to generally understand object selection in dense datasets and to provide recommendations also for non-botany scenarios, we targeted non-expert participants. Also, recruiting such participants ensures that all the decisions are made by referring to the associated numbers, rather than being fully or partial based on our participants' own knowledge of cell division. We recruited 24 people via social networking and our local university's mailing list (8 females, 16 males; 24–31 years old, with a mean age of 26.96 years). All participants had at least a master degree, were right-handed, and were well trained in the usage of mouse and keyboard interaction. None of them was color deficient. Twelve of them had previous experience in 3D manipulation including 3D video games playing, and none of them had knowledge about cell division before. The latter aspect is important as it suggests that all participants made their assignments only based on the number we showed, rather than their previous knowledge of cell division patterns.

4.3.7 Procedure

We conducted the experiment via remote video calls due to the limitations that arose from the Covid-19 pandemic for our research environment and for the participants. We minimized the remoteness effects by checking in advance whether every participant could smoothly conduct the experiment with their preferred devices. We first explained participants the purpose of our study, asked them to fill in basic demographic information, and sign a consent form if they agreed to participate. Because we conducted the study online, for those participants who preferred not to install our experimental software by themselves, we asked them to use a dedicated remote interaction software to allow them to remotely control the experimenter's computer. The others had downloaded the software and installed the software in advance and shared their screen while they communicated with the researcher via video conferencing.

We divided the experiment into three blocks, one for each technique. Each block began with a non-timed training session in which the experimenter first explained the task using written instructions in the interface and a study script, and then asked participants to try their best to traverse all the neighbors of a target cell and to find the correct answer as soon as possible. Before transferring to the main task, the experimenter ensured that participants understood the task and were able to conduct the tasks correctly and independently. After finishing all tasks, we conducted the mentioned post-study interview to explore participants' strategies and individual experiences.

Our first objective with the experiment was to compare the *List* and *Explosion*

techniques. We thus only presented these two techniques in the first two study blocks. We counter-balanced the order of both techniques to reduce order effects. Our second objective was to assess how participants would interact when having the choice of using the *Combination* technique, after having experienced the *List* and *Explosion* techniques separately. In the third block we thus always presented the *Combination* technique to participants. In addition, we were interested in the effect of occluded vs. surface cells, so we alternated between these types and also counter-balanced the type a participant would see first. We did not expect an effect of the specific order of cells in the list view, so we always used the same order (by name) for all participants. In *List* and *Explosion* tasks, we showed the next target cell in orange after participants had finished the former assignment, while we marked all target cells at the start of a *Combination* task to explore in which sequence participants would assign them. The order of the specific cell subsets may play a role, so we counter-balanced the order of the three subsets. In total, we thus had a 2 techniques \times 2 cell types \times 3 data subsets design, resulting in 12 combinations in total, and each possible combination was experienced by two participants. We used 10 trials per technique and the resulting experiment lasted about one hour per participant.

4.4 RESULTS

We now present our experimental results of completion time, accuracy, and clicking ratio for the two selection techniques *List* and *Explosion*. We then individually examine the use of *Combination*, which we cannot analyze together with the other techniques due to potential order effects. We also compared the performance of the different techniques in assigning cells from two positions (on the surface or occluded). Cells on the surface (surface cells) typically have less neighbors and clearer layers, while enclosed cells (occluded cells) are hidden entirely from an outside view. We also discuss our participants' strategies and subjective feedback.

We gathered totally 720 trials (24 participants \times 3 tasks \times 10 trials). Recent recommendations from the statistics community made us choose an analysis of the results using estimation techniques with confidence intervals (CIs) and effect sizes to avoid the dichotomous decisions [16, 51, 103], instead of using a traditional analysis based on p -values [9]. However, it is still possible to transfer CIs to p -values [50, 103]. We report all CIs by default as 95% CIs. We did not find all measurements to be normally distributed, so we used bootstrapping CI [95] to analyze completion time, accuracy, and clicking ratio. We visualized our output distributions to increase the transparency of our reporting.

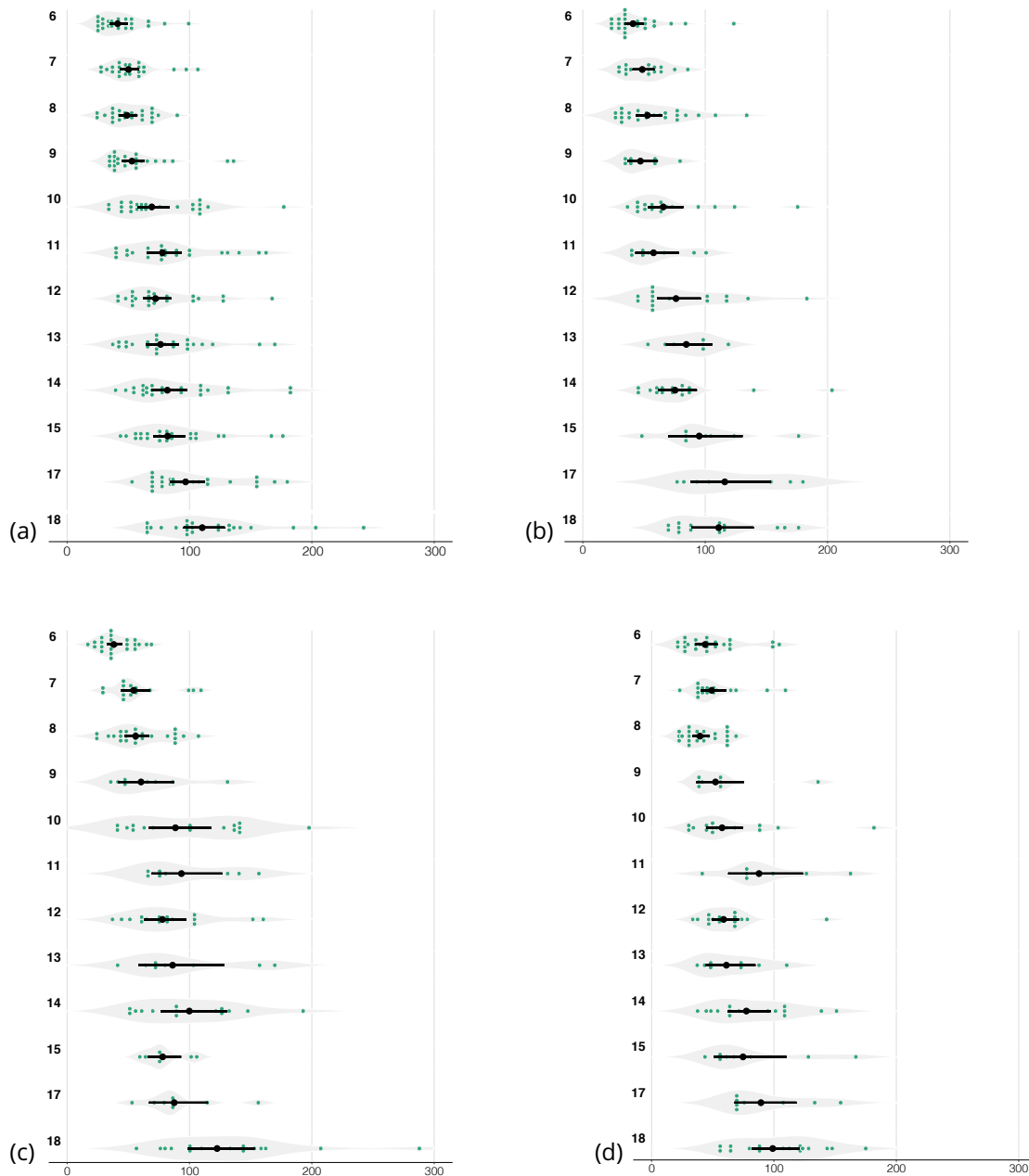


Figure 4.6 – Completion time (absolute mean time) for different numbers of cell neighbors in seconds : (a) overall time, (b) *List* selection, (c) *Explosion* selection, and (d) *Combination* selection.

4.4.1 Completion Time

We can naturally assume an impact of neighbor count on completion time and we indeed observed an approximately linear relationship—globally for all

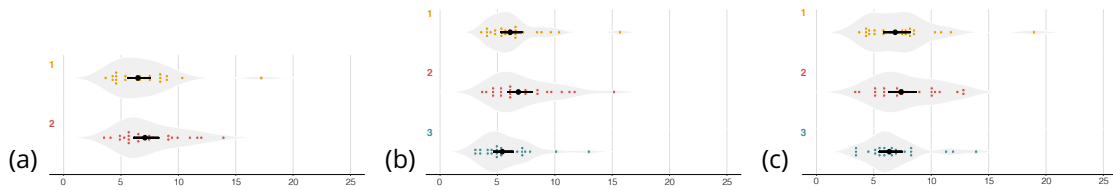


Figure 4.7 – Completion time (absolute mean time) in seconds (*List* in yellow and *Explosion* in red) : (a) the overall results, (b) selection of occluded cells, and (c) selection of surface cells.

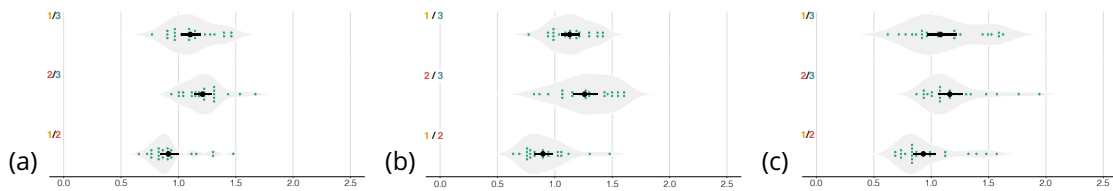


Figure 4.8 – Pair-wise differences for completion time : (a) the ratio overall, (b) the ratio for occluded cells, and (c) the ratio for surface cells.

tasks (4.6(a)) and also for the individual tasks (4.6(b)–(d)). The mean neighbor count per dataset, however, was approximately similar (10.4 vs. 10.1 vs. 10.8). Moreover, each combination of task with dataset was seen by the same number of participants (fully counter-balanced), so in our remaining global analysis of completion times this relationship does not play a role.

Techniques. In Figure 4.7 we present the absolute mean values of time in seconds for each technique. With *List*, the average time is 63.81s (CI [56.25s, 74.82s]), while using *Explosion*, the average time for one target cell is 69.75s (CI [60.64s, 80.26s]). Since the CIs overlap a lot, to better demonstrate the difference in the completion time, we checked the pair-wise ratio for these two techniques (see Figure 4.8). The ratio for *List/Explosion* is 0.91 (CI [0.86, 1.01]). As we can see, the upper bound CI of *List/Explosion* is 1.01, close to but above 1, so there is some evidence that the *List* selection tool less time than *Explosion*. The absolute difference, however, is only small as evident in the similar completion times. We also investigated the completion time differences with these two techniques in two task parts : discovery and traversal. For the discovery part (i. e., the accumulated times from the start of a trial to the selection of the target cells), the average mean times are 7.57s (CI [6.79s, 8.52s]) with *List* and 5.23s (CI [4.31s, 6.36]) with *Explosion* (see 4.9(a)). Since the upper bound of CI in *Explosion* is smaller than lower bound of CI in *List*, the *Explosion* is evidently faster in discovering target cells than *List*. We also checked the pair-wise ratio of *List/Explosion* and it is 1.45 (CI [1.27, 1.69]), which confirmed that *List* selection needed more time than *Ex-*

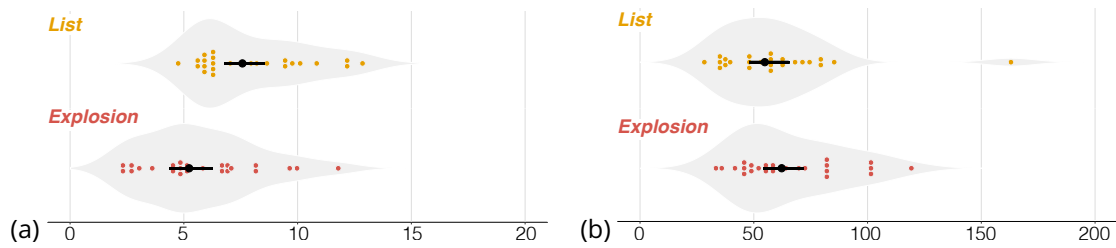


Figure 4.9 – Completion time (absolute mean time) in seconds with two steps (*List* in yellow and *Explosion* in red): (a) the target cell discovery, and (b) neighborhood traversal.

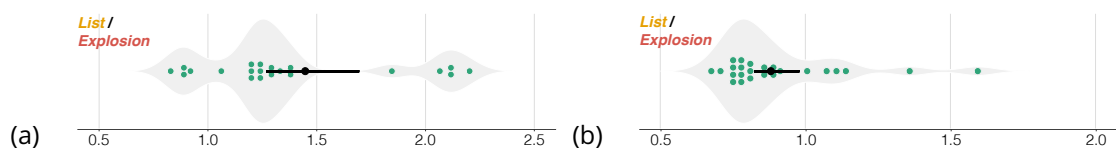


Figure 4.10 – Pair-wise differences for completion time in two steps: the ratios for (a) discovery and (b) traversal.

Explosion (see 4.10(a)) for object discovery. As for traversing (i. e., the accumulated times for checking all neighbors of a cell), the average time for *List* is 54.84s (CI [47.98s, 65.12s]), while for *Explosion* it is 62.26s (CI [54.37s, 71.49s]) (see 4.9(b)). Because the CIs overlap a lot, we examined the pair-wise ratio to better analyze the difference. As 4.10(b) shows, the ratio for *List/Explosion* is 0.88 (CI [0.82, 0.98]), so there is some evidence that *List* selection is faster for traversal than *Explosion*.

Positions. We were also interested in the possible influence of the cell position on performance. We investigated the average completion time for occluded cells (4.7(b)), which was 79.42s (CI [69.83s, 93.52s]) in *List* and 88.58s (CI [77.43s, 102.33s]) in *Explosion*. Because this difference of mean completion times is small and the CIs overlap, we again checked the pair-wise ratio, which is 0.90 (CI [0.84, 0.97]). The upper bound of the CI is again close to 1.0, so there is some evidence that with *List* participants could finish the task quicker than *Explosion* when dealing with occluded cells. We did the same analysis for surface cells. Here, the average times are 51.62s (*List*; CI [45.05s, 61.23s]) and 54.92s (*Explosion*; CI [46.87s, 63.27s]), and the pair-wise ratio for *List/Explosion* is 0.94 (CI [0.86, 1.06]). We thus cannot find much evidence that, in assigning surface cells, *List* selection would be faster than *Explosion*.

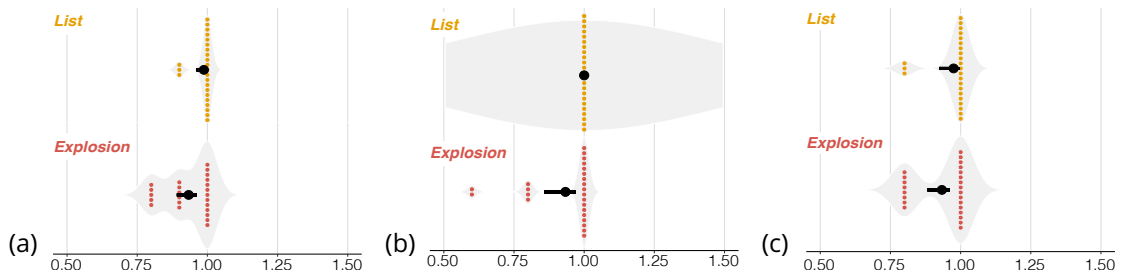


Figure 4.11 – Accuracy rate (*List* in yellow and *Explosion* in red) : (a) overall, (b) selection of occluded cells, and (c) selection of surface cells.

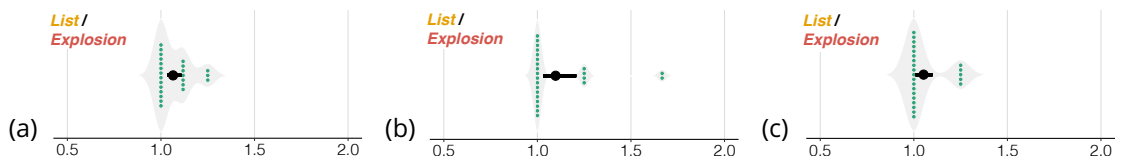


Figure 4.12 – Pair-wise differences for accuracy : (a) the ratio overall, (b) the ratio for occluded cells, and (c) the ratio for surface cells.

4.4.2 Accuracy

We measured the accuracy of the assignments with two techniques (*List* and *Explosion*) and two positions. We calculated the accuracy by dividing the correct assignments count by the total trials count.

Techniques. We report the absolute mean values of correctness in two techniques in Figure 4.11 and the pair-wise ratio for comparison in Figure 4.12. The accuracy was high in both techniques so we kept three decimals for a better comparison. For the *List*, the absolute mean value of accuracy is 0.987 (CI [0.963, 0.996]), while in *Explosion*, the value is 0.933 (CI [0.892, 0.958]). From 4.11(a) we can see that all participants found at least 8 correct sisters (as every participant used each technique to make assignments for 10 cells). In addition, the fact that CIs do not overlap provides evidence that *List* resulted in more accurate assignments than *Explosion*. We also analyzed the pair-wise ratio (*List/Explosion*) to better understand the difference, which was 1.06 (CI [1.03, 1.10]). This result provides evidence that *List* works more accurate than *Explosion*, although the mean accuracy values are similar and are both high.

Positions. We also present the absolute mean values of accuracy for the two positions in the two techniques in Figure 4.11 and the pair-wise ratios between them in Figure 4.12. For occluded cells, the absolute mean values of *List* and *Explosion* are 1.000 (CI [NA, NA]) and 0.933 (CI [0.858, 0.967]), respectively (4.11(b)). Using the *List* technique, all participants thus assigned all occluded cells correctly

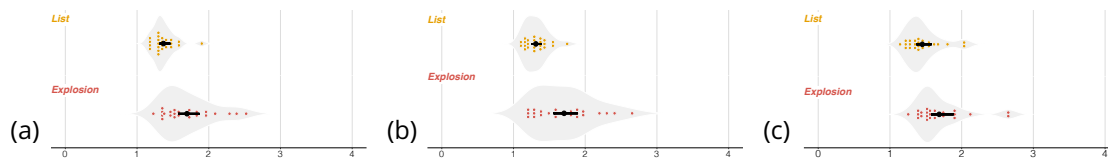


Figure 4.13 – Clicking Ratio (*List* in yellow and *Explosion* in red) : (a) overall, (b) selection of occluded cells, and (c) selection of surface cells.

and we can say that the *List* technique achieved more correct assignments than *Explosion*. The pair-wise ratio (*List/Explosion*), which turned out to be 1.10 (CI [1.03, 1.20]), confirms this finding, yet its lower bound being close to 1 makes this result only weak evidence. For the surface cells, the absolute mean values for the two selection techniques (*List* and *Explosion*) are 0.975 (CI [0.925, 0.992]) and 0.933 (CI [0.883, 0.958]). The largely overlapped CIs show limited information for the differences. The pair-wise ratio is 1.05 (CI [1.01, 1.09]) which also only provides weak evidence that *List* performed more accurately than *Explosion* for surface cells.

4.4.3 Clicking Ratio

We also counted the click events in both the lists and on the 3D view. We separated the clicks needed for rotation in the 3D view for both techniques as these were right clicks—in contrast to the left clicks in the list or 3D view for selection. Thus, we only counted clicks to access cells. We defined the clicking ratio as the average times participants clicked on every neighbor to get the right answer, i. e., the click counts divided by the number of neighbors. Ideally, participants click all neighbors once to find the right sister, with a clicking ratio of 1. In practice, however, participants usually clicked one same cell for multiple times. We chose this variable as a factor to evaluate the efficiency of the selection techniques. The more this number deviates positively from 1, the worse is the efficiency.

Techniques. We report the absolute mean values of the clicking ratio for the two techniques in 4.13(a). *List* had the smallest absolute mean value which with 1.37 (CI [1.32, 1.45]), while the value for *Explosion* was 1.70 (CI [1.58, 1.86]). Though the CIs are non-overlapping and there is evidence that supports that *List* has a lower clicking ratio than *Explosion*, to further explore the differences we also calculated the pair-ratio of *List/Explosion* (4.14(a)). The ratio turned out to be 0.84 (CI [0.77, 0.90]), which provides good evidence that *List* required less clicks than *Explosion*.

Positions. We also examined the absolute mean values of the clicking ratio for the two positions. The absolute mean values for occluded cells are 1.31

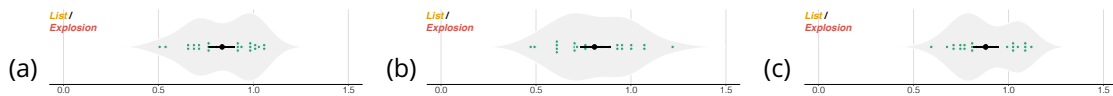


Figure 4.14 – Pair-wise differences for clicking ratio : (a) the ratio overall and (b) the ratio for occluded cells, and (c) the ratio for surface cells.

(*List*; CI [1.26, 1.38]) and 1.71 (*Explosion*; CI [1.56, 1.88]) respectively. The upper bound CI of *List* being much smaller than the lower bound CI of *Explosion* provides evidence that *List* required fewer clicks than *Explosion*. The pairwise ratio (*List/Explosion*) being 0.81 (CI [0.73, 0.89]) confirms this assessment. For the surface cells, the mean values are 1.45 (*List*; CI [1.37, 1.56]) and 1.69 (*Explosion*; CI [1.58, 1.87]) as shown in 4.13(c). The confidence intervals are close to we further checked the pair-wise ratio (*List/Explosion*), which is 0.88 (CI [0.82, 0.94]). This evidence supports that using *List* required fewer clicks than *Explosion* also for surface cells.

4.4.4 Techniques Used in Combination

We analyzed the *Combination* technique individually because we presented this technique to participants always last—participants first had to learn the individual techniques. In *Combination*, participants were able to complete the task freely, with both *List* and *Explosion* available to them. We were interested in how participants would combine them and whether the neighbor number would influence their choice. We thus calculated the proportions of their click counts in the *List* condition (over *List* plus *Explosion* clicks together) to present the strategy, which we show in 4.15(a) (top bar; the *Explosion* click proportion is the complement of the *List* proportion). The absolute mean value of the list proportion is 0.87 (CI (0.85, 0.90)), meaning that participants clicked more frequently in the list widgets than in the 3D view (for discovery or traversal). We also calculated the proportions for discovery and traversal separately, whose ratios are 0.50 (CI [0.37, 0.63]) and 0.79 (CI [0.75, 0.83]). We also analyzed the list clicking proportion individually by cell neighbor counts (4.15(b)). As we had noted already, however, the numbers of neighbors varied depending on the dataset and some neighbor counts received only few trials. We thus only analyzed those numbers which had more than 10 trials. In all cases, the average values of the percentage are higher than 0.5, which means participants clicked more often in the list widgets than in the 3D view. Although the differences are small, we observed that the *List* click proportion increases with a growing number of neighbors. While these numbers suggest a strong preference for list interaction, this observation is skewed by the fact that by far the most clicks naturally happened in the traver-

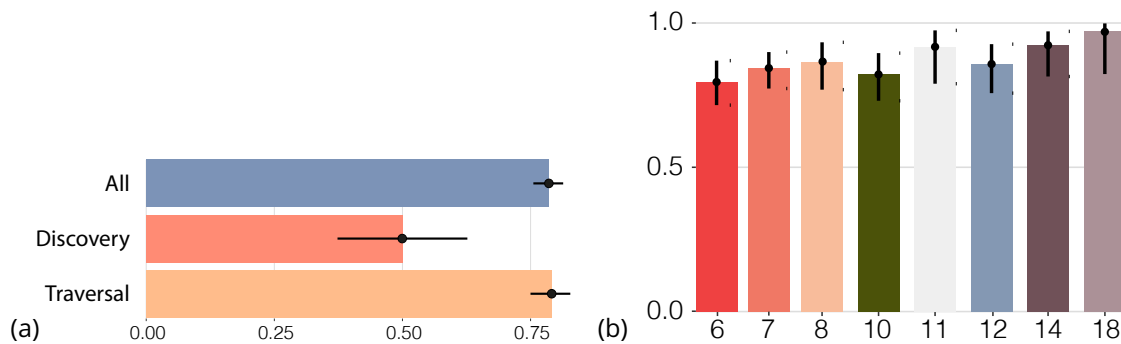


Figure 4.15 – Clicking proportions of *List/(List + Explosion)* in the *Combination* task : (a) overall and (b) by neighbor count (for discovery + traversal; x represents the numbers of the cell neighbors, and y represents the clicking proportions).

sal phase (0.082% on average). Looking only at target cell discovery, however, in the post-study interview feedback 13/24 participants stated that, after trying and adjusting their strategies, they finally chose to examine the exploded embryo in the 3D view to find the target cells, while the other 11/24 participants checked the list by scrolling from the top to the bottom. We show this difference of strategies in the click proportions in the two lower bars in 4.15(a). We also investigated, for the *Combination* task, the order participants chose to assign the cells. According to our logs, 8 participants always stuck to the list order, without taking the cells' positions into consideration. Another two participants switched the strategies and finally followed the list order. Others simply clicked on random orange cells they saw.

4.4.5 Task Strategies

We were also interested in our participants' approaches to finding target cells and traversing the neighbors, especially for the *Explosion*, and their choice of methods for the *Combination* condition. Here we report the strategies based on participants' statements in the post-study interview, combined with our observations of the participants as they interacted during the experiment. In the *List* condition, all participants scrolled up and down the cell list to find the orange item and then traversed the neighbors by going through the neighbor list. Participants memorized the largest associated number and either the cell name or its position in the list to complete the task.

Because we provided no lists in the *Explosion* condition, participants could not rely the same strategies as with the *List*. We thus specifically asked them about their detailed strategies in the 3D explosion condition, organized their

ideas, and grouped similar points. To help with traversal, 8/24 (33.3%) participants stated that they mentally divided neighbors into different layers and zones based on the spatial placement. For staying oriented, 7/24 (29.2%) participants rotated back to the original position every time when they finished checking the associated number of one neighbor, while 4/24 (16.7%) tried to rotate the embryo by only one fixed axis. One participant kept the best candidate cell on top during traversal. Another participant observed the relative positions of the cells and matched them into a special shape like a sphere or triangle. Then he traversed neighbors by referring to his chosen shape's corner cells. Other participants tried to memorize the cell shape, their 3D relative position, and the temporally largest number during the trial.

During the *Combination* task, 10/24 (41.7%) participants used the same steps as they did in *List* because they were afraid to get lost in 3D interaction. One person exclusively used the *Explosion* interaction in the *Combination* task because she was bored to scroll the long list. Another 10 participants discovered target cells with *Explosion* and traversed neighbors with the *List* technique. Only 3/24 (12.5%) participants chose the techniques based on the number of neighbors. When this number was small, they used *Explosion*, and otherwise the *List* technique. Among them, two participants discovered target cells with direct interaction in the 3D view, while the other one searched the target cells in the list.

4.4.6 Subjective Feedback

In the post-study interview we asked about participants' preferences for the three techniques and their general thoughts on the interaction.

As Figure 4.16 shows, more than a half of participants (16/24) liked the *Combination* selection most. Two participants considered the *Combination* and *List* to be equally satisfying, while another one favored the *Combination* and *Explosion* techniques equally. The remaining 5/24 participants preferred the *List* technique. For this technique, participants appreciated its item order (e. g., "*much easier to follow which have been clicked*"). However, the interaction was troublesome (e. g., "*was boring to scroll the list,*" "*I had to fast move the mouse cursor between the lists on the right and 3D cells on the left*"). Moreover, when the associated number was similar to the cell name by chance, it was easy to get confused (e. g., "*I got messed up with the name and associated number. I forgot which one was the temporally best candidate cell.*"). Meanwhile, they stated that they did not pay attention to information such as the shape and 3D relative position of the cell because they only looked at the associated number in the 3D view and otherwise focused on the list ("*[I] only remembered the numbers and did not examine the shape*"). In the *Explosion* condition, participants appreciated the convenience to fast click on the cells (e. g., "*all [are] the interactions in the 3D view*") and the usefulness of

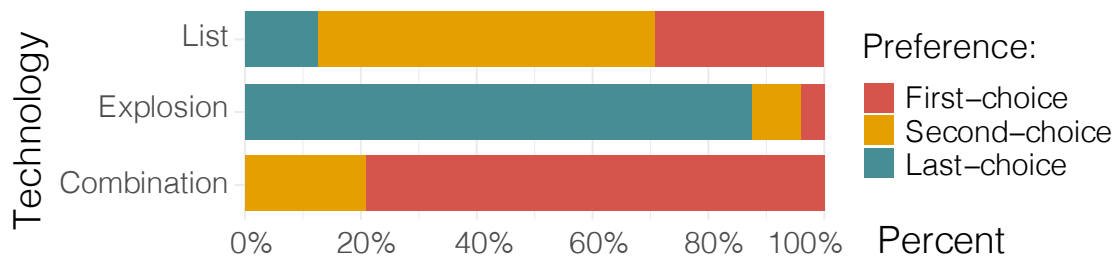


Figure 4.16 – Accumulated participant preference ranks. Note that we allowed participants to rank two techniques as their first choice and then counted none as the second, resulting in ranks 1, 1, and 3.

being able to control the distance between two cells (e. g., “*spreading out the cells is useful in targeting cells*”), but they disliked the need to rotate the view because this led them get lost and forget which cells they had already examined (e. g., “*less useful in checking out neighbors,*” “*it was easy to get lost when rotating the embryo ... I am not sure whether I have traversed all the cells or not*”). For the *Combination*, participants liked the freedom to spread out cells and the convenience of the default order in the list (“*supports both techniques and I could be quicker*”). Nonetheless, some participants would just use the same technique they preferred in the previous two tasks and thought it was useless. Others reported confusion (“*I struggled to choose the technique*”). One participant also reported being bored and tired in doing the last task.

Commenting on the whole interaction, participants proposed some changes (e. g., “*The interaction is good, and it will be better if there is a mark on the cells I have checked in all techniques,*” “[I] would like to have more context in the background of the 3D view to help orientation,” “[you should] show the name of cells in 3D view so that I could have a name order to follow,” and “*hiding the least possible candidate cell manually would accelerate the process*”). Some participants thought the two techniques should not be combined. One participant, e. g., stated that “*List has an order and 3D view has another order (layer). These two orders do not have a similar logic or strategy and could not be combined. These two techniques in the same interface will disturb each other’s use ... could present a 3D order based on the 3D position and link to 2D order in the list.*” Though most participants liked the explosion bar, one argued that horizontally moving the bar, for him, did not intuitively represent the conceptual increase of inter-cell distance.

4.5 DISCUSSION

4.5.1 Performance Differences

We found evidence that *List* led to more efficient (faster, fewer clicks) and more precise input than *Explosion* overall. This indicates that traditional list-based selection was more familiar to participants, compared with 3D interaction which was unfamiliar to many. Moreover, the *List* condition provided an order of the potential neighbors of a target, which supported participants in traversing every cell in the list without missing one as well as remembering the cell with the highest associated number, regardless of potential view manipulations in the 3D view. In contrast to the overall results and the results for occluded cells, we did not find clear differences in completion time and accuracy of two techniques for studying surface cells. This finding may be due to the fact that surface cells usually have fewer neighbors and a clear arrangement of the cells such that participants had less problems when traversing these in the 3D view.

We also found that a direct interaction in the 3D view has advantages. While the *List* condition enabled participants to traverse neighborhoods faster than with the *Explosion* technique, with the latter participants were faster in discovering the next target. This last point probably is due to the 3D view showing all remaining targets in a single view (with only some rotation necessary), in the lists participants had to use scrolling to get to the next target. In the traversal, in contrast, the lists of potential neighbors had a lot fewer entries than the overall list of cells, so that the participants did not need to scroll and thus their speed improved. Moreover, the need to rotate the 3D view to traverse all neighbors often led to participants losing orientation such that they no longer remembered which cells they had looked at already.

While this is a problem that was apparent in our pool of participants, the situation may be very different in our envisioned application domain of plant biologists constructing lineage trees. Here, the experts will not look for numbers but instead investigate the potential sister cells based on the cell's overall shape as well as the size and shape of the shared surface between the cells, properties that are essential for making the lineage decision. This means that the plant biologists not only inherently have to focus much more on the 3D view, but they also do not necessarily traverse all neighbors because they can easily reject some candidates based on their shape. Because we had to use a number associated to the cells as a proxy for the biologists' experience, our participants, in contrast, only focused on this abstract property and thus could more easily focus almost entirely on the list as their main reference point, which in turn likely

led to the *List* condition's performance advantage.

4.5.2 Subjective Ratings

We can also find these assumptions supported by our participants' qualitative feedback. In particular, they preferred the *List* technique because they felt it led to a lower mental load, requiring less memorization. Essentially, because they were not experts they turned our envisioned spatial decision into an abstract task because they did not need to examine the cell's shape etc. They thus focused on and used the arbitrary order of cells in the *List* condition. Consequently, our participants also disliked that they had to move back and forth between list and 3D view in the *List* condition.

In the *Explosion* condition, in contrast, participants liked to be able to explode the embryo, to freely explore it, and to have a whole view and direct access to the cells. The downside of this aspect was the lack of a clear order of the elements that they could follow to traverse all neighbors. Moreover, the needed rotations made participants more likely to lose the orientation in the 3D view, and consequently also to forget which of the already visited cells had the highest associated number. Participants had to memorize this intermediate result based on the cell's shape and 3D position, which was much harder for them than memorizing a position or a label in the 1D list. While these aspects made the task more mentally demanding for participants compared to the *List* condition, experts likely will not suffer from the same problems as we noted above.

Another problem with the *Explosion* condition was that the discovery phase and the traversal phase needed different view configurations: in the former participants needed to see all cells of the embryo, while in the latter they needed to focus on only the 1-neighborhood of a single cell. We had specifically ensured that the positions of the cells did not change when switching between overall and focused view to maintain spatial continuity; yet this meant that in the *Explosion* condition participants had to frequently manipulate the view (adjust the zoom factors). In the *List* condition, in contrast, we automatically centered the view on a newly selected target because people focused on the overall cell list when selecting targets, which led to much less need for view adjustments.

4.5.3 Implications

One of our main insights is that **3D interaction techniques work best for truly three-dimensional tasks which have no additional informative tags**. When we asked participants to perform a purely 3D action such as to discover colored objects among a set of exploded cells of the embryo, e. g., the 3D *Explosion* technique performed well and our participants used them when they had the choice.

In contrast, for tasks like the traversal which our participants converted into an abstract search task as we had discussed, the *List* technique was faster, more accurate, and preferred. As we discussed in sous-section 4.5.1, for the realistic task in the biology domain the actual sister cell selection is likely much more a 3D task than our proxy, so we hypothesize that the *Explosion* technique will be a strong competitor (but this will have to be verified in a separate experiment).

We also found that ***the use of explosion techniques as an interaction metaphor makes it possible to access objects in tightly packed 3D environments***, such as for selection as in our application. For discovering target cells, our participants increased the distance between two cells and zoomed out to have a clear overview of the embryo and the relative positions of cells, while for traversal, they tended to shorten the distance and zoomed in so that they could examine cells and find a structure to traverse. Also, our participants reported that they would freely adjust the distance between two cells to have a better overview or check cell details.

Next, the ***Combination seems to combine the advantages of the single techniques***. While we always showed it last to participants and thus cannot rule out order effects for its performance, participants clearly preferred this type of interface over only the (1D) *List* or the (3D) *Explosion* interaction. It allows users to freely choose which technique works best for them, for a given task and dataset, and also allows them to transition to a 3D interaction as they progress and as 3D aspects become more important. Nonetheless, even though with the *Combination* both individual interaction methods were available to participants, a constant ***switching between 3D view and lists is inconvenient***. Participants who preferred to use *List* chose strategies that operating the objects in the right part of the 3D view which is placed close to the lists, while others tried to directly interact in 3D view.

While we studied the specific scenario of cell division analysis in botany, we believe that our results can apply to or, at least, inform many other settings in which objects need to be selected from dense environments. Even if more work will be needed to confirm the applicability, those contexts include machine assemblies [191] and datasets in brain connectomics [19]. In such settings, experts similarly need to be able to select parts with virtually no space in-between, and have to be able to understand spatial and logical relationships between neighbors. Also, we designed our experiment specifically such that participants were not experts from our application domain of biology, but came from the general public.

4.5.4 Limitations

Naturally, our work is not without limitations. We already pointed out that, while we aimed to replicate the biologists' spatial analysis task as well as possible in our experimental setting, it turned out that our proxy for "experience" allowed participants to turn the 3D spatial analysis task into an abstract search task, and we have explained the implications of this change in sous-section 4.5.1. While in the future we plan an empirical validation with experts, we think that our work still sheds valuable light on how we can realize selection and access tasks in tight 3D environments.

Beyond this point, the fact that we were required by our IRB to conduct our work via video conferencing also may have affected the outcome. Naturally, participants had different types of equipment (screen resolution and size, PC power, general environment, etc.). An on-site experiment may have resulted in a more controlled environment and procedure. Nonetheless, this spread of environment reflects real-world working conditions, so we do not see this point as a strong limitation. Next, our specific choice of application case and, consequently, study dataset is a unique setting: all cells in the dataset were of roughly the same size and were "well" distributed. Other datasets in other application domains—even if they are densely packed—may have different properties and may thus lead to slightly different selection performance. Yet we believe that our general conclusions still hold. Finally, we only tested manual selection techniques. In the future, however, we foresee the use of machine learning (ML) approaches to support the biologists in establishing the cell lineage and, thus, the interaction requirements will change from manual selection to ML supervision and verification.

4.6 CONCLUSION

We have advanced our understanding of interaction techniques for the selection of objects in dense 3D environments with our chosen example of cell lineage assignment, but completed by members of the general public. We saw that a list-based selection has advantages when the number of elements is large and when the needed information can be represented in (or "projected" to) lists. We also saw, however, that if the relevant criteria are three-dimensional properties then an explosion-based selection can have advantages, in particular when the target audience is familiar with orienting themselves in 3D space. A combination of both techniques, ultimately, provides the best of both worlds.

LINEAGED : INTERACTIVE VISUAL SYSTEM FOR PLANT CELL LINEAGE ASSIGNMENTS BASED ON MACHINE LEARNING

5

The previous study described in chapitre 4 uncovered the efficient 3D selection techniques for the densely packed embryos. For assisting the cell lineage problem, besides supporting the precise cell selection, a system should also provide enough details for decision-making (3D spatial data and hierarchical information). That requires us to answer **RQ2** and **RQ3** in chapitre 1. In addition, since the manual work is time-consuming, it will be optimal to include a machine learning model to help predict the assignments as well as be easily accessible. We, thus, decided to develop a web-based system with the explored selection techniques to systematically support biologists in completing the cell lineage process for a new embryo dataset.

In this chapter, we describe LineageD—a hybrid web-based system to predict, visualize, and interactively adjust plant embryo cell lineages. Currently, plant biologists explore the development of an embryo and its hierarchical cell lineage manually, based on a 3D dataset that represents the embryo status at one point in time. This human decision-making process, however, is time-consuming, tedious, and error-prone due to the lack of integrated graphical support for spe-

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

5.1 INTRODUCTION

How individuals of species develop is an essential question in specific sub-fields of biology. By looking at the historical development of a plant embryo, botanists investigate whether the plant develops normally or not. As an embryo develops, single cells divide into two new cells and these parent-child relationships can be described using temporal hierarchies. The process of creating these hierarchies is called *cell lineage assignment* and requires biologists to (1) take images of a multi-cell plant embryo using a confocal microscope, (2) segment the resulting 3D volumetric data to identify the embryo cells, and (3) indicate pairs to reconstruct the hierarchy backward toward the proper embryo as the first cell (i. e., the *ovum*) of the embryo. The Figure 5.1 illustrates the overall traditional process further.

In our collaboration with botanists, we identified several parts of this workflow that can be improved with a dedicated visual analytics tool. Existing tools (a) only use two-dimensional slice-based interaction to mark sister cells, (b) do not take advantage of either 3D interaction or interaction with a visual representation of the hierarchy, (c) do not allow the scientists to interactively divide

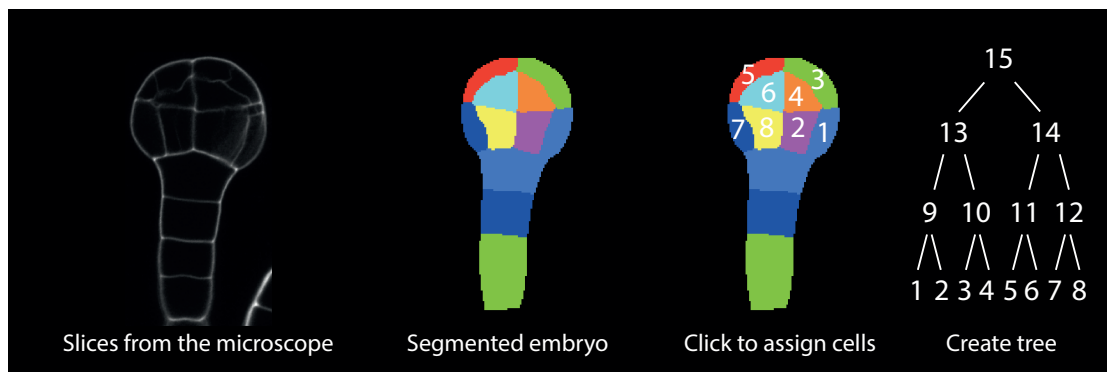


Figure 5.1 – An illustration of the traditional way of biologists doing cell lineage based on slices.

the 3D geometry into parts that can be identified as being part of early hierarchy levels in the cell division, and (d) do not provide any means to automate the cell assignment process based on known cell division patterns (see Figure 2.4). Given these limitations of existing tools, the current, largely manual workflow of biologists does not scale to embryos with hundreds of cells (i. e., 7 or more generations of cell divisions). A purely automatic approach would also not be ideal as they want to retain manual control over the process, to be able to deal with incorrect assignments and the special behavior of mutants.

We thus developed LineageD to address these limitations. LineageD is based on specific domain requirements that we elicited with our collaborator and described in detail later in the paper. Specifically, it was important for our collaborator that the tool would be *interactive* to retain control over the hierarchy construction process, contain the familiar *3D representations* of the embryo state, display a *temporal hierarchy tree* that encodes abstract temporal information about cell divisions, *closely link* the 3D and abstract representations, and include *machine learning* approaches to propose cell lineages and reduce repetitive work. We built the tool based on these requirements, while ensuring that all views are interactive and proposed cell lineages could be efficiently reviewed and interactively corrected if needed. Our goal was for biologists to arrive at a final cell lineage more easily and with higher confidence. To better understand whether our design actually meets these goals, we also conducted an evaluation with five experts from the domain and described their feedback and our observations of their tool use.

Overall, in this work, we contribute (1) an interactive, web-based approach to establishing cell lineage, in which we combine 3D spatial and abstract data representations with correctable machine learning; (2) design considerations and decisions that led to our approach, including about the visual and interaction

mappings that we used; (3) the approach of top-down and bottom-up hierarchy building and the interaction concepts that link the 3D and 2D representations at different scale levels; and (4) the evaluation with five experts to learn about the suitability of the employed approach, which also generated insights for applied visualization at large.

5.2 RELATED WORK

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

5.2.1 Visualization of and with Hierarchies

Hierarchical structures are essential for organizing many types of data and exploring inner relationships [166]. Such inner correlations can be, e. g., about classification [12, 24], family development [20], and evolution [85, 108]. Over the past 40 years, much research has been conducted to design and compare different visual representations (e. g., treemaps, sundown charts, sunburst charts, and icicle plots) of both 2D and 3D tree structures (e. g., [156, 144, 165]), to analyze hierarchical data (e. g., [172, 197]), and to help with visual decision making (e. g., [7]). Work has also been done to evaluate the different representations [209] and to propose guidelines for future design in hierarchical trees [99]. However, most of these 3D trees were designed to add one dimension to store more information [161, 190, 116]. Our collaborating biologists are most familiar with 2D node-linked diagrams for representing the hierarchy, while the actual cell shapes have to be interpreted in 3D views to assist them with making lineage decisions. We thus rely on a combination of both a 3D view and a 2D tree. One of the most closely related approaches is HyperLabels [102], which uses a model's hierarchy to navigate the 3D structures. While Kouřil et al. also combine the abstract hierarchy data with 3D shapes that represent the actual physical entities, their abstract data representations do not represent additional information such as, in our case, the plant embryos' temporal development.

The size of datasets is constantly increasing, so researchers have also explored methods to flexibly adjust the space a tree representation takes to increase its efficiency [113, 27, 3]. For treemaps, for example, Tu et al. [197] used a spiral layout to visualize the updates of the hierarchical data. Similarly, van de Wetering et al. [198] redesigned the icicle plot to reclaim empty space, to avoid the situa-

tions where nodes deeper in the hierarchy have less space. These techniques are not applicable in our scenario because we need to show all nodes equally such that we do not have excess empty space. Moreover, we do not have the full hierarchy at the start, and biologists are not always certain about the cell lineage hierarchy root in the beginning, which is the ovum of an embryo.

Another important aspect of our hierarchies is that they represent temporal development, and researchers have visualized temporal data with trees before. For example, Carvalho et al. [41] combined aspects of treemaps with a calendar, so that users could target data within a certain time slot. In our case, the division time is different for each cell and the former and latter stages should always be presented for reference, so that we cannot use filters to target at a single level. Shen [171] also used index to mark the cell division within a time interval, but it is hard to read especially the tree is large. We thus encode the generation rounds with the layers.

In addition to innovative layout design, some work also focused on improving the interaction with trees to adapt the substructures and compare the results with the original data, which we also need in our work. AdaptiviTree [189] encoded real-time game status information and indicated changes by color. Other work [11, 23] used visual cues to allow viewers to compare different trees that were shown horizontally, with color-coded similarity metrics based on juxtaposed icicle plots. Because we use the trees proposed by ML only as an initial guess and there is no ground truth, we do not show multiple trees and simply rely on color coding to encode differences. Related to this approach is the encoding of the uncertainty of the input data, for which researchers have used, e. g., glyphs [109, 211]. While these methods can handle diverse forms of uncertainty including locations [109] and sub-tree structures [109], we can also use color-coding to indicate the ML model's certainty with respect to its results.

5.2.2 Biology-related Visualization

Biologists often study complex patterns or processes. While various stakeholders have different levels of interest in the respective datasets [108], we focus on interactive tools for experts—specifically botanists interested in plant embryo development.

Within the biology domain, several visualization approaches and tools are currently being used. For example, Briggs et al. [24] combined dendrograms and heatmaps for the visualization of taxonomic diversity. Such representations, however, are usually not space-efficient [128]. Moreover, we need to show both abstract cell lineage data as well as the actual 3D shape of the entities. Numerous platforms (e. g., OsiriX [157], Fiji ImageJ [162], and Avizo) are designed to display such multi-scale 3D data, and these are currently being used by our collabora-

ting experts to establish the cell lineage (e. g., the TreeJ plugin for Fiji ImageJ). Within academic research, Leeuw et al. [43] explored the visualization of time-dependent confocal data. For visualizing inner cell structures, Mindek et al. [134] proposed a multi-pipeline for visualizing the organelle system. Similarly, other work [110, 78, 77] proposed different systems to interact with diverse scales of cell or genome datasets. In contrast to these approaches, however, we are not interested in the representation of spatial scale but in the temporal development in the form of cell lineage. Of course, also general tools exist that could support the representation of 3D shape such as VTK [164] which was used to create visual representations of biological data (e. g., [157, 162]) and ParaView [2]. We also use the VTK library for creating the visual representations in our tool, but our emphasis is on the combination of 3D spatial and abstract data representations and their interactive use for the establishment of cell lineage for plant embryos.

5.2.3 Machine Learning in Visualization

In the visualization field, machine learning is used to create, improve, and assess all kinds of visualizations [61, 206, 4]. Researchers also adopted machine learning models to develop pipelines for the whole process of visualizations [93, 92]. In our case, we use a machine learning model in predicting assignments for a plant embryo and then visualize the predictions in the hierarchy. Unlike interactive machine learning (IML) where users could review and refine the model interactively [52] (i. e., an interactive correction of mislabeled data with the goal of getting a more accurate result from the model [210]), we treat the ML output only as a first suggestion and allow our users to directly interact with the output of the model for a more efficient correction. Also, researchers proposed different suggestions for designing such interfaces [181, 52, 86]. Inspired by these, we predict cell inheritance with a neural network model and display partial assignments to engage users [53] and reduce manual labor.

5.3 BACKGROUND AND DESIGN CONSIDERATIONS

As we just discussed, trees can be used to visualize cell hierarchies in embryos. Our collaborator also confirmed that biologists use and appreciate 3D views to assist them with understanding the spatial structure and to make decisions on the lineage. Here they rely on actual 3D meshes that they derive from the confocal microscopy slice data by segmenting and then processing them into

the shapes of the embryo cells at the time of data capture. Yet, no suitable tools exist to determine cell lineage for several reasons :

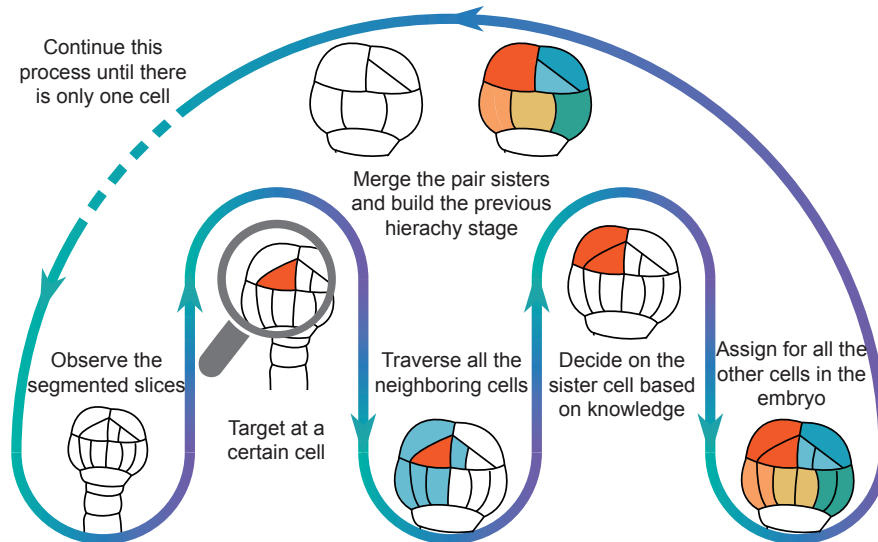


Figure 5.2 – A 2D illustration of biologists' cell lineage workflow.

A cell lineage tree only represents approximate division stages that apply to most cells, while some cells may remain unchanged from a given level to the next and the lineage hierarchy needs to account for such cases. Consequently, there exists no single generic (binary) tree that would represent the development of all embryos. In addition, the tree does not yet exist at the start of the process and cannot be computed from the data, instead, it has to be established step-by-step by the experts based on what they observe in the 3D spatial and abstract data. Figure 5.2 illustrates this process, in which the experts reason based on the embryo's configuration at a given time to establish how it developed into this stage from a previous set of cells, and then they repeat the process. Existing tools are usually created to focus either on 3D spatial or on abstract data representations, while in our case both types of representations are tightly coupled and biologists need both to make decisions (the spatial cell data as well as abstract data such as the hierarchy as established so far and information such as shared area—either type alone is not sufficient).

Based on all these constraints, we set out to design a new interactive tool for establishing and visualizing the development of plant embryos. To reduce the tediousness of having to make many lineage decisions for larger embryos, we targeted a process that relies at least partially on machine learning, yet also leaves the biologists in control and allows them to override decisions as needed. We also based our design on inspirations from existing tree representations

such as icicle plots [209]. Moreover, we followed the principles and advice on designing hierarchy structures [99], cell visualization systems [43], and mixed-initiative user interfaces [86]. Next, we summarize our design considerations.

D1 : Provide an overview of the hierarchical structure to indicate the cell division process. We aim to create complete and accurate cell division hierarchies so that plant biologists can study the division history. Yet every cell divides in a different time and it is impossible to measure the precise division time from a single confocal microscopy dataset. So we rely on the fact that an embryo's cells roughly divide within similar time ranges, and use these division times to build the hierarchy. The tree should also clearly emphasize cells that divide outside of the regular time range. The tree thus not only needs to contain the parent and children relationships but also represents the time-dependent division process. As such, any node in the tree represents a given cell and thus its connected 3D representation at a certain stage, so navigation in the hierarchy needs to lead to adjustments of the 3D visuals.

D2 : Support the history building in two directions. Biologists traditionally build the tree in a bottom-up approach. From the overall shape of the embryo, however, they can also deduct how it divided in the very beginning. Introducing this knowledge early could improve the ML predictions and make it easier for the experts to specify assignments. It should thus be possible for biologists to build the tree from two directions : bottom-up and top-down (Figure 2.3).

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

D4 : Provide assistance for building the hierarchy with machine learning (ML). Our previous work [83] indicated that, even with an interactive system, traversing neighbors to find sisters for more than 200 cells is too time-consuming, and ML-based assignment suggestions could greatly improve the workflow. These predictions then need to be checked by the experts with the help of an integrated hierarchy and spatial 3D interaction. Any mistake, however, would invalidate any prediction for earlier cell divisions. The system should thus use an ML model that predicts the most likely cell pairs for a given hierarchy level, and bases future levels on verified or corrected assignments. The full po-

wer of our approach relies on a fluid change between the bottom-up ML-based prediction, expert correction, and the top-down hierarchy specification so that the expert is always in full control over the process. At any time, the hierarchy view should show the status of the lineage, indicate decisions confirmed from the ML or corrected by the human, and navigate the temporal development of the embryo.

D5 : Provide easy access to biologists who are not computers professionals. Many experts may be hesitant or not allowed to install extra software on their workstations. Moreover, such local installations would require cross-platform system development. We thus aim for a web-based implementation that facilitates easy access, interaction, and data sharing among researchers [110].

5.4 SYSTEM DESIGN

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

5.4.1 Front-end Interface

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

Our LineageD system (Figure 5.3) centers around two main views : the **main 3D view** of the embryo and the **hierarchy view** that shows the cell lineage tree the biologists are building. The hierarchy on the right of the interface initially consists just of small blocks, where each represents a single cell at the time when the confocal microscopy dataset was captured; i. e., the leaf nodes of the tree (D1). We chose a tree layout that grows from top to bottom (i. e., root at the top) to match the traditional format used by the biologists (we illustrate the design process in Figure 5.4). As the experts work on establishing the hierarchy and declaring sister cells, this view actually becomes a hierarchical representation that we based on (completely filled) icicle plots. Unlike node-linked diagrams that

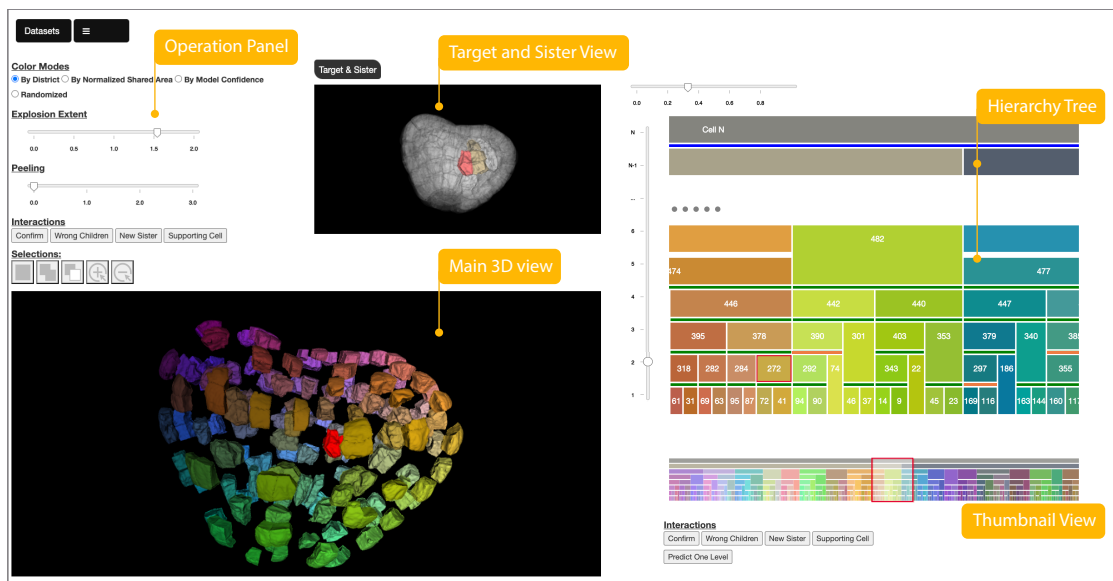


Figure 5.3 – Screenshot of LineageD, for the example of assigning a 256-cell embryo (in progress). The core elements are the **Main 3D View** of the embryo, here exploded and with the target cell highlighted, and the **Hierarchy Tree** of the lineage, which is interactively established by the biologists. The **Target and Sister View** shows the relative position of the target cell within the whole embryo and its proposed sister. The **Thumbnail View** of the tree provides the context of the full hierarchy. Finally, the **Operation Panel** supports further control of the tool.

take more space and are harder to navigate in a wide tree, this mapping has the benefit of establishing clear layers, which can then be navigated with the help of the slider on the left of the hierarchy view. We compute the widths of the leaf nodes based on their normalized volumes such that the biologists can directly see the relative sizes of the cells' volumes in the tree (D3). The icicle mapping then also nicely reflects that the volume of a mother cell is roughly the sum of the volumes of its daughters in consideration that the potential volume loss is not essential to biologists, and that all cells of a given level together always form the whole embryo. We chose the vertical direction for the hierarchy layering because the biologists are used to this layout.

The hierarchy view is tightly linked to the main 3D view on the left. In the latter, we show the whole embryo at the chosen division time, i. e., the level selected in the (partial) hierarchy view on the left. A regular 3D representation has the problem of the dense packing of the embryo cells, which results in inner cells being fully occluded by the outermost cell layer. To address this visibility issue we offer two visibility management techniques : explosion views (e. g., [114, 191]; see Figure 5.3–main 3D view) and layer peeling (e. g., [184]; see Figure 5.5–3),

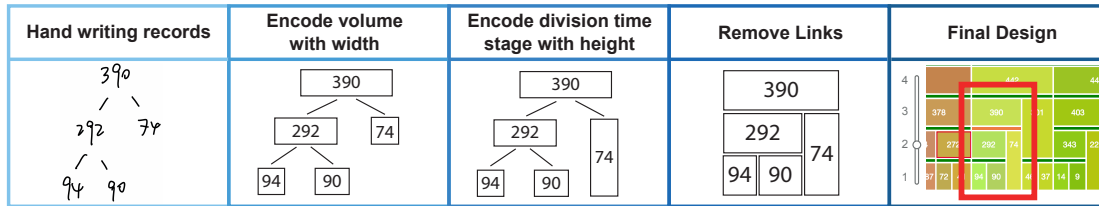


Figure 5.4 – An illustration of our reasoning for how we designed the lineage hierarchy tree representation, based on the traditional hand-written or (ASCII) text-based records used by the biologists.

both controllable through sliders. The 3D view also provides typical 3D navigation interaction, including zoom, translation, and rotation to allow biologists to explore the spatial character of the embryo.

We closely coupled both the 3D view and the hierarchy, not only through cross-linked selections. When we select a given cell in either view, we not only select it in the other as well but also show this cell and its currently assigned sister (if any) in a secondary 3D view on the top, the **target and sister view** that shares the same camera orientation as the main view. Here we highlight the selected (target) cell in red and its assigned sister in its original color, while we show the rest of the embryo's cells transparently to illustrate the target and sister cell' relative positions in context. Unlike in the main view, in the target and sister view we always show all cells in a tightly packed configuration and do not apply any explosion. This way we allow the biologists to understand the combined shape of the proposed cell match and make it easy for them to identify incorrect assignments. We specifically decided to only provide a single target and sister view. Initially we considered showing just the target and sister cell, a view of the target cell's children without context, as well as a view of all direct neighbors of the target cell. In a pilot study, however, we found that these views were confusing to our collaborator, and we settled on only showing the target and sister view as a secondary 3D view and moved some of the other functionality to be accessible on demand (see below).

We use the same color mapping in all views, both the 3D views and the hierarchy. We initially color-code both views by spatial location [146], which we calculate based on the relative 3D positions of each cell's center of mass within the whole embryo, mapped to RGB. This mapping allows the users to clearly see if two cells are located close to each other, even in the abstract hierarchy view. During the later interaction we also offer color-coding by trust of the ML model, by the shared area between two matched cells, and a random color mapping that biologists are used to from their existing tools and that allows them to easily see the decomposition of the embryo into the cells of a given hierarchy level (D3).

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

Due to the exponential nature of cell division, the corresponding cell lineage tree can become very large horizontally very quickly, already for embryos with only a few generations of development. We thus provided two means to cope with the resulting view limitations. First, we provide a slider on the top of the hierarchy to control the horizontal zoom, allowing the users to see the whole tree in one view or to zoom into details. For the latter case we can ensure that the respective labels of the cells (numbers in our case) are visible, yet a viewer may lose the context of what part of the hierarchy they are examining. We thus also added a horizontal World-in-Miniature **thumbnail view** to make it easier to navigate the wide icicle plot hierarchy tree and which is often used in virtual reality (e. g., [180, 39]) and, more relevant to our case, in 2D video games (then often called “mini-map”; e. g., [71]). This view, which we include below the hierarchy view, always shows the whole hierarchy and highlights the subsection that is currently visible in the detailed hierarchy view and facilitates navigation.

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

5.4.2 Server-side Back-end

Our tool’s server-side takes care of data processing and analysis. Our datasets comprise the cells’ names (i. e., numeric labels) and mesh data for their surfaces. The experts create these datasets with dedicated tools (e. g., Avizo, Fiji [162]). In the back-end we use this input to build the necessary data structures to later be able to construct the hierarchy. We also derive the needed information

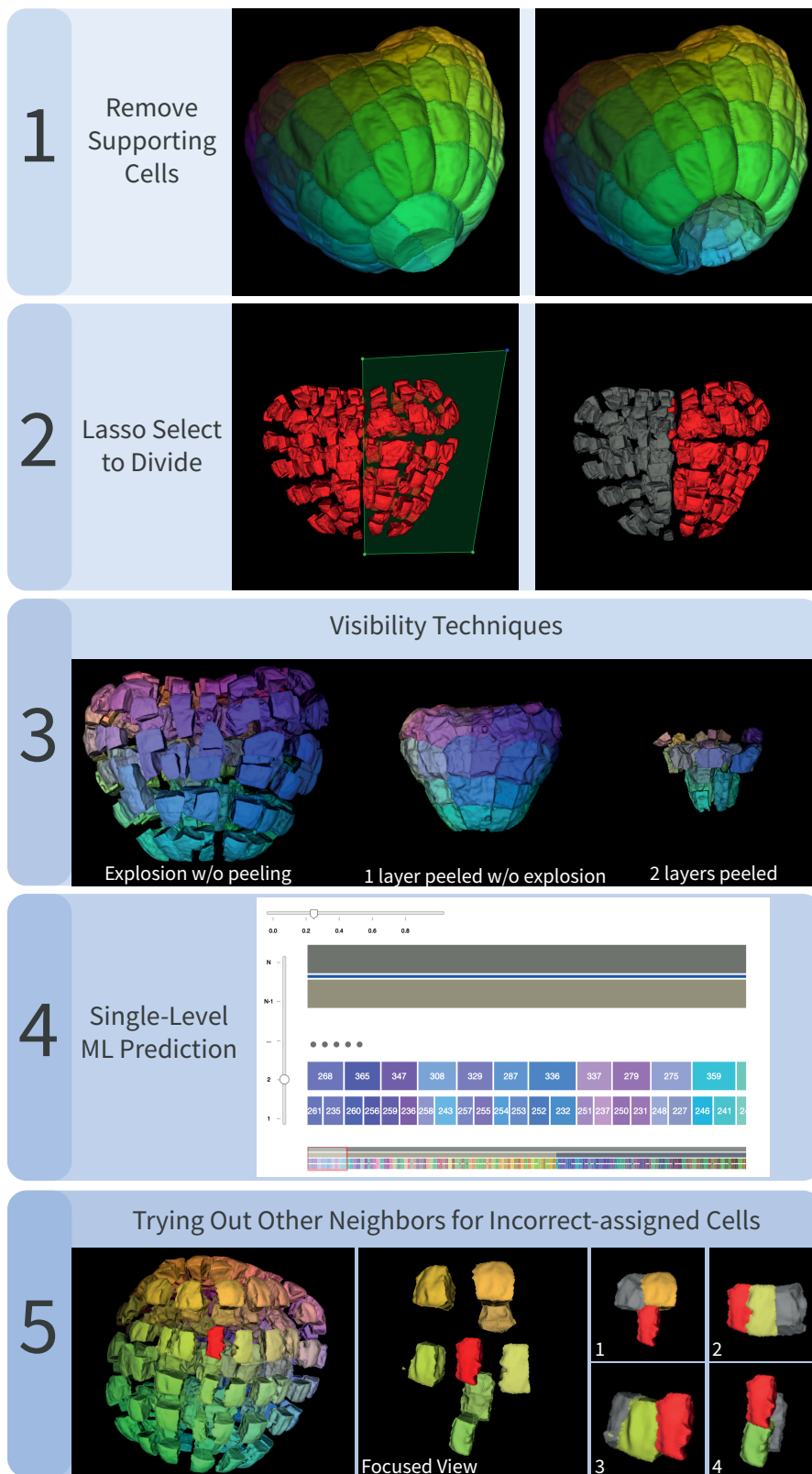


Figure 5.5 – Several interaction techniques in LineageD.

such as the 3D cell locations (centers), neighbor counts and respective neighbor look-up tables, normalized cell volumes, normalized cell surfaces, and normalized shared areas between neighboring cells. After the completion of the lineage, the back-end also allows us to export the results in the tree data format needed by the biologists.

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

5.4.3 Interaction Design

Our interaction design centers around the processes of proposing new matches of likely sister cells (manually and with the help of our neural network model) and the verification of whether these matches are correct. Both processes rely on abstract as well as 3D information, so our expert users have to constantly switch back and forth between the abstract hierarchy view and the spatial 3D views. Next, we describe how, in LineageD, we support the biologists both in the top-down and the bottom-up lineage specification.

For a new dataset we show all cells as blocks on the lowest hierarchy level on the right, and in the 3D view we show the embryo with the cells colored based on their 3D location. Usually the first step of the biologists is to remove those cells from the consideration that are part of the *suspensor* (i. e., those that connect the plant embryo with the rest of the seed) by selecting them in the 3D or hierarchy view and assigning them a respective label (Figure 5.5-1).

With the top-down specification we offer a new functionality not supported by the traditional tools. We take advantage of the fact that, for most embryos, the first few cell division generations happen in a predictable way—one that can still be reconstructed from the geometric shape of the embryo and its cell decomposition, even at a later time. For this purpose biologists start to explore the shape of the embryo in the 3D view, usually via rotations. To better see and understand the inner structure, they can also use the explosion functionality. Then they decide on the most likely initial division—often a rather straight subdivision between the existing cells. We then allow them to mark one half using a spatial lasso selection [122] (Figure 5.5-2) to register the selection as the first

subdivision, which can also be further refined.

As they continue this top-down process, it is important to note that the interaction naturally relies on the selected level in the hierarchy. For example, to specify the third generation $N - 2$ (i. e., the second split), the experts first select one of the daughters of the *ovum* in the hierarchy. This highlights all parts of the embryo that belong to this cell at the $N - 1$ level in red, as if this was a single cell. Any potential explosion setting, however, applies to the level *currently* specified in the hierarchy just below; at the beginning this would be level 1.¹ The following lasso operation on a subset of these cells then selects them for the next split of only the selected cell, in a context-aware fashion [216]. This step would then have to be repeated for splitting the second cell of the $N - 1$ level. To mark these fully human-decided divisions, we display blue bars between the divided parent cells and their children in the hierarchy tree Figure 5.3. Naturally, this top-down process only works for a few generations, but it provides valuable constraints both for a manual and for the ML-supported process that we describe next.

For the bottom-up process we follow the biologists' traditional work process, as illustrated in Figure 5.2. They would start to traverse all cells at the lowest level and specify likely matches by examining each selected cell's direct neighbors to find its most likely sister. To reduce this workload, we can now use our neural network model and make ML-based predictions for the lineage. Initially we had used this process to predict the entire lineage, but in pilot studies it quickly turned out that this approach was not ideal. The experts disagreed with certain mappings, rendering the entire remainder of the path toward the root node obsolete when they corrected the mistakes. We thus now restrict the classifier to only predict the lowest non-specified hierarchy level (Figure 5.5-4).

After such a ML prediction step, the biologists can traverse the newly found pairings and either confirm or reject them (D4). To do that they can select a daughter cell in the hierarchy or 3D view, which shows it in a red highlight color, with the rest of the cells retaining their colors according to the chosen color scale (Figure 5.5-5-left). In the target and sister view, we also show its currently assigned sister in color, while the rest is transparent. A single click on a mother cell² is similar, only then we highlight the whole mother in red and show no sister. With these techniques the experts can assess the mother cell's shape and decide if it is plausible. To better understand the alternatives, the experts can also double-click in either the hierarchy or the 3D view on a daughter cell, which leads to the 3D view showing a focused view with only the direct neighborhood

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

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

of the selected cell (Figure 5.5-5-center), while the target and sister view then only shows the cell and its currently specified sister. Upon a further click on one of the neighbors in the main 3D view we show this neighbor also in the target and sister view, along with a potentially assigned sister of the neighbor (Figure 5.5-5-right).

With these different mechanisms to see detailed views, the biologists then make a decision about whether the pairing is plausible or not. If the biologist then confirms a proposed mapping when a given mother cell is selected, we mark that pairing as confirmed by showing a green bar between the mother cell and its daughters in the hierarchy (Figure 5.3-right). If they disagree with the pairing, we break it and, for the time being, treat the cells as non-dividing cells for the chosen hierarchy transition. They can be left as such or can be re-assigned. In the latter case, biologists select one cell, show its neighborhood with a double-click, and then pick one of the neighbors to mark it as the new sister. To distinguish such human specifications from confirmed ML pairs, we mark them with an orange bar in the hierarchy between the mother and its daughters.

After examining all pairs of a given level and confirming or correcting them, the experts can then go on to predict the next level. We treat any pairing that has not been explicitly confirmed as implicitly confirmed in that case, and also mark it with a green bar. The process then continues until the whole hierarchy is specified, the top and bottom trees merge, and we can export the result. The detailed interaction process is illustrated in Figure 5.6.

5.4.4 Implementation Details

We implemented LineageD in *JavaScript* and *Express.js* (v. 4.17.1). For the front-end, we rely on *D3* [21] to realize the hierarchy tree rendering and render the embryo in the 3D views with *vtk.js* (v. 19.0.4) [163]. We implemented the back-end in *Node.js* and used *MongoDB* to create our database. We used *TensorFlow's* (v. 2.8.0) *Keras* library to train the multi-layer sequential neural network model offline with *Python*. We created two hidden layers with *ReLU* activation and adopted *softmax* for the final output layer to get the probability score, and the model's single-pair accuracy reaches approximately 94%. The raw data is produced by *Avizo* and contains the cells' information including names, vertices, and triangles. We then parsed the raw shape data and set each neighboring pair as one record. For each pair, we extracted features for the training data based on the recommendations by the biologists : the normalized distance between the cell pair, their normalized shared surface area, their neighbor counts, their normalized volumes, normalized surface area, and the directions of neighbors with target cells. With *TensorFlow.js* (v. 3.8.0), we used the trained model to predict assignments in the website back-end. For an embryo with 256 cells, such as one

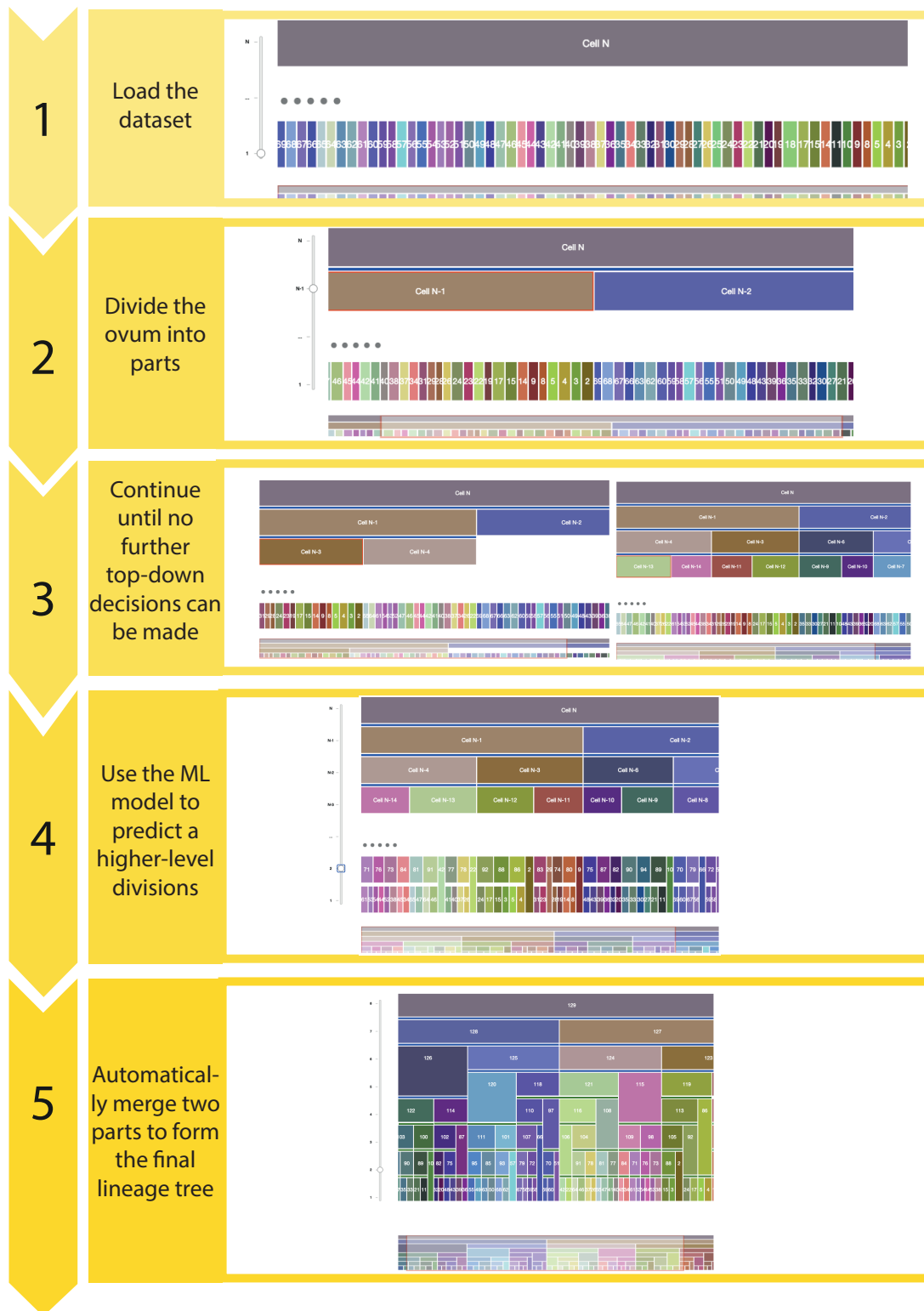


Figure 5.6 – Illustration of the typical interaction workflow of LineageD.

of our default datasets that we show in Figure 5.3, there are more than one million vertices with more than five million triangles. Depending on the network speed when connecting to the Web tool, typical datasets can thus be loaded in less than a minute and can be rendered at interactive speeds (approx. 20 fps for the 256-cell example on a macOS with AMD Radeon Pro and Intel UHD Graphics 630).

5.5 EMPIRICAL CASE STUDY EVALUATION

While we developed our overall approach and our LineageD tool based on conversations with our expert collaborator, we were also interested in feedback from independent plant biologists, by whom we envisioned our tool to be used. For this purpose it does not make much sense to run a quantitative experiment with many participants because, first, we do not have a reasonable alternative technique to which to compare our approach and, second, we would not find enough participants skilled in the traditional lineage process to draw meaningful conclusions. Instead we conducted case study evaluations [96] with expert participants, as it has been done in the past in the visualization field (e. g., [123, 97, 88, 205]). The benefit of this approach is that experts are trained in the traditional cell lineage process and can thus assess our new tool based on their experience.

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

5.5.1 Participants

We recruited 5 expert biologists (2 females, 3 males, ages 31–54 years, mean 43.4 years) via social networking and mailing lists, denoted as P1–P5. We offered prospective biologists free access to our tool as compensation. Before the study, we asked them to fill in a consent form, a research media records release

form, and a background and demographic data collection form. All of the participants had a PhD degree and have been doing post-PhD research on plant biology or related fields for 2.5–25 years (mean : 14.1 years, sd : 9.4 years, median : 12 years). Two of them were experts in plant biology, while the other ones focus on computational biology, biomathematics, or bio-images processing. We conducted our study via videoconferencing due to the COVID19 limitations.

5.5.2 Procedure

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

Observational study. We conducted the experiment with each participant individually. We began by presenting a brief introduction about the features of LineageD. Next, we asked the participant to open the website on Chrome using the credentials we provided and to share their screen. We then briefly explained to them the components of LineageD and introduced the main interaction mechanisms, as explained in section 5.4. Then the participant could freely explore and establish the lineage for the 64-cell dataset. We used the think-aloud protocol and with the participant’s permission, we recorded the screen and audio during the process for our later analysis.

Survey and semi-structured interview. To better understand the usability of the tool and avenues for future improvement, we distributed a questionnaire which was designed based on the System Usability Scale (SUS) [80]. We also conducted a semi-structured post-study interview to understand our expert participants’ usage experience and insights they may have gained.

5.5.3 Study results

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

Learning the Functionality. We found that three of our participating experts traditionally used 2D slice-based interaction (i. e., in TreeJ) to navigate the embryo dataset and to mark sister cells, as opposed to using 3D interaction with a surface-based dataset that we offer. This means that participants first had to get used to the fundamental 3D interaction techniques (rotation, translation,

scaling) in projected 3D views. They also traditionally used text-based node-link representations of the hierarchy (see Figure 2.7), so they had to get used to the icicle plot representation we used. Web-based implementations as we used for LineageD were also new to one participant : P1 thought that the tool is quite different from the traditional software he is using. He stated that they generally use desktop software, but he believes that with sufficient time to play with it, most people could learn the new tool quickly. A final aspect was that P3 said that it was difficult for her to learn the actions to expect from the system based on specific input, such as what part of the tree to which the prediction applied. Overall, these observations meant that the training phase in our experiment was relatively long for four out of the five participants, but we expect that with more training, they can get used to the new interaction concepts.

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

Interface and Visuals. All participants stated that they appreciated the visual representations, both 3D and 2D. They also liked, for example, the mechanisms for encoding the cell volumes in block widths and the coloring by shared surfaces or location. Overall, P1 considered LineageD much more powerful than any tool that they currently have, especially in selecting cells from both 3D environments and the tree. P2 specifically appreciated the target and sister view that shows the proposed pairing as well as the transparent context. He liked

the representations and interaction design of the tree, but also thought it may take time for biologists because they are used to node-link trees. He also added that some other functionality may need to be adjusted further to the mental concepts of the biologists, but thinks this is a normal process that he would expect for most tools. Overall, all comments about the design were on the positive side. For instance, P3 thought that all individual components are needed and easy to use and she liked the hover effects in the hierarchy tree. P4 was happy to see that the tool is versatile, because he could use peeling and explosion independently. P5, finally, liked the hierarchy tree and thought all the options are good.

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

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

Improvements. We asked the experts what interactions or elements they

missed in the interface. P1 and P4 would like to have functions to label or categorize the cells so that they would have more features to examine and analyze the situation. P1 and P3 also missed a function to temporarily hide selected cells to get access to the inner cells and to distinguish checked pairs from unchecked ones. Also, P3 wanted the ability to explicitly highlight the shared surface between two neighboring cells to help her to decide between two possible cell pairings. The System Usability Scale rating we received for LineageD was 68 on average, which is exactly the average SUS score [129]. This means that we need to improve the overall usability of our tool, but this is to be expected for a research prototype. We will use the specific feedback that we received during the study to make these improvements.

5.6 DISCUSSION

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

Influence of past interaction habits of experts. Our specific application domain required us to provide both abstract 2D and spatial 3D representations : the cell lineage hierarchy that the experts constructed along with data such as cell volumes or area of shared surfaces on the one side and the actual shapes of the 3D cells that make up the embryo on the other side. Interestingly, it turned out that most experts were not yet experienced with working with (projected) 3D representations, instead they used 2D slices from the segmented microscopy data to interactively mark cells. In addition, traditionally they do not use the hierarchy tree as an interaction proxy, and it only serves as the result of the lineage specification (also see Figure 2.4). This fact may be the reason that many of our experts were reluctant to work directly in the 3D view and, instead, primarily interacted with the abstract hierarchy as this somewhat resembled a version of their previous slice-based view (yet stripped from its spatial properties). More longitudinal studies will be needed to see if experts can learn the benefits of interacting in 3D space, and to take full advantage of the linked 2D and 3D views. Our observations, however, make this a likely development as some of our participants already understood the advantages of our 3D visibility techniques such as explosion and peeling.

Reliance on both 2D and 3D representations. Our specific application exemplifies a case where both 3D spatial and abstract information and, in particular, the tight connection between both is essential to solve a task. Nonetheless, both types of representations show different aspects. Only the tree view can show inheritance and, thus, only here can the biologists specify which cell division time

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

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

Generalization. Even though we focused on plant embryos, there is no reason that our findings could not also apply in a broader scope. In particular, biologists that study other kingdoms of life such as animals or fungi may similarly profit from our work directly. The most closely related past work on the depiction of parts may be research on the creation of assembly instructions [1, 114, 191, 76], even though the actual interactive control of such depiction was typically not discussed in much detail in such work. Other past work [102] specifically discusses the navigation of 3D data through hierarchy interaction, yet in our work we actually change the composition of the 3D scene through this input. We thus think that our interactive combination of abstract hierarchy and spatial 3D views can inform future visualization tools that need to support such inter-dependencies.

5.7 LIMITATIONS AND FUTURE WORK

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

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

5.8 CONCLUSION

Having developed LineageD together with a domain collaborator, we were surprised to see that most domain experts did not have even basic visual representations such as tree representations [165] as interactive elements in their traditional tools. A similar situation happens in the top-down and bottom-up approach where biologists spend reasonable amount of time to get familiar with the interaction. Moreover, when we provided them with this tool along with a standard 3D representation of their data, they primarily interacted in the 2D abstracted hierarchy as opposed to the 3D spatial view that showed the relevant data (cells within the context of the embryo)—almost like an extension of their previous 2D interaction on segmentation slices. So one of the main results of our work seems to be that, as a community, we may need to spend more effort on studying how established representations (from our perspective) can actually be transferred to tools in our application domains. We plan to continue our work with the biologists to do just that, and to also see how LineageD can evolve over a longer time to better suit the needs of the biologists.

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

COMPARATIVE VISUALIZATION OF MACHINE LEARNING PREDICTIONS TO IMPROVE HUMAN-AI TEAMING ON THE EXAMPLE OF CELL LINEAGE

6

As we found in the evaluation study of chapitre 5, a single ML model may make mistakes, which could result in biologists losing confidence in ML. We then planned to create multiple ML models and combine their prediction results to provide comparably reasonable predictions. Such design requires visualization to clearly demonstrate the detailed ML predictions from different models and also an overview of all predictions. Thus, based on LineageD, we created LineageD+ to enhance the ML support with visualization in solving the cell lineage problem and tried to answer **RQ4** from chapitre 1.

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 LineageD+ can effectively improve the assigning efficiency, reducing the time it takes to assign cell lineage and improving biologists' confidence in the ML models.

6.1 INTRODUCTION

In biology, a plant cell (the *parent*) normally divides into two daughter (or *sister*) cells over time, and an embryo grows to eventually comprise hundreds of cells. To explore the history of an embryo's development, biologists utilize a 3D microscopy snapshot and assign sister relationships for every cell in the embryo. This is done backward across a series of snapshots, where biologists iteratively reason backward in time to arrive at the previous cell division stage. 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. Different ML classifiers, however, have different performances, so some classifiers may provide the correct prediction for a given pair while others may not. Unfortunately, there is no guarantee that the same model will perform correctly (or incorrectly) for another cell pair at another division stage. Ideally, the biologist should train multiple models and explore 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 [188, 118]. Visualization tools have been developed to illustrate all steps of the machine learning pipeline, including data processing, training, and evaluation (e. g., [158]). 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 enhancing this *human-AI Teaming*¹ 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 more efficiently derive the correct lineage.

We collaborated with plant biologists, collected 93 embryo datasets 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 6.1(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 for biologists (Figure 6.1(b, c)). We sorted the hierarchical tree by the models' overall confidence and support similarity pattern detection for target cells such that 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 the user-adjust model weight 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 the following : (1) a visualization system to assist biologists in effectively establishing cell lineage through human-AI teaming ; (2) a novel visual representation to compare multiple ML models and help users detect possible model mistakes for improved decision-making ; and (3) operations to predict cell sisters, assess the prediction results, and interactively revise the assignments.

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 see section 6.6.

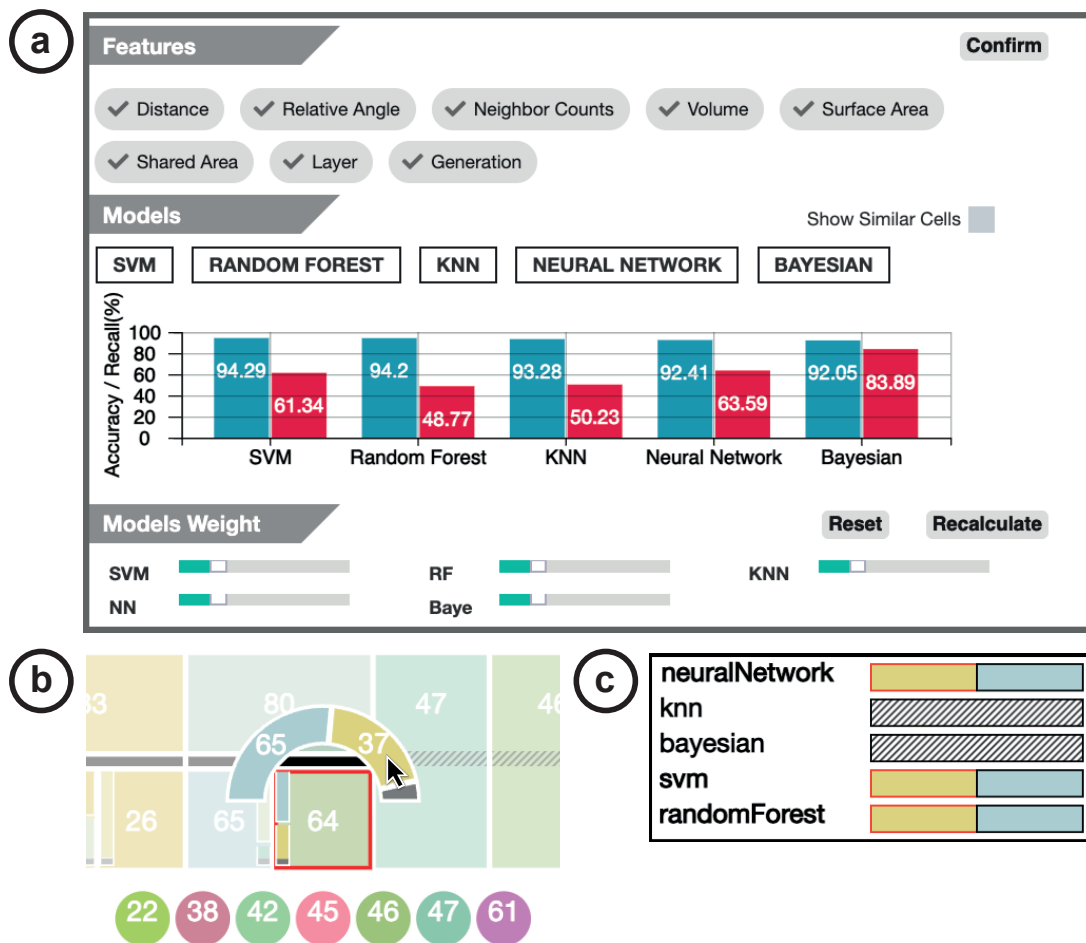


Figure 6.1 - The visualization design for our multiple machine learning models. (a) The overview of the model performance. Users can customize the features to train models. Also, they can change the model weights via the panel. (b) The detailed prediction of a specific cell (cell 64 as an example in this figure). (c) The model results of the target cell (cell 64) and the interested proposed sister cell (cell 37) being sisters.

6.2 RELATED WORK

In our work we deal with improving the interaction of domain experts with multiple machine learning models rather than training better models. In this section, we summarize the relevant literature with respect to the visualization of hierarchical information in biology, cell lineage reconstruction in biology, and the visualization of ML model output.

6.2.1 Visualization of Hierarchical Information in Biology

Numerous general visualization approaches focus on building different kinds of hierarchy representations [166, 165]. Besides showing the necessary hierarchical information, biologists usually need to add additional information to the current tree design, e. g., time [55], relative object sizes [84], etc. In our scenario we also need to add the cells' volumes and model predictions for each cell in the tree. Previous work has dealt with similar requirements, and researchers have created various dedicated visual representations for bioinformatics data. Eisen et al. [55], for example, designed a visual representation with colored nodes to represent DNA data that relies on a basic cluster dendrogram with additional temporal data, which resulted in a representation that resembles a heatmap. Based on this design, the Hierarchical Clustering Explorer [169] 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 such as bar charts.

Along with 2D representations, researchers also investigated approaches to visualize biological data using 3D views. Arena3D [148], for instance, presents network data both on the 2D graph and in the 3D space. This combination helps users in solving the overlapping and intersection problems that are inherent to 3D views when the entities reach the thousands. We also use such a combination of 2D and 3D views, but, 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. We employ a traditional 2D icicle plot design for visualizing our hierarchical tree and combine this with a 3D view of the actual embryo shape and its cells that can be viewed for any given hierarchy level.

6.2.2 Cell Lineage Reconstruction

This visualization of hierarchies can help with predicting the possible developments of specific cells [75] in biology. In Salvador et al.'s [159] 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 that biologists can target cells from either 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 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 [157], Fiji ImageJ [162], and the TreeJ plugin for Fiji, ImageJ, but these tools typically rely on 2D slices. Because the hierarchy resulted from numerous cell divisions and each cell division orientation is decided by the position and angle of mitotic spindle [111], researchers have explored diverse machine learning models to predict the division planes [133, 124]. These prediction methods, in the past, have been based on the microscopy slices [133]; 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 our ML training.

6.2.3 Visualization of ML Model Output

Such ML prediction processes can also be assisted by visualization. For instance, researchers have enhanced the interpretability of ML models [28, 158], with problems ranging from clustering [104] to classification [188]. Visual support can assist practitioners to better understand where, why, and how ML models make predictions [158]. Such work involves visualizing the steps of model generation as well as the actual prediction, including data preparation [183], model training [117, 188, 118], results evaluation [37], and model comparison [104].

In our case, however, biologists are most interested in the final results of the model predictions—the final cell lineage hierarchy—rather than the ML model generation. Thus, we focus on visualization of the model output. Prior work in visualizing model output focus 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 [104], scatter plots to show classification results [89], and histograms for visualizing perplexity [149]. Inspired by these representations, we adopted the hierarchical tree to represent an embryo's development history and we enhance this with a variety of view to show the predicted cell lineage. Prior work in interactive machine learning uses people's feedback to re-train the ML model to get better results [176] or to pick the optimal model among multiple models [183]—approaches that we do not adopt. EMA proposed by Cashman et al. [31], for instance, is a process that asks people to explore different models and to select their preferred ones. Also, Gil et al. [72] proposed human-guided machine learning (HGML) to encourage domain experts to fully make use of relevant knowledge in getting a model with high quality, even if they have no experience in ML. These examples provided us with good examples of how to help biologists to understand how models perform and how to enhance their overall experience : we need to visualize the properties and performance of the different models. However, in our work, we focus on the comparison bet-

ween models, rather than the selection of a single model or on improving models, as there is a limited amount of manually assigned datasets to train the cell lineage models. As such, model optimization is not possible at this stage, partially because there may even be disagreements between experts on the correct lineage.

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. [48] showed that people will likely trust models more if the system allows them to disagree with some of the predictions. Such trust could establish people's beliefs and their willingness to use the system and complete specific tasks [174]. When interacting with multiple models, the examination of various predictions and being able to decide on which model to trust are also types of interaction to control model predictions. Thus, it is essential for biologists to interact with the prediction results, especially when they have no experience in dealing with the ML models. As previously stated, it is impossible to produce a single ideal model for cell lineage classification. To overcome this, we produce multiple machine learning models for biologists to compare. Interaction methods usually include the use of a pre-designed control panel [169] and the direct interaction with the visualization itself [148]. Different from the aforementioned work, our tool provides linked views of both the 3D environment and 2D panel, both of which are needed to determine cell lineage and assess the ML decisions.

6.3 SYSTEM DESIGN

Based on this background, we set out to support our collaborators 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.

6.3.1 Design Considerations

Based on discussions with our collaborator in biology, we set a number of design goals for the new system. First, we want to take the biologists' **traditional work process** into account and support them in getting fully involved. Biologists usually do not have experience in dealing with machine learning models, and they do not 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, en-

hancing their control over the models, such as selecting customized features, could help us to increase their involvement and establish usage confidence. As such, they need intuitive and efficient interactions to detect wrongly-paired cells and correct them if necessary.

As indicated in sous-section 6.2.3, 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. Furthermore, even experts can have difficulties deciding the right sister for a specific cell, and any given manual lineage solution is not necessarily unique. While, in the future, more research into ML support for cell lineage may produce such an optimal model, our focus is on training **multiple models** such that 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.

Additionally, the fact that the cell lineage is deducted backward in time from a single stage of the plant embryo's development means that the process is **inherently uncertain**. Also, 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 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 the biologists to understand the uncertainty.

Our final major consideration is that we need to give biologists **guidance on the variety** of ML predictions at a local level for each proposed pairing. It does not make sense to show them one hierarchical tree per ML model as this would 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 of the ML predictions should augment the global comparison of the different models.

6.3.2 ML Features, Model Training, and Prediction

To determine the features to use for ML, we asked our collaborators about which properties they refer to when they decide on which cells they assign as

sisters. We also referred to universal division rules as described in the literature [150, 18]. Finally, we decided to take 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 these 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 collaborators. As we show schematically in Figure 6.2, 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 6.2(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. The reason for this layer marking is that there are, in fact, two 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 columns of cells. Younger embryos usually divide periclinally, while anticlinal division usually produces cells with new functionality. We capture this feature through the layer property, along with the *eighth property*, the generation information.

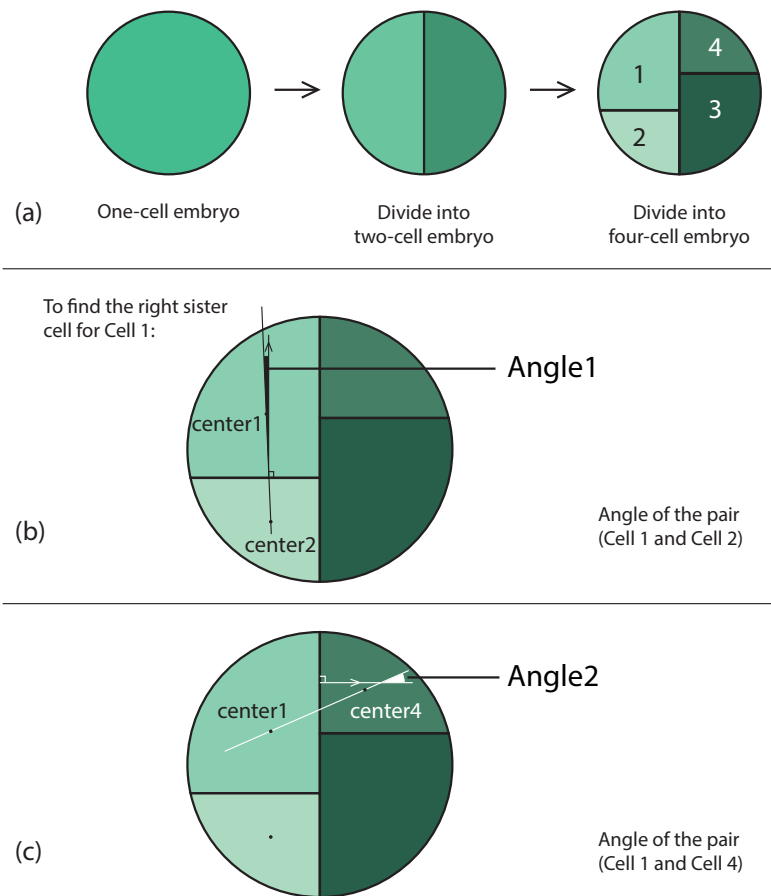


Figure 6.2 – Illustration of (a) the first two stages of cell division and (b, c) the angles we used in our ML training to identify likely sisters (illustrated in 2D for clarity). Here, *angle2* is bigger than *angle1*, which means that *cell2* is more likely to be a sister of *cell1*, compared to *cell4*.

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. LineageD supports 5 different machine learning techniques : support-vector-machines (SVM), random forest [22], 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 could directly re-train the model locally without setting up the corresponding environment. Models from other families, such as XGBoost [35], 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 [139]. 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 shown in Tableau 6.1.

Table 6.1 – 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

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. Then, for each positive prediction, we 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). Thus, for each proposed pair, we can 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 that at least two models agree on. 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 higher probability. In this way every cell can be marked with a comparatively most likely sister.

2. We only predict one level at a time as we rely on user feedback to correct this prediction, as a wrong prediction would invalidate any further hierarchy levels [84].

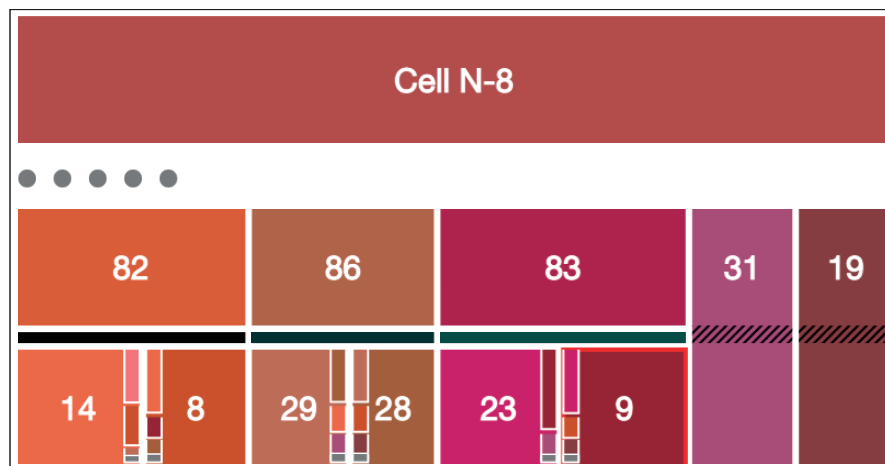


Figure 6.3 – For each quarter (here we take the quarter Cell N-8 as an example), we predicted the new level with the quarter constraints and sorted the predictions based on the overall confidence of models from left to right. As the image shows, the left-most pair has multiple possible sisters for both affected cells. The position of a cell in the subtree 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 subtree 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 a greenish color. For the cell 83, the ML models are 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.

6.3.3 Visualization Design

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

Visualization of ML prediction data

The multi-model prediction process described in sous-section 6.3.2 allows us to produce a single (partial, bottom-up) hierarchy similar to the previous one-model approach [84]—which can be shown to experts in an icicle plot. Our multi-model prediction, however, provides users with more information about each pairing and the overall model parameters.

We summarize the model predictions from the five models and visualized them in stacked bar charts that shows the prediction for each cell (Figure 6.3), adjusted by model weight. We sort the bar charts descending from top to bottom

so viewers can focus more on the most likely predictions. In Figure 6.3, the vertical 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 pairing) to allow biologists to directly compare the individual predictions of any given pair. For those 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).

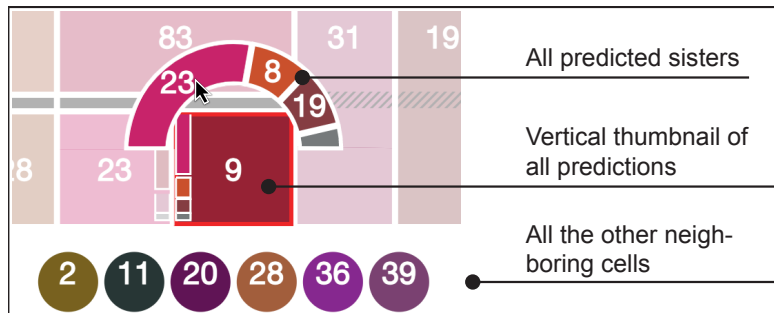


Figure 6.4 – The details of model predictions for one cell. Here we take cell 9 as the target cell for example. From the colors, we can also tell the relative distances between two cells. The percentage of each model depends on its accuracy rate and the customized model weight. The colors of cells in 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.

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 6.4), inspired by previous work on necklace maps [175], 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



Figure 6.5 – 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.

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 the overall predictions from five models (Figure 6.5). To display which models proposed a specific sister cell, we added another pop-up view to the current interface (Figure 6.6), which we show when the biologists hover over one of the bar in the half-donut pie chart. In the example in Figure 6.6 which shows the popup upon hovering on the cell 9 curved bar in Figure 6.4, 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. In Figure 6.7 we summarize the connection between the different detail levels and how we derive the necessary proportions that we show.

Support for detecting possible errors

It is also important to provide guides for 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 subtree 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 to 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 subtrees in order 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

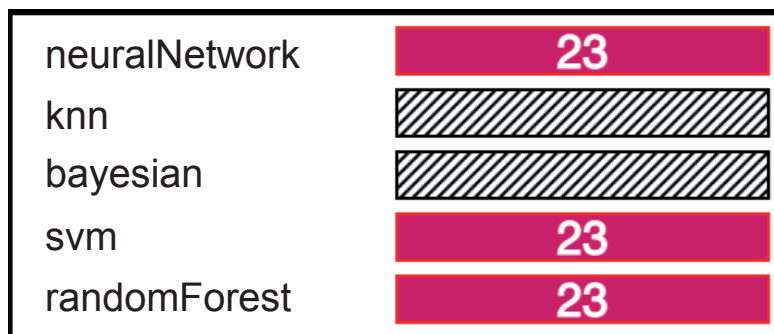


Figure 6.6 – When hovering over a specific proposed sister cell in the semi-donut chart (as we did in Figure 6.4, where the mouse hovers over the proposed sister 23), we show the detailed view on the chosen pairing as opposed to showing all predictions as before. Different from Figure 6.5, here we only 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.

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 [8], which we calculate for all possible pairs of mother cells of a given level. Then, as one target (mother) cell is highlighted (e. g., in the view in Figure 6.3), in the 3D view, we show those other mother cells similar to the target if its Tanimoto similarity is larger than 25%. We also color-code the similarity with a sequential color scale (based on magenta (see Figure 6.8)). Then the biologists can review this limited range of possibly wrongly predicted pairs.

6.3.4 Further Interaction Design and Decision Making

We based our new interaction design on our previous tool LineageD [84] (see Figure 6.9) and added ML model interactions to help the biologists to understand the decisions of the different models and to make potential adjustments (as illustrated in Figure 6.10). 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 to build the top tree—to constrain the ML later-on and to create better predictions. Then, the 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 subsection 6.3.3, 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 6.11). To meet the requirements of some biologists who are used to

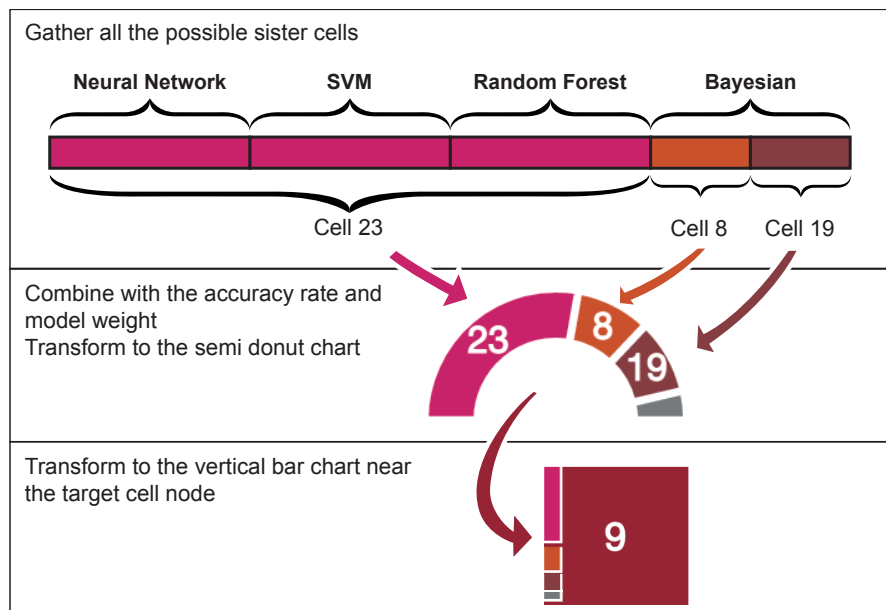


Figure 6.7 – The connection of visualization design in the different information levels. The gray part is the uncertainty where models can make mistakes about the target cell (cell 9 in this example).

doing the lineage assignment top subtree by top subtree, we also enable the experts to double-click a specific top-level subtree to only display the cells within this subtree, without all the other cells in the 3D view and with the selected subtree highlighted in the hierarchical tree (Figure 6.12). 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.

6.4 IMPLEMENTATION DETAILS

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

6.4.1 Datasets

We received the original embryo datasets in SURF format from our biology collaborators, 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

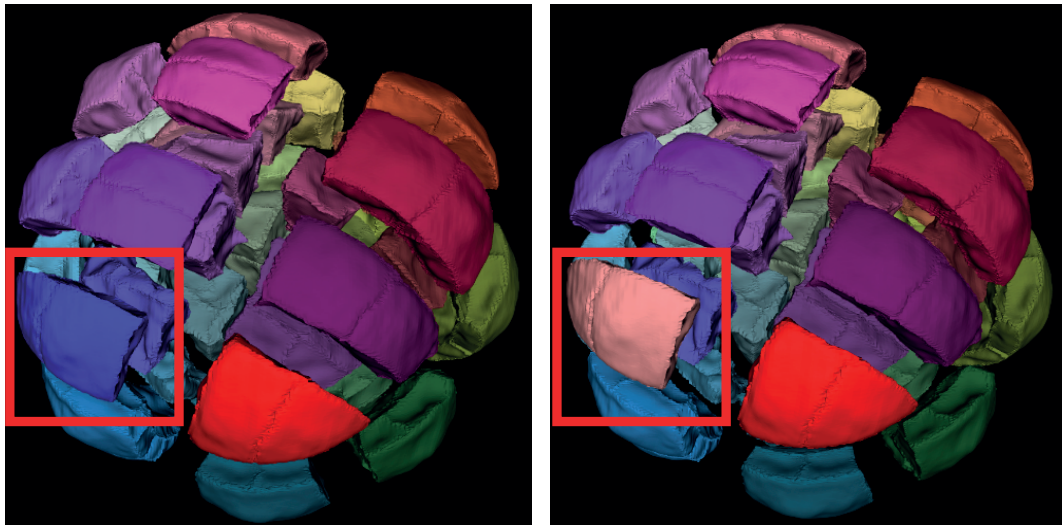


Figure 6.8 – This view shows the similar pairs with the target cell. The left image indicates the original embryo, and the right image demonstrates the color change of the similar cell in the detection mode. In these figures, the target cell is highlighted in red, and the similar cell in pink. In this figure we highlighted the similar cell with a red box.

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. We extract from this file the corresponding lineage hierarchy for the given embryo.

6.4.2 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`. The specific packages we used are `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). The neural network model consists of two hidden layers with *ReLU* activation functions and one final output layer with *softmax*. We also used the library `ml.js` to get the model performances with a k-fold (k=50) cross-validation approach. Also, 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 customi-

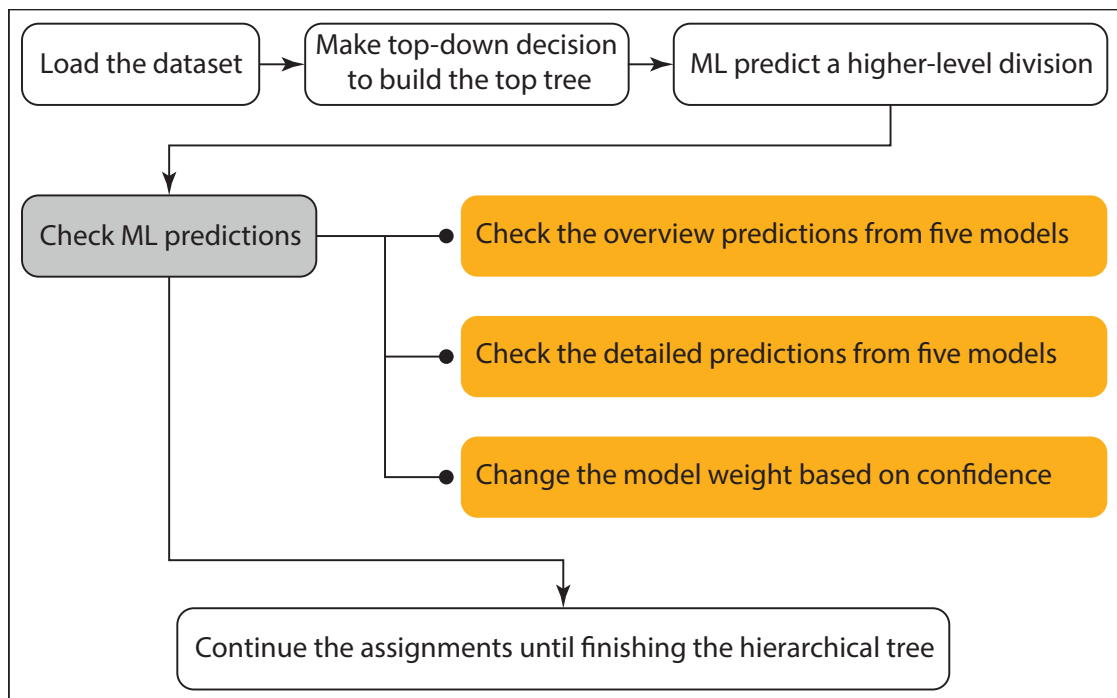


Figure 6.9 – The workflow of using LineageD to do cell lineage. The steps with white backgrounds are based on our previous work [84], and the steps in orange represents the new stages we added in LineageD.

zed by the biologists, yet a re-training with a custom feature selection takes a computation time in the order of hours to days.

6.4.3 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 [21]. For visualizing the embryos in 3D, we used vtk.js (v. 19.0.4) [163].

6.5 CASE STUDY

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. We extended our existing ethics application for

our work on cell lineage (AVIS n° 2021-46) to get permission for this human-subjects evaluation from our institution's (Inria) ethical review board. We also pre-registered the study online (osf.io/2f6uc).

6.5.1 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 [84], 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.

6.5.2 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.

6.5.3 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 fill in. 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 we received their formal approval to participate. 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 in the expert understanding 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, asking them about their experience with the interface. We also asked them to fill out a System Usability Scale (SUS) [80] to assess their experience.

6.5.4 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 in assignment 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 with

similar steps taken using LineageD [84], which means that they did not refer to the detailed predictions of each cell during the assignments and instead made decisions on their own. 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 volunteer 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 the 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 subtrees. They would finish the assignments of all cells in one subtree (a quarter in our example data) and then move on to the next one. Four of them used our subtree view (Figure 6.12) to only focus on a specific quarter, while the other two went through the cells in the tree order but without engaging the subtree view.

Visual Representations. All participants appreciated the visual information design for the hierarchy and machine learning. P1 and P5 noted that they can understand everything in 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 replacing the traditional 2D slice images so that biologists can see the embryo and have a better understanding of it. P4 also suggested to export 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 for 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+ is 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, 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 it makes sense. She felt she did not have to think much but just traversed the proposals from machine learning.

As for the improvement of machine learning itself and how it can be deployed, though P1 and P2 did not refer to the detailed predictions of each cell from machine learning, 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 interacting 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 the 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 to predict future pairs using the updated model. Another wish of P2 and P6 was that the machine learning 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 subtree, instead of only a single level.

Interaction Design. We asked participants about their interaction experience both with the general interface and with the machine learning 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 subtree focusing function. It enabled them to clearly make decisions within the range constrains 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 thought it may be better to provide 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 of working with LineageD+, P4, P5, and P6 thought they would feel very confident because they can easily understand and use the tool to check

very 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 biologists about whether they thought the tool would change the traditional approach they used in the assignment process. All of them believed that LineageD+ can possibly change the strategies they used. P1 and P2 assigned the cell lineage quarter by quarter, and the machine learning predictions built the tree level by level. They were used to the quarter-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) [80]. 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.

6.6 DISCUSSION

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

6.6.1 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 em-

bryo, 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.

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, for instance, treated the ML proposals as if the machine learning was suggesting the potential sister cells to her in real time, and she “explained” to ML models (i. e., to us in the think-aloud protocol) the reasons of why a prediction was reasonable or not. Though she spent more time in traversing the proposals, such a workflow was appealing to her and may be similarly 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 are needed.

6.6.2 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. For example, in our case 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 benefit them in the future. To help experts with such a situation, when designing a system we need to consider recording their behavior and developing 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.

6.6.3 Visualization for 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 one should interact. Our visualizations of the details of the ML model predictions thus serve as a mechanism to support this human-AI teaming, 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.

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. Although multiple ML models can partially make up for this problem, the biologists would feel more confident if they check 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 a 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.

6.7 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. 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 the 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. Other minor usability issues addressed 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. We used the mesh data from a paid tool instead of a more generally-used tool, and it adds difficulty to the generalization.

6.8 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 instead treated it as a tool that helps people make decisions but that no longer is the final authority. Visualization, in this relationship, empowers people to enable them to find a balance between their own experience and ML models’ proposals and thus to engage with the ML as if it was another collaborator. As people have a diverse set 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 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 a supportive partner 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.

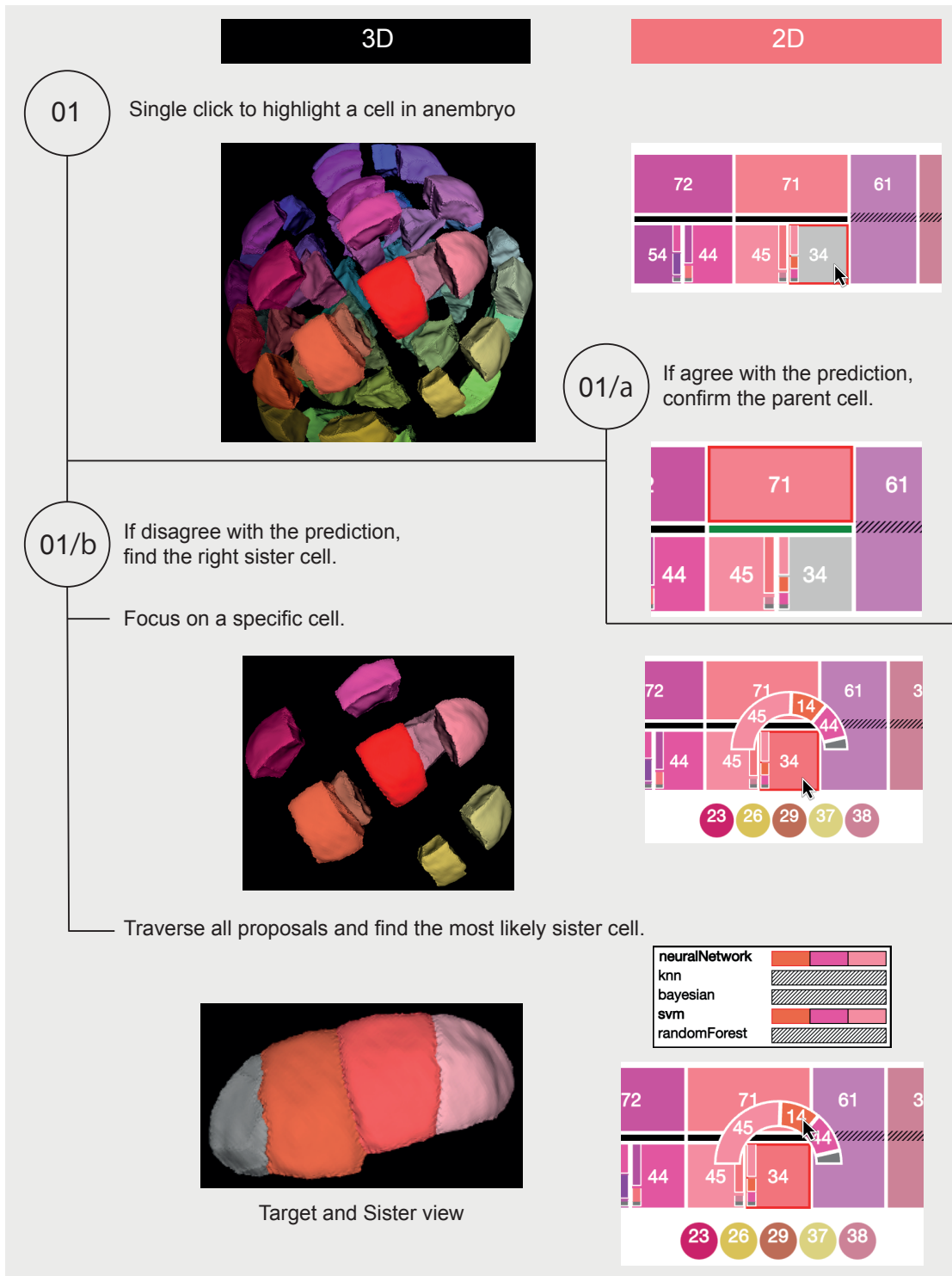


Figure 6.10 – The interaction process of checking the model predictions. Biologists can target one cell from either the 3D view or the 2D hierarchy tree. If they are happy with the prediction, they can confirm the pair, and the bar then turns green as in 01/a. If they disagree with the prediction, they can double click to see the details of prediction and make an informed decision.

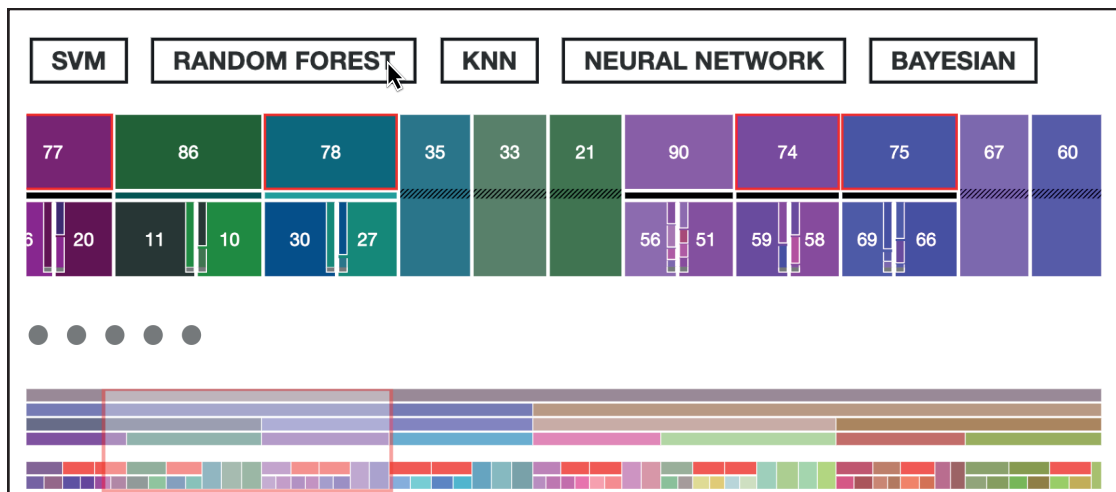


Figure 6.11 – The illustration of the hovering effect. In this figure, users hover the model Random Forest. All the pairs predicted by this model would be highlighted with red strokes in the hierarchical tree. Also, the corresponding cell in the thumbnail would turn red too.

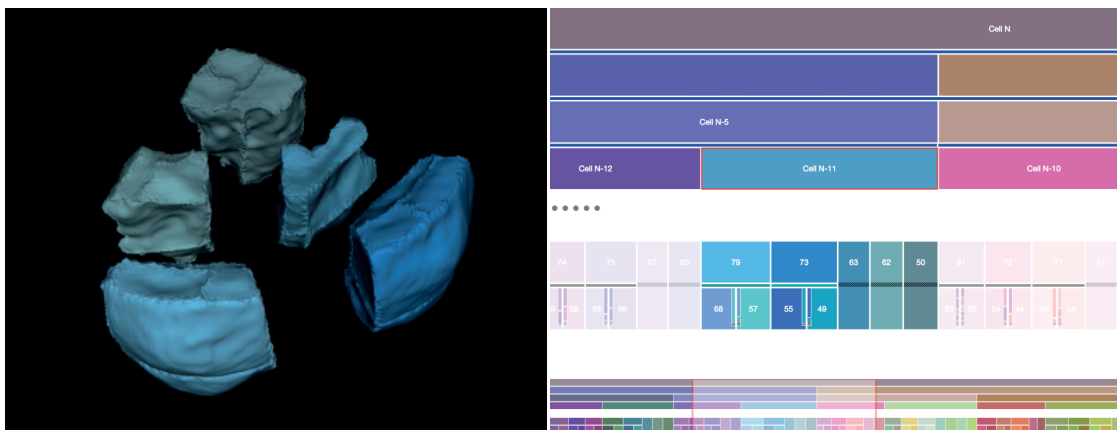


Figure 6.12 – The illustration of users targeting one subtree. Here a user focuses on Cell N-11. The 3D view only presents the cells within this part, and in the hierarchical tree, we make the other cells transparent.

DISCUSSION AND CONCLUSION

7

In this thesis, we mainly contribute by using diverse techniques (interaction, visualization, and machine learning) to solve scientific problems such as plant cell lineage. We conducted a series of design studies as described in the previous chapters (chapitre 4, chapitre 5, and chapitre 6) to explore how to apply them in different steps of solutions. The methodologies and the design process of our systems could potentially be applied to other scientific domains, such as the way to collaborate with domain experts, the way to link the 3D and 2D representations, and the use of Human-AI teaming to deal with the cases where both algorithms automation and design study are needed. We discuss the takeaways from our thesis in this chapter.

7.1 INTERACTING WITH DENSELY-PACKED OBJECTS

As we have described in chapitre 4, plant cells in an embryo are densely packed together without any space between every two adjacent cells. In such a densely packed environment, it is not possible to use traditional interaction techniques such as ray-casting [155] for selection, especially in the case where biologists have no idea which cell to select unless they see it. So the occlusion of inner cells by the outside largely prevents biologists from accessing them. Moreover, biologists need to precisely analyze local neighborhood situations of cells to be able to make correct decisions about their development, and the embryo can be large such as containing more than 200 cells. Hence we need specific interaction techniques to access and observe cells within an embryo.

We investigated three selection approaches in 1D and 3D as stated in chapitre 4 : a technique that lists all the 3D objects in a separate view (1D List Selection), a technique that uses 3D explosion to generate space between the objects in the 3D view (3D Explosion Selection), and a combination of these two techniques. In chapitre 5 and chapitre 6, we also implemented an interaction technique that iteratively peels off the surface layers of the objects (Peeling Selection). All these approaches translate the interaction in a dense environment into one that we can work with : either by showing them in a separate list (List Selection), by spreading the cells out into a more sparse representation (Explosion Selection), or by revealing layers iteratively (Peeling Selection). In addition, another idea is the slice-based approach that was used by our domain collaborators traditionally, which only shows a single 2D cutting plane for the dense 3D environment. It is also a possible solution as we could add a cutting plane for biologists to control and show the corresponding cross-section view of the embryo, but we found it in our pilot studies to not work as an efficient approach. One possible reason can be that it was difficult for biologists to master how to freely control the cutting plane as they are not familiar with the 3D interaction techniques.

We can thus summarize all these approaches into three categories for solving the selection problem with the densely-packed objects : 1. Display the abstract information of datasets in a separate view (List Selection and the traditional slice-based selection); 2. Adjust the datasets in 3D so that the problem is essentially converted to the familiar traditional selection approaches (Explosion Selection); 3. Display only a part of the whole dataset (Peeling Selection). These approaches (List Selection, Explosion Selection, and Peeling selection) may also work for other data such as organs in biology [159] or high-entropy nanoparticles in chemistry [213]. For other datasets or in other application domains, however, they may not work as well. Take the Explosion Selection as an example. To apply this technique, we need to find one/multiple center(s) as geometrically meaningful baseline(s) to explode the object, and it can be difficult for data with other characteristics. For example, the DNA structure is linearly structured, very long, interconnected only along the structure, and then folded or packed in many levels of structure. Though we could use the explosion technique to partially divide two chains, it may not work for checking the detailed components. Another example is machine assemblies, where the parts are not as compact as the cells in our application, and they usually have inherent assembly directions. In such cases, we could face non-meaningful explosions unless we use structured data about the assembly that maintains the meaningful assembly [76, 137]. Another aspect to consider is that selection purposes can be different, and techniques may not apply to all scenarios. The purpose can be selecting a specific data point,

multiple points, or a range of points. The List Selection is especially helpful in traversing the datasets, while it would fail in other cases where people only need to select a rough range of data points that are spatially close.

Besides these techniques, there are other potential solutions for interacting with densely-packed objects, as we can shrink these objects while keeping them in the same positions. Also, we can add a cutting plane with limited rotation angles and only show objects on one side of the cutting plane [66] instead of one simple cross section view. We can support combining two/multiple techniques to interact with the objects. For example, by combining the Explosion Selection and Shrinking Technique in the interaction with plant cells, we could explode the embryo and shrink the surface cells so that biologists may better examine the inner cells.

7.2 DESIGNING FOR LINEAGE

Fundamentally, lineage relates to both structural and temporal information : it shows the structure (e. g., the cell division relationship) and also encodes part of the actual temporal change in its very hierarchy (e. g., the embryo division time stage) [203]. Traditional trees, even rooted, do not necessarily encode temporal data (e. g., a folder structure only encodes a hierarchy). As such the tree representation in our special case is unique because it encodes both aspects. The way we designed to visualize this special cell lineage is to encode the temporal division stage with tree levels as we discussed in chapitre 5.

One of the reasons why we chose to represent the temporal information in the tree itself is that a division generation is a qualitative form of temporal development that the biologists deduct by reasoning backward, and we can assume that there is regular (i. e., taking approximately the same amount of time) and repeated waves of cell division. We can then use the hierarchy levels to represent the division generations. Another reason is that independent of this special case, a plain abstract tree representation can already use a lot of screen space, in particular in situations with the exponential growth of the depicted structure as it is the case for cell lineage. Adding other representations aside would make the interface too crowded and add difficulties for users to connect multiple views. Yet, the tree itself can serve as the basis of additional data representations about the depicted elements. For example, in LineageD (chapitre 5), we used the tree nodes' colors, widths, and heights to encode cells' 3D relative positions, relative volumes, and division stages. In LineageD+ of chapitre 6, we added stacked bar charts to tree nodes to represent the overall model predictions. There are also numerous other examples from literature which use different properties

of trees to encode other datasets. For example, EVEVis [132] encoded time and species in the tree, and Zheng et al. [219] added various diagrams, e. g., sunburst charts and icicle plots, on treemaps. We can see that trees have great potential in combining with other multiple representations.

Moreover, in our case, each level of the tree actually relates to a cell division stage and thus to a 3D representation, i. e., when biologists switch the division level in the tree, the 3D objects need to change accordingly to display the corresponding status. With the added components, the tree can then serve as an interaction medium to control the 3D view when contents in two views are linked. As in our scenario, every cell has a corresponding object in the 3D environment and a matching node in the 2D tree. Due to the temporal character of our specific trees, they show the progress and development of the 3D representation, i. e., of the embryo as well as the progress of the biologists in doing their lineage assignments. We also, in fact, encode the progress in LineageD+ with additional elements overlaid to the tree representation by using horizontal bars between parent cells and children cells to indicate whether biologists have checked this pair or not. Thus, by referring to the bars, biologists can see how many remaining pairs to check.

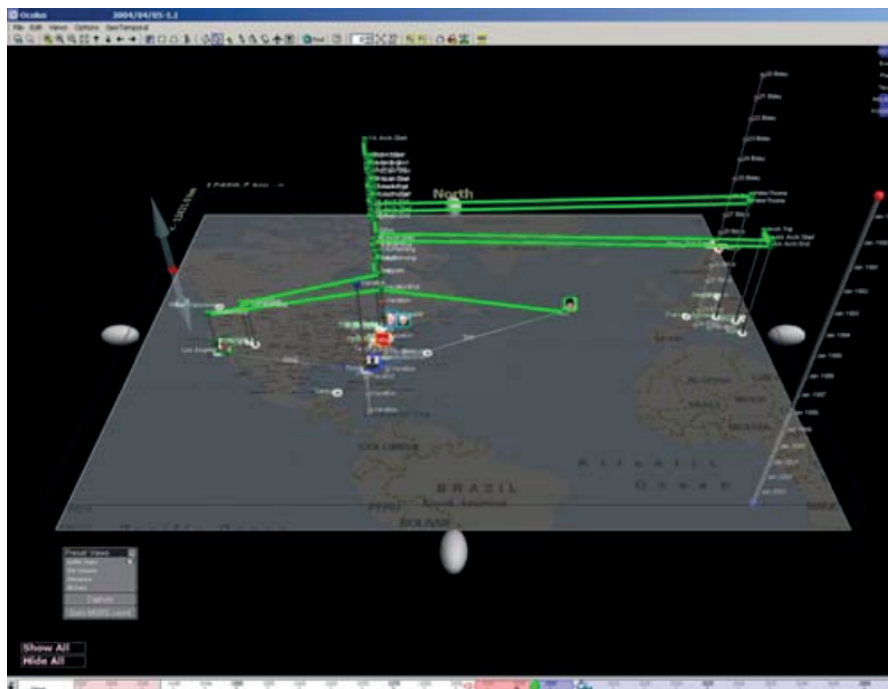


Figure 7.1 – The screenshot of GeoTime [90] to combine both temporal and geospatial visualizations. Image © 2005 SAGE Publications, used with permission.

Not only for establishing a binary hierarchical tree for plant cell lineage, our

tree design can also be extended to other tree problems because (1) Our tree contains many components which can be used to encode diverse data (we could, for instance, also replace stacked bar charts with other types of visual representations); (2) Our tree can be established using two approaches : top-down and bottom-up approaches; (3) Our tree is a control panel for the associated 3D view. For example, we could use the system to represent and study assembly instructions where a large number of parts come from different places. The tree could then represent the assembly order with approximate delivery time periods. By using the tree, users can control the 3D view to show the corresponding assemblies on the map in a specific time period, similar to GeoTime [90] which visualizes locations and events on a map (see Figure 7.1).

7.3 DESIGN FOR DOMAIN EXPERTS

From our working experience with biologists, we found that the essential part of enhancing the collaboration between visualization people and domain experts is to try to understand each other. People from diverse backgrounds would have different ways of thinking. That requires us to have regular meetings and keep track of all the discussions we had so that we could refer back to them at any time. Also, despite the physical constraints, it would be better to exchange ideas in person instead of having meetings remotely, as in-person communication could be smoother and more effective.

Other than the collaboration approach, we also found that experts' traditional working process often needs but may not (yet) have access to advanced interactive visualization techniques. For example, in our case, biologists need visualization to display the 3D embryos and interaction techniques to explore the embryo and to build and view the hierarchy.

Such a lack of advanced interactive visualization techniques may cause problems. For example, in our case, the biologists were not familiar with such techniques, and we found that some thus expected that current technology somehow magically solves their problem. For instance, they may expect that machine learning can predict the whole lineage hierarchy for an embryo at the push of a button. When they then face reality (i.e., that ML cannot predict lineage completely precisely, such as due to a lack of enough training data), they may lose confidence in certain techniques, such as ML, once they find out that the results differ from their expectations. As described in chapitre 5, the biologists told us that they felt that ML did not perform well, and they would then assign cells by themselves.

Because experts usually have limited knowledge about advanced visualiza-

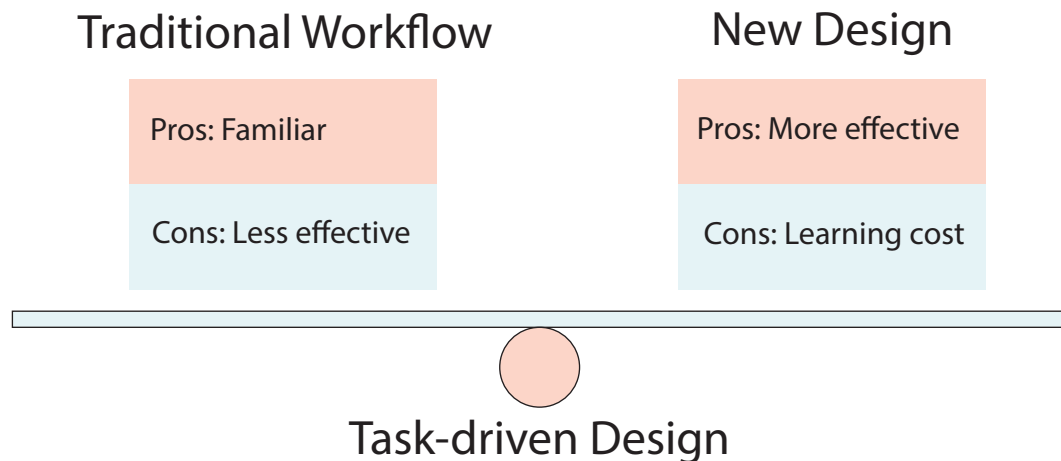


Figure 7.2 – The illustration of balancing traditional workflow and new design with advanced technologies to solve a scientific problem. The new design usually involves balancing different techniques inside, e.g., visualization, interaction, and machine learning.

tion and interaction techniques, they may also want to stick to their traditional tools. For instance, in our case, they expected to stick to their slice approach and then wanted the ML to predict the sisters based on the slices. This sticking to traditional ways of problem solving brings designers a problem as illustrated in Figure 7.2. On the one hand, their traditional workflow usually keeps drawbacks such as being tedious and less efficient, and we can address this with better visualization and interaction design. On the other hand, while an improved workflow may be intuitive, experts can take time to get familiar with it. To overcome this issue, we recommend implementing their needed technology (i. e., machine learning, even if it is not perfect) and including other techniques (e. g., visualization and interaction) that can improve the problem-solving process. During the design, we need to balance the experts' familiar workflow and the new one. To do that, we could design iteratively with domain experts, or we can design one or several proper workflows in the interface. As in LineageD, we support either establishing the hierarchical tree manually or checking and updating it based on ML predictions in 3D or 2D views.

7.4 COMBINATION DESIGN FOR 3D AND 2D REPRESENTATIONS

We saw that there are application cases where we need to have both 3D and 2D representations when we want to refer to spatial datasets and abstract information. When two views are inherently linked, such as in our case, where the cells in all views are identical or related, connecting these views is necessary because viewers can gain a thorough overview of information at a specific stage. This is inspired by traditional linked views [185], but we apply this concept to 3D-2D hybrid systems.

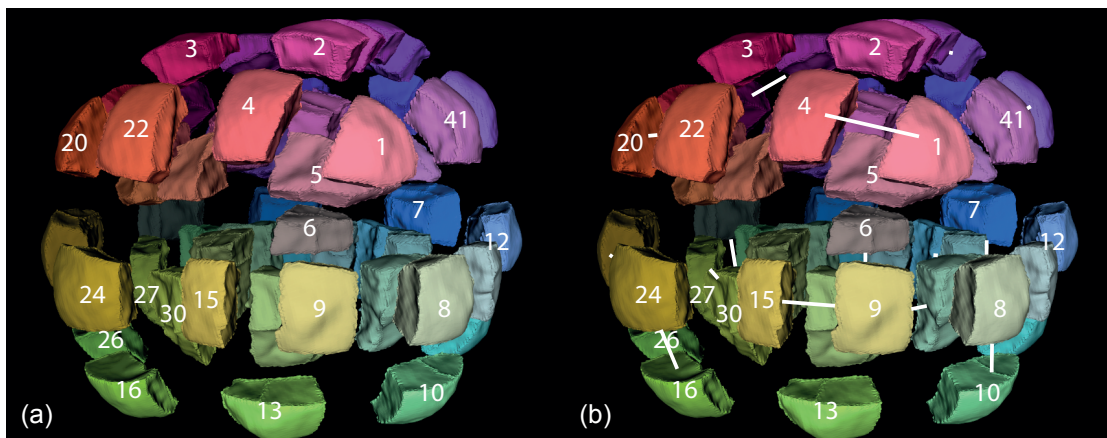


Figure 7.3 – The illustrations of potential design with (a) cell names on 3D and (b) both the names and lines representing the relationship of two cells being sisters.

Constrained by the complicated context and difficult orientation interaction (e. g., rotation, zooming, and panning), the 3D environment usually contains only 3D information or includes simple 2D abstracted data on that. For instance, in our plant embryo visualization, if we attach a cell name to each cell, as shown in Figure 7.3(a), there is a severe overlapping problem which makes the interface messy. If we add node-linked lines to connect every two potential pairs within this time stage (see Figure 7.3(b)), the overlapping problem becomes more severe. We could imagine that when users rotate it, the situation could be worse. We usually have to visualize lots of information (e. g., the hierarchy data for all the time stages) in addition to the 3D spatial data to solve a complicated scientific problem because all these data are important for decision-making. In these cases, we then need to encode other additional/abstract data within 2D representations. Linking the 3D view with 2D representations is necessary and important because with linking, the information related to 3D objects can be better

received and understood through the 2D view.

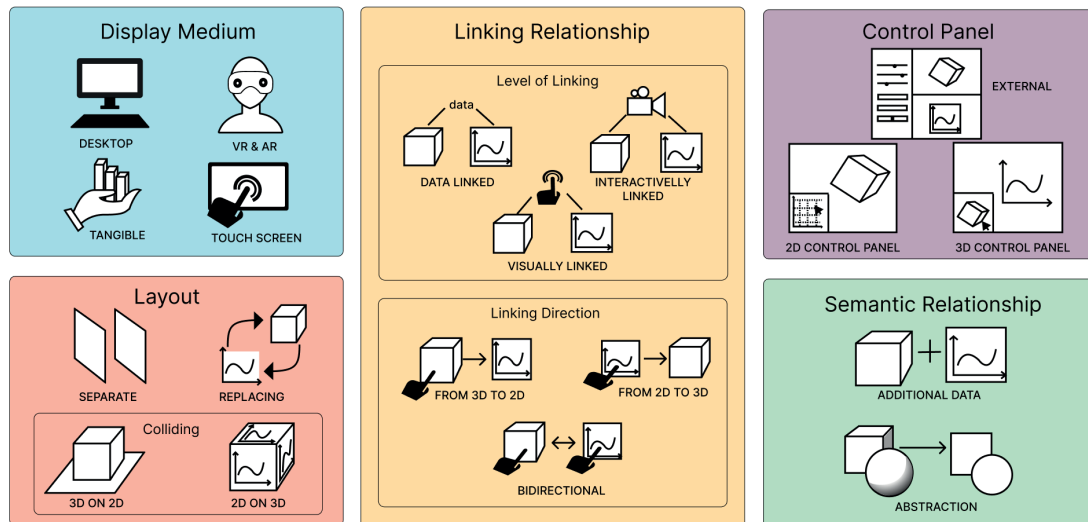


Figure 7.4 – We duplicated this figure from Figure 3.7. In this poster [160], we made five categories to group literature which contains 3D and 2D representations.

In face, we explored the different ways of linking 2D with 3D views in a survey of past visualization literature [160]. Here we found that there are three linking relationships between two views (see Figure 7.4). One is data linking, where views share the same data, but users cannot directly manipulate the views. In these cases, the 2D representations are usually the abstraction of 3D representations, e. g., Eulzer et al. [64] used a flattened 2D diagram to represent essential parameters of 3D mitral valve (see Figure 7.5). Another relationship is the visual linking where users can rotate, zoom in, or zoom out one view, and the other view will be updated accordingly, e. g., in the work of Eulzer et al. [63] as shown in Figure 7.6, the system used 2D diagrams to show the detailed parameters of spinal anatomy, and rotating the 3D spinal discs would rotate 2D diagrams as well. The last one is interactive linking. Interactively linking two views has two aspects. One is that operating with a view can result in changes in the other view. For example, in previous work, users can select, filter, and brush across views via interaction. In LineageD, we also supported highlighting one cell in all views by selecting that cell in any view. We go beyond this highlighting linking in a way that 2D representations can, in fact, control the status of 3D models. That is to say, by picking a specific level in the hierarchical tree, the 3D cells will be combined/separated to match the corresponding level. The other aspect of interactively linking is that the operations in either view are the same or equivalent. For instance, a lot of related work links two views with the first approach,

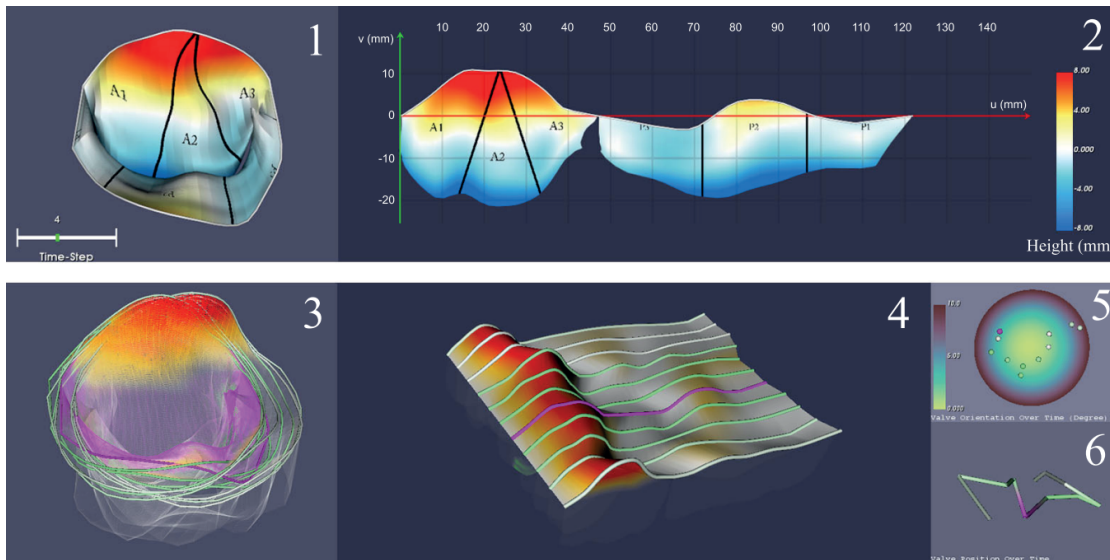


Figure 7.5 – In the screenshot of the interface developed by Eulzer et al.[64], we can see that the 2D diagram (part 2) is the flattened representation of 3D mitral valve (part 1). Image © 2020 IEEE.

e. g., Byška et al. used a 2D collar representation to display the protein tunnel bottleneck contour and control the 3D view [29]. We still believe that the second approach, i. e., using the same or equivalent operation in both views, also eases people’s learning curve. For example, we used the same interaction to correct assignments in both 3D view and the hierarchical tree for targeting cells and correcting predicted pairs. Once biologists noticed this, they were able to interact with both views easily. Though designing the same interaction for all views enhances people’s awareness of that the views are linked, we have to admit that it is an added value and designers do not have to pursue it since it can be difficult to realize.

To design a system with both 2D and 3D representations, we need to first identify the inner relationship between different types of representations. One relationship is that one view adds information to the other. In this case, the linking should clearly state what this information is related to. For example, we encoded the ML prediction results in the tree, and these predictions are not visible in the 3D environment. We emphasize for which cell the predictions are made by using the same colors of cells in both the 3D environment and the hierarchical tree. Also, when biologists hover over the proposals, the corresponding cell would be highlighted in the 3D view. Another potential relationship is that the 2D view visualizes the abstracted information for the 3D view. The specific design of this visualization of abstract data about 3D objects should provide an over-

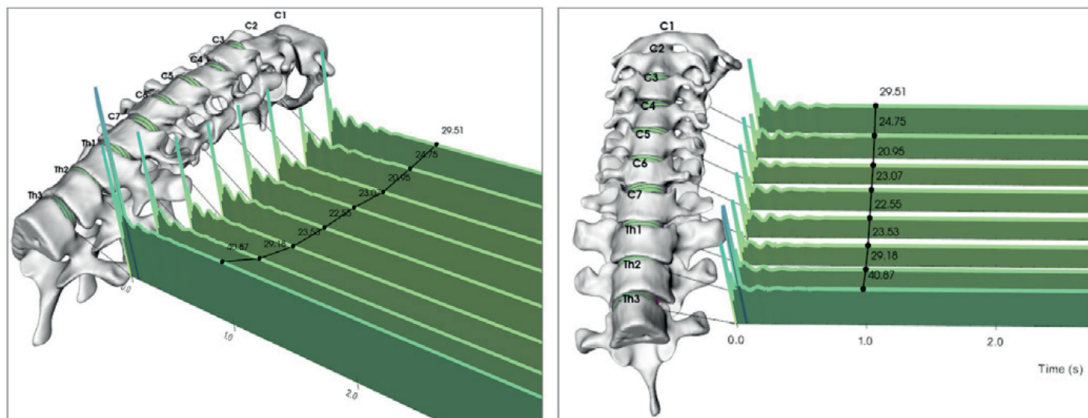


Figure 7.6 – In this example from the work by Eulzer et al. [63], a pack of 2D diagrams are attached to eight spinal discs, and when users rotate the 3D objects, the 2D representations would rotate accordingly. Image © 2020 IEEE.

view of the abstraction and the related 3D model to highlight the abstraction relationship. For example, in LineageD, the level in the lineage tree represents the cell division stage, and each node represents one cell in a specific division stage. By changing the level, cells in the 3D would merge/separate to form the embryo shape in that time period. With this interaction, biologists can clearly understand what the level and each node mean for the embryo.

Such combination design for linking 2D and 3D visualization could potentially be applied to various fields which have spatial data or multi-variable data. We could use the 3D dimension to encode the additional variable. For example, based on a 2D plane, we could encode the temporal trajectory information on top as Tominski [195] designed to visualize individual and cross sets of trajectories. The idea can also be used in genealogies. Previous work such as GeneaQuilts [20] introduced a diagonally-filled matrix to represent genealogies, but it missed the geometry information which could be valuable for historians to analyze. Adding the individual moving path on top of it could potentially bring new insights.

7.5 VISUALIZATION FOR MACHINE LEARNING

Some tasks in the scientific domain can be improved with the help of ML, especially for those complicated problems which traditionally require tedious and repetitive manual work like cell lineage. Traditionally, these tools with ML support make the ML control the tasks, and either the ML is merely a tool (that works at the push of a button) to complete the task or humans interact with the

system to improve the performance of the ML models. These cases where ML is dominant requires ML models to be as precise as possible. However, sometimes it is impractical or impossible because it would require too much work or a lot more training data than is available to generate such ML models. For example, in the cell lineage problem, humans used to manually do cell lineage for numerous embryos, and thus, the training datasets are limited. In these situations, there are inherent uncertainties in ML predictions, and humans need to be aware of them in the decision-making process.

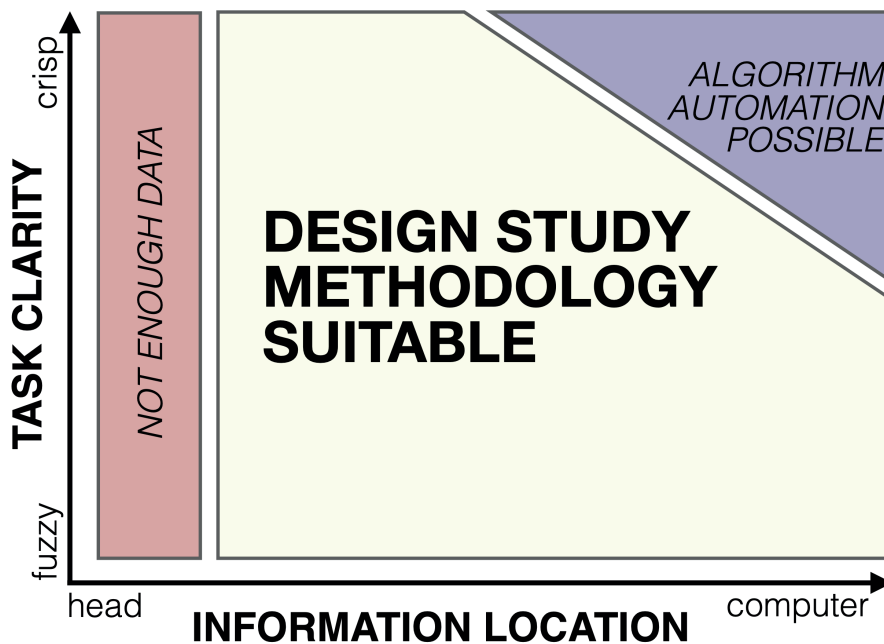


Figure 7.7 – Sedlmair et al. [167] proposed two axes (task clarity and information location) for identifying the usage scope of design study methodology. © 2012 IEEE.

This situation reminds us of the work of Sedlmair et al. [167], who used two axes (task clarity and information location) to identify the design study suitability as shown in Figure 7.7. On the top-right area where the computer has enough data for a relatively crisp task, algorithms can automatically produce solutions, while under the borderline, the authors recommended using their design study methodology. In our case, we have precisely a situation on the boundary between the two regions because (1) we do not have enough data for a complete prediction of the lineage; (2) the task is not so crisp because making the decision based on experts' observation and relied on their experience, and the decision-making is not clear and easy; and (3) ML is possible and crucial to ease experts' workload, yet the experts disagree occasionally, and the training data is limited. This situation actually falls in the boundary between the right-top corner and

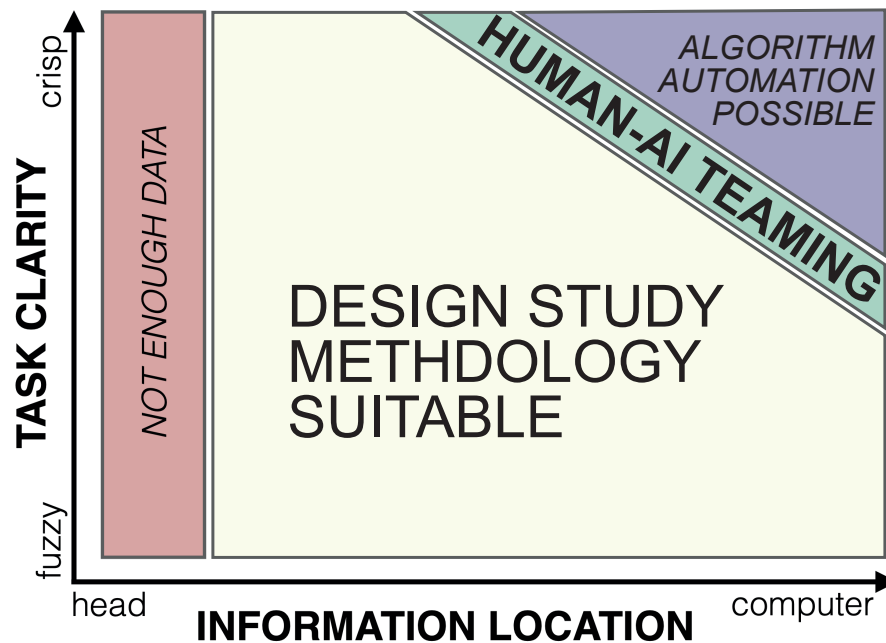


Figure 7.8 – Based on the Figure 7.7, we added an area between Algorithm Automation and Design Study Methodology. In that area, we recommend using human-AI teaming to combine the advantages of human thinking and AI automation.

the white part in Sedlmair et al.'s model, and we would argue that there is not a clear cut-off as depicted in their figure (Figure 7.7) but more of a fuzzy transition zone. Our solution is to use ML paired with additional visualization and interaction for Human-AI Teaming. In a way our approach addresses the problem which was not posed by Sedlmair et al., i.e., for a task with limited computer data and undecided task clarity, whether design study or ML alone would be the best solution. We thus propose a new area to replace the fuzzy boundary between the white part and the top-right area in the suitability diagram proposed by Sedlmair et al. as we show in Figure 7.8. In this new area (which in practice also has a fuzzy boundary), even if complete automation is not possible due to either the limited computer data or non-crisp tasks, visualization can present the imperfect predictions from ML and allow people to understand different possibilities and potentially correct the ML's results. In addition, for those tasks where ML is possible according to Sedlmair et al., but people are unlikely to always be satisfied with the results, visualization and interaction then enable people to similarly freely make final decisions. Thus, in our newly proposed area, humans and AI are collaborating with each other, and visualization provides humans with the ability to control the ML models and the task completion process.

This is where visualization can come in as a tool to improve the work. Visuali-

zation can visualize the results of ML predictions from different models and also visually highlight the potential errors that ML can possibly make. For example, in our evaluation study with LineageD+ in chapitre 6, visualization display the overview and detailed predictions from ML as well as the model confidence of each predicted pair, and biologists could make decisions based on this information. However, from our study with LineageD, we found out that people's perceived accuracy rate can largely affect their confidence in models. If people keep receiving wrongly predicted pairs, they may lose their confidence in ML models. Yet, in our case, biologists want to check the least confident predictions from ML. Thus, we need to trade-off between presenting ML predictions with less confidence and more confidence. The approach we adopted in our system is to enable biologists to build the hierarchical tree from a top-down way, and then within each sub-tree, we ordered the predicted pairs from left to the right according to their model confidence.

Thus, within this relationship, visualization can be a way to allow humans to remain in control, not by showing the ML's reasoning (explainable AI), but by showing the results of one/multiple ML model(s) and their certainty, and to allow the human to make the final decision. We believe that, for complicated scientific problems such as ours, humans should be in control of the whole task, including the usage of ML, visualization, and interaction. Humans need to combine all these techniques to make better decisions.

7.6 FUTURE WORK

Our work certainly needs to be continued in the future, and there are plenty of avenues of future work. The first direction is about our tool. We want to explore how our tool can be used for other datasets (such as animal embryo datasets and organ datasets). However, we talked with animal biologists about the potential collaboration and found out that, because the division of animal embryos is highly reproducible [75], they have reasonably good techniques to automatically reconstruct the history. Except for the lineage problem, based on biologists, they have challenges in building the history of mutant embryos of plants and animals. We are interested in how we could use our tool to deal with these embryos for whom the division may not follow typical patterns, and thus the tree may deviate quite a bit from the typical binary tree. Moreover, we currently use traditional devices (computer, mouse, and keyboard) for interaction. We would like to investigate the interaction methods with advanced techniques, e. g., VR and AR, to see how it works for solving complicated scientific problems. Also, we are interested to know whether our design can change biologists' workflow in

doing cell lineage in the long run. Thus, we plan to conduct a long-term study to see whether a well-designed system can change experts' traditional workflow and whether their feedback would change or not. Also, we are curious about how our tool and approach could be adopted in the education field. Unlike experts, students are at the stage of learning the rules, such as cell division and cell assignments in our case. From the tools, they can visually learn how cells divide and check whether their newly assigned cells make sense or not. Having a study to compare students learning new concepts in a traditional way with a newly-design approach would be interesting.

Then we want to systematically investigate how researchers usually link 2D and 3D representations and why they choose that specific approach. For each approach, we want to investigate the advantages, disadvantages, and applicable scenarios. Also, we intend to explore whether the number of these views will influence the linking interaction design. Based on these, we plan to summarize the design guidelines for future combination system design of 3D and 2D representations.

The third direction is about how we can better make use of ML in solving scientific problems. For example, we may need multiple ML models for diverse steps, e. g., models for cell assignments and models for feature extraction in one system. Then there could be multiple groups of ML models where each group aims to solve one step of the task. For such a system, the question of how we can organize, correlate, visualize, and interact with them remains unclear. One idea for solving this problem is going through the literature and investigating how models responsible for diverse steps in the workflow are connected together.

The final direction we are interested in is the collaboration design for Humans and ML. As we discussed in the paper LineageD+ in chapitre 6, ideally, ML can serve as a collaborator to help humans to make decisions better. However, we have no idea how to use visualization and interaction to enhance such a collaboration relationship. For example, in terms of different scenarios, to what extent should people be able to control ML and the solution process? How could we support people to control the involvement of ML with the help of visualization?

7.7 SUMMARY

In this thesis, we worked on using visualization and interaction to solve scientific problems using the example of the cell lineage problem. We mainly investigated this question in four parts. First, for such complicated problems, different scientific datasets sometimes have unique properties that demand spe-

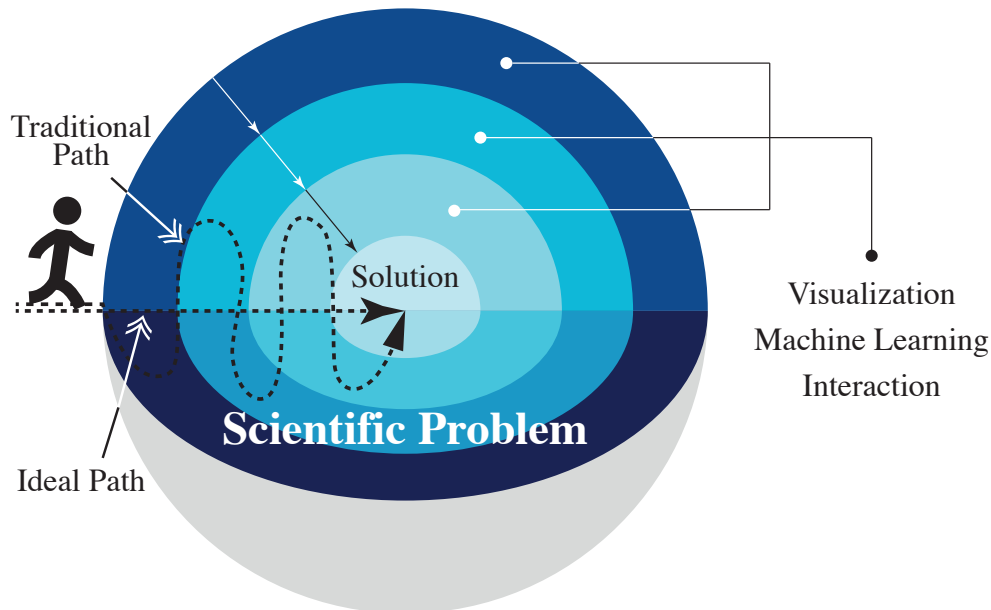


Figure 7.9 – The illustration of the peeling process to solve a scientific problem. The layers refer to diverse techniques (visualization, machine learning, and interaction) or combinations of these techniques.

cific interaction/visualization techniques. Second, we need to understand biologists' actual needs and balance their technique-driven requirements with the new tool, which has improved but contains unfamiliar interaction and workflow. Third, since scientific data usually has spatial information, we need to visualize scientific datasets both in 3D and 2D representations. Connecting these two views visually and interactively enables people to have a better overview of all the information and thus make better decisions. Lastly, we often need to include ML support in the system to help experts with problem-solving. Experts are skilled and knowledgeable, so humans should be in control of the decision-making process. They should be able to control how to use ML models, what visualization they want to refer to, and what workflow they want to pursue during the process. To solve complex scientific problems, we, in the end, would like to propose the following peeling metaphor to indicate how people could get access to the solution with different techniques (ML, VIS, and interaction) as shown in Figure 7.9. Experts who are working on a scientific problem usually have an idea about the solution, i. e., the core of the image, since they are experienced and skilled. Yet, they still need enough useful information to make satisfying decisions while all the information is intertwined together like a spherical hairball. The process is like they are walking through the sphere to get to

the core. In their traditional ways, they used to have unorganized information, and they needed to handle it by themselves. For example, during the plant cell lineage establishment, though only adjacent cells can be sister cells, when assigning a target cell, biologists used to see all the cells because 2D slices from the microscope cannot filter the adjacent cells for them. In this case, the path can be complicated, as shown in the image, which means experts spent more time and energy solving the problem. Thus, we need to provide biologists with different techniques to filter useful information so that they can follow an ideal path. Visualization can provide detailed spatial data, abstract information, and can make patterns apparent. Interaction can correlate different views and help users to interactively extract the interesting information. ML can do some work for experts so that users have less work to deal with. These three techniques can largely help experts to effectively get the necessary information and then solve the tasks. In this illustration, the path indicates the solving process, and layers represent the techniques experts used in this process. That is, layers do not have to be three as in Figure 7.9. For example, in a case where an expert used ML first, then visualization, ML, and interaction, there should be four layers. Also, when one step contains multiple techniques, the layer could represent multiple techniques as well. Moreover, the order of experts adopting these three techniques in getting access to the solution is not strictly limited, so the order of layers are not constrained as well. Experts could freely operate the system to get the goal.

BIBLIOGRAPHIE

- [1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. *ACM Trans Graph*, 22(3):828–837, July 2003. doi: 10.1145/882262.882352
- [2] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization. In C. D. Hansen and C. R. Johnson, eds., *The Visualization Handbook*, chap. 36, pp. 717–732. Elsevier, Oxford, UK, 2005. doi: 10.1016/B978-012387582-2/50038-1
- [3] N. Ahuja. Efficient planar embedding of trees for VLSI layouts. *Comput Vision Graph Image Process*, 34(2):189–203, May 1986. doi: 10.1016/S0734-189X(86)80058-5
- [4] S. Amershi, D. Weld, M. Vorvoreanu, A. Fournery, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-AI interaction. In *Proc. CHI*, pp. 3:1–3:13. ACM, New York, 2019. doi: 10.1145/3290605.3300233
- [5] F. Argelaguet and C. Andujar. Efficient 3D pointing selection in cluttered virtual environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, Nov./Dec. 2009. doi: 10.1109/MCG.2009.117
- [6] F. Argelaguet and C. Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, May 2013. doi: 10.1016/j.cag.2012.12.003
- [7] T. Asahi, D. Turo, and B. Shneiderman. Visual decision-making: Using tree-maps for the analytic hierarchy process. In *Proc. CHI*, pp. 405–406. ACM, New York, 1995. doi: 10.1145/223355.223747

- [8] D. Bajusz, A. Rácz, and K. Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):1–13, 2015. doi: 10.1186/s13321-015-0069-3
- [9] M. Baker. Statisticians issue warning over misuse of P values. *Nature*, 531(7593):151, Mar. 2016. doi: 10.1038/nature.2016.19503
- [10] M. Baloup, T. Pietrzak, and G. Casiez. RayCursor: A 3D pointing facilitation technique based on raycasting. In *Proc. CHI*, pp. 101:1–101:12. ACM, New York, 2019. doi: 10.1145/3290605.3300331
- [11] F. Beck, J. Melcher, and D. Weiskopf. Identifying modularization patterns by visual comparison of multiple hierarchies. In *Proc. ICPC*, pp. 9:1–9:10. IEEE CS, Los Alamitos, 2016. doi: 10.1109/ICPC.2016.7503712
- [12] M. Beham, W. Herzner, M. E. Gröller, and J. Kehler. Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE Trans Vis Comput Graph*, 20(12):1693–1702, Dec. 2014. doi: 10.1109/TVCG.2014.2346626
- [13] H. Benko and S. Feiner. Balloon Selection: A multi-finger technique for accurate low-fatigue 3D selection. In *Proc. 3DUI*, pp. 79–86. IEEE CS, Los Alamitos, 2007. doi: 10.1109/3DUI.2007.340778
- [14] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. R. Blum, and J. R. Cooperstock. Did “Minority Report” get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In *Proc. INTERACT*, pp. 400–414. Springer, Berlin, Heidelberg, Germany, 2009. doi: 10.1007/978-3-642-03658-3_45
- [15] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, et al. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods*, 16(12):1226–1232, 12 2019. doi: 10.1038/s41592-019-0582-9
- [16] L. Besançon and P. Dragicevic. The continued prevalence of dichotomous inferences at CHI. In *CHI Extended Abstracts*, pp. alt14:1–alt14:11. ACM, New York, 2019. doi: 10.1145/3290607.3310432
- [17] L. Besançon, M. Sereno, L. Yu, M. Ammi, and T. Isenberg. Hybrid touch/tangible spatial 3D data selection. *Comput Graph Forum*, 38(3):553–567, June 2019. doi: 10.1111/cgf.13710
- [18] S. Besson and J. Dumais. Universal rule for the symmetric division of plant cells. *Proceedings of the National Academy of Sciences*, 108(15):6294–6299, 2011. doi: 10.1073/pnas.1011866108
- [19] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE Trans Vis Comput Graph*, 19(12):2868–2877, Dec. 2013. doi: 10.1109/TVCG.2013.142

- [20] A. Bezerianos, P. Dragicevic, J.-D. Fekete, J. Bae, and B. Watson. Genea-Quilts: A system for exploring large genealogies. *IEEE Trans Vis Comput Graph*, 16(6):1073–1081, Oct. 2010. doi: 10.1109/TVCG.2010.159
- [21] M. Bostock, V. Ogievetsky, and J. Heer. D³: Data-driven documents. *IEEE Trans Vis Comput Graph*, 17(12):2301–2309, Dec. 2011. doi: 10.1109/TVCG.2011.185
- [22] L. Breiman. Random forests. *Mach Learn*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324
- [23] S. Bremm, T. von Landesberger, M. Heß, T. Schreck, P. Weil, and K. Hamacher. Interactive visual comparison of multiple trees. In *Proc. VAST*, pp. 31–40. IEEE CS, Los Alamitos, 2011. doi: 10.1109/VAST.2011.6102439
- [24] B. R. Briggs, J. W. Pohlman, M. Torres, M. Riedel, E. L. Brodie, and F. S. Colwell. Macroscopic biofilms in fracture-dominated sediment that anaerobically oxidize methane. *Applied and Environmental Microbiology*, 77(19):6780–6787, Oct. 2011. doi: 10.1128/AEM.00288-11
- [25] C. Brinkrolf, S. J. Janowski, B. Kormeier, M. Lewinski, K. Hippe, D. Borck, and R. Hofestädt. VANESA—a software application for the visualization and analysis of networks in systems biology applications. *Journal of integrative bioinformatics*, 11(2):43–57, 2014. doi: 10.1515/jib-2014-239
- [26] D. Brodbeck, R. Mazza, and D. Lalanne. *Interactive Visualization - A Survey*, pp. 27–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-00437-7_2
- [27] M. Burch, M. Raschke, A. Zeyfang, and D. Weiskopf. A scalable visualization for dynamic data in software system hierarchies. In *Proc. VISSOFT*, pp. 85–93. IEEE CS, Los Alamitos, 2017. doi: 10.1109/VISSOFT.2017.16
- [28] N. Burkart and M. F. Huber. A survey on the explainability of supervised machine learning. *J Artif Intell Res*, 70:245–317, 2021. doi: 10.1613/jair.1.12228
- [29] J. Byška, A. Jurčík, M. E. Gröller, I. Viola, and B. Kozlikova. MoleCollar and tunnel heat map visualizations for conveying spatio-temporo-chemical properties across and along protein voids. In *Comput Graph Forum*, vol. 34, pp. 1–10. Wiley Online Library, 2015. doi: 10.1111/cgf.12612
- [30] J. Cashion, C. Wingrave, and J. J. LaViola, Jr. Dense and dynamic 3D selection for game-based virtual environments. *IEEE Trans Vis Comput Graph*, 18(4):634–642, Apr. 2012. doi: 10.1109/TVCG.2012.40
- [31] 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*, 38(3):185–199, 2019. doi: 10.1111/cgf.13681

- [32] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Trans Vis Comput Graph*, 23(1):111–120, 2017. doi: 10.1109/TVCG.2016.2598468
- [33] S. Charbonnier, O. Gallego, and A.-C. Gavin. The social network of a cell: recent advances in interactome mapping. *Biotechnology annual review*, 14:1–28, 2008. doi: 10.1016/S1387-2656(08)00001-X
- [34] F. d. Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. L. Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin. Icy: An open bioimage informatics platform for extended reproducible research. *Nature Methods*, 9(7):690–696, July 2012. doi: 10.1038/nmeth.2075
- [35] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proc. KDD*, pp. 785–794. ACM, New York, 2016. doi: 10.1145/2939672.2939785
- [36] M. Choi, D. Sakamoto, and T. Ono. Bubble Gaze Cursor + Bubble Gaze Lens: Applying area cursor technique to eye-gaze interface. In *Proc. ETRA*. ACM, New York, 2020. doi: 10.1145/3379155.3391322
- [37] 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*, pp. 27–34. IEEE CS, Los Alamitos, 2010. doi: 10.1109/VAST.2010.5652443
- [38] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31, 1 2009. doi: 10.1145/1456650.1456652
- [39] K. Danyluk, B. Ens, B. Jenny, and W. Willett. A design space exploration of worlds in miniature. In *Proc. CHI*, pp. 122:1–122:15. ACM, New York, 2021. doi: 10.1145/3411764.3445098
- [40] J. Dardaillon, D. Dauga, P. Simion, E. Faure, T. A. Onuma, M. B. DeBiasse, A. Louis, K. R. Nitta, M. Naville, L. Besnardeau, et al. ANISEED 2019: 4D exploration of genetic data for an extended range of tunicates. *Nucleic acids research*, 48(D1):D668–D675, 2020. doi: 10.1093/nar/gkz955
- [41] M. B. de Carvalho, B. S. Meiguins, and J. M. de Moraes. Temporal data visualization technique based on treemap. In *Proc. IV*, pp. 399–403. IEEE CS, Los Alamitos, 2016. doi: 10.1109/IV.2016.65
- [42] G. De Haan, M. Koutek, and F. H. Post. IntenSelect: Using dynamic object rating for assisting 3D object selection. In *Proc. EGVE*, pp. 201–209. Eurographics Assoc., Goslar, Germany, 2005. doi: 10.2312/EGVE/IPT_EGVE2005/201-209
- [43] W. C. De Leeuw, R. Van Liere, P. J. Verschure, A. E. Visser, E. M. M. Manders, and R. Van Drielf. Visualization of time dependent confocal microscopy

- data. In *Proc. VIS*, pp. 473–476. IEEE CS, Los Alamitos, 2000. doi: 10.1109/VISUAL.2000.885735
- [44] P. B. de Reuille, A.-L. Routier-Kierzkowska, D. Kierzkowski, G. W. Bassel, T. Schüpbach, G. Tauriello, N. Bajpai, S. Strauss, A. Weber, A. Kiss, et al. MorphoGraphX: A platform for quantifying morphogenesis in 4D. *elife*, 4:e05864, 2015. doi: 10.7554/eLife.05864
- [45] H. Dehmeshki and W. Stuerzlinger. Intelligent mouse-based object group selection. In *Proc. Smart Graphics*, pp. 33–44. Springer, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-85412-8_4
- [46] H. Dehmeshki and W. Stuerzlinger. GPSel: A gestural perceptual-based path selection technique. In *Proc. Smart Graphics*, pp. 243–252. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-02115-2_21
- [47] J. Diepstraten, D. Weiskopf, and T. Ertl. Transparency in interactive technical illustrations. *Comput Graph Forum*, 21(3):317–325, Sept. 2002. doi: 10.1111/1467-8659.t01-1-00591
- [48] 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*, 64(3):1155–1170, 2018. doi: 10.1287/mnsc.2016.2643
- [49] A. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-computer interaction*. Pearson Education, Feb. 2005.
- [50] P. Dragicevic. HCI statistics without p-values. Research Report RR-8738, Inria, France, June 2015.
- [51] P. Dragicevic. Fair statistical communication in HCI. In J. Robertson and M. Kaptein, eds., *Modern Statistical Methods for HCI*, chap. 13, pp. 291–330. Springer International Publishing, Cham, Switzerland, 2016. doi: 10.1007/978-3-319-26633-6_13
- [52] J. J. Dudley and P. O. Kristensson. A review of user interface design for interactive machine learning. *ACM Trans Interact Intell Syst*, 8(2):8:1–8:37, June 2018. doi: 10.1145/3185517
- [53] K. Early, S. E. Fienberg, and J. Mankoff. Test time feature ordering with FOCUS: Interactive predictions with minimal user burden. In *Proc. UbiComp*, pp. 992–1003. ACM, New York, 2016. doi: 10.1145/2971648.2971748
- [54] B. Eilam. Possible constraints of visualization in biology: Challenges in learning with multiple representations. In *Multiple representations in biological education*, pp. 55–73. Springer, 12 2012. doi: 10.1007/978-94-007-4192-8_4
- [55] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the Natio-*

- nal Academy of Sciences*, 95(25):14863–14868, Dec. 1998. doi: 10.1073/pnas.95.25.14863
- [56] O. Elek, J. N. Burchett, J. X. Prochaska, and A. G. Forbes. Polyphorm: structural analysis of cosmological datasets via interactive physarum polycephalum visualization. *IEEE Trans Vis Comput Graph*, 27(2):806–816, 2021. doi: 10.1109/TVCG.2020.3030407
- [57] N. Elmqvist, U. Assarsson, and P. Tsigas. Employing dynamic transparency for 3D occlusion management: Design issues and evaluation. In *Proc. INTERACT*, pp. 532–545. Springer, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-74796-3_54
- [58] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the Dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans Vis Comput Graph*, 14(6):1139–1148, Nov./Dec. 2008. doi: 10.1109/TVCG.2008.153
- [59] N. Elmqvist and P. Tsigas. A taxonomy of 3D occlusion management for visualization. *IEEE Trans Vis Comput Graph*, 14(5):1095–1109, 6 2008. doi: 10.1109/TVCG.2008.59
- [60] N. Elmqvist and M. E. Tudoreanu. Occlusion management in immersive and desktop 3D virtual environments: Theory and evaluation. *The International Journal of Virtual Reality*, 6(2):21–32, Mar. 2007.
- [61] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Comput Graph Forum*, 36(8):458–486, Dec. 2017. doi: 10.1111/cgf.13092
- [62] L. Errera. Sur une condition fondamentale d'équilibre des cellules vivantes. *Annales de la Société belge de microscopie (1876)*, 13(1) :12–16, 1886.
- [63] P. Eulzer, S. Bauer, F. Kilian, and K. Lawonn. Visualization of human spine biomechanics for spinal surgery. *IEEE Trans Vis Comput Graph*, 27(2):700–710, Feb. 2021. doi: 10.1109/TVCG.2020.3030388
- [64] P. Eulzer, S. Engelhardt, N. Lichtenberg, R. De Simone, and K. Lawonn. Temporal views of flattened mitral valve geometries. *IEEE Trans Vis Comput Graph*, 26(1):971–980, Jan. 2020. doi: 10.1109/TVCG.2019.2934337
- [65] P. Eulzer, M. Meuschke, C. M. Klingner, and K. Lawonn. Visualizing carotid blood flow simulations for stroke prevention. In *Comput Graph Forum*, vol. 40, pp. 435–446. Wiley Online Library, 2021. doi: 10.1111/cgf.14319
- [66] G. Fouché, F. Argelaguet, E. Faure, and C. Kervrann. Immersive and interactive visualization of 3D spatio-temporal data using a space time hypercube, 2022. doi: 10.48550/ARXIV.2206.13213

- [67] S. Fukatsu, Y. Kitamura, T. Masaki, and F. Kishino. Intuitive control of “Bird’s Eye” overview images for navigation in an enormous virtual environment. In *Proc. VRST*, pp. 67–76. ACM, New York, 1998. doi: 10.1145/293701.293710
- [68] K. Furmanová, M. Jarešová, J. Byška, A. Jurčík, J. Parulek, H. Hauser, and B. Kozlíková. Interactive exploration of ligand transportation through protein tunnels. *BMC bioinformatics*, 18(2):1–16, 2 2017. doi: 10.1186/s12859-016-1448-0
- [69] G. W. Furnas. Generalized fisheye views. *ACM SIGCHI Bulletin*, 17(4):16–23, Apr. 1986. doi: 10.1145/22339.22342
- [70] L. A. Garrison, I. Kolesar, I. Viola, H. Hauser, and S. Bruckner. Trends & opportunities in visualization for physiology: A multiscale overview. In *Comput Graph Forum*, vol. 41, pp. 609–643. Wiley Online Library, 2022. doi: 10.1111/cgf.14575
- [71] A. Gekker. (Mini) mapping the game-space: A taxonomy of control. In C. Wilmott, C. Perkins, S. Lammes, S. Hind, A. Gekker, E. Fraser, and D. Evans, eds., *Playfyl Mapping in the Digital Age*, chap. 8, pp. 134–155. Institute of Network Cultures, Amsterdam, 2016.
- [72] 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*, pp. 614–624. ACM, New York, 2019. doi: 10.1145/3301275.3302324
- [73] K. Gooh, M. Ueda, K. Aruga, J. Park, H. Arata, T. Higashiyama, and D. Kurihara. Live-cell imaging and optical manipulation of Arabidopsis early embryogenesis. *Developmental Cell*, 34(2):242–251, 7 2015. doi: 10.1016/j.devcel.2015.06.008
- [74] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In *Proc. UIST*, pp. 3–12. ACM, New York, 2006. doi: 10.1145/1166253.1166257
- [75] L. Guignard, U.-M. Fiúza, B. Leggio, J. Laussu, E. Faure, G. Michelin, K. Biasuz, L. Hufnagel, G. Malandain, C. Godin, et al. Contact area-dependent cell communication and the morphological invariance of ascidian embryogenesis. *Science*, 369(6500):eaar5663, 7 2020. doi: 10.1126/science.aar5663
- [76] J. Guo, D.-M. Yan, E. Li, W. Dong, P. Wonka, and X. Zhang. Illustrating the disassembly of 3D models. *Comput Graph*, 37(6):574–581, Oct. 2013. doi: 10.1016/j.cag.2013.05.020
- [77] S. Halladjian, D. Kouřil, H. Miao, M. E. Gröller, I. Viola, and T. Isenberg. Multiscale unfolding: Illustratively visualizing the whole genome at a glance. *IEEE Trans Vis Comput Graph*, 28, 2022. doi: 10.1109/TVCG.2021.3065443

- [78] S. Halladjian, H. Miao, D. Kouřil, M. E. Gröller, I. Viola, and T. Isenberg. ScaleTrotter: Illustrative visual travels across negative scales. *IEEE Trans Vis Comput Graph*, 26(1):654–664, Jan. 2020. doi: 10.1109/TVCG.2019.2934334
- [79] C. D. Hansen and C. R. Johnson. *Visualization handbook*. Elsevier, 2011.
- [80] 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*, pp. 224–228, 1993.
- [81] O. Hertwig. *Das Problem der Befruchtung und der Isotropie des Eies : eine Theorie der Vererbung*, vol. 18. Fischer, 1884.
- [82] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. Passive real-world interface props for neurosurgical visualization. In *Proc. CHI*, pp. 452–458. ACM, New York, 1994. doi: 10.1145/191666.191821
- [83] J. Hong, F. Argelaguet, A. Trubuil, and T. Isenberg. Design and evaluation of three selection techniques for tightly packed 3D objects in cell lineage specification in botany. In *Proc. GI*, pp. 213–223. CHCCS, Mississauga, ON, Canada, 2021. doi: 10.20380/GI2021.33
- [84] 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*, 41(3):195–207, 2022. doi: 10.1111/cgf.14533
- [85] M. S. Horn, M. Tobiasz, and C. Shen. Visualizing biodiversity with Voronoi treemaps. In *Proc. ISVD*, pp. 265–270. IEEE CS, Los Alamitos, 2009. doi: 10.1109/ISVD.2009.22
- [86] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. CHI*, pp. 159–166. ACM, New York, 1999. doi: 10.1145/302979.303030
- [87] S. Houde. Iterative design of an interface for easy 3-D direct manipulation. In *Proc. CHI*, pp. 135–142. ACM, New York, 1992. doi: 10.1145/142750.142772
- [88] C. Hurter, N. H. Riche, S. M. Drucker, M. Cordeil, R. Alligier, and R. Vuillemot. FiberClay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Trans Vis Comput Graph*, 25(1):704–714, 2019. doi: 10.1109/TVCG.2018.2865191
- [89] 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*, 24(1):88–97, 8 2018. doi: 10.1109/TVCG.2017.2744718
- [90] T. Kapler and W. Wright. GeoTime information visualization. *Information visualization*, 4(2):136–146, 2005. doi: 10.1057/palgrave.ivs.9500097
- [91] F. Karray, M. Alemzadeh, J. Abou Saleh, and M. N. Arab. Human-computer interaction: Overview on state of the art. *International journal on smart*

- sensing and intelligent systems*, 1(1):137–159, Dec. 2017. doi: 10.21307/ijssis-2017-283
- [92] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual Analytics: Definition, process, and challenges. In *Information Visualization*, pp. 154–175. Springer, Berlin, 2008. doi: 10.1007/978-3-540-70956-5_7
- [93] D. A. Keim, T. Munzner, F. Rossi, and M. Verleysen. Bridging information visualization with machine learning. Dagstuhl Seminar Report 15101, Germany, 2015. doi: 10.4230/DagRep.5.3.1
- [94] G. Kim, S. Dhar, and J. Lim. The SHORT-ROOT regulatory network in the endodermis development of arabidopsis roots and shoots. *Journal of Plant Biology*, 60(4):306–313, Aug. 2017. doi: 10.1007/s12374-017-0134-8
- [95] K. N. Kirby and D. Gerlanc. BootES: An R package for bootstrap confidence intervals on effect sizes. *Behavior Research Methods*, 45(4):905–927, Mar. 2013. doi: 10.3758/s13428-013-0330-5
- [96] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *IEEE Softw*, 12(4):52–62, July 1995. doi: 10.1109/52.391832
- [97] T. Klein, F. Guéniat, L. Pastur, F. Vernier, and T. Isenberg. A design study of direct-touch interaction for exploratory 3D scientific visualization. *Comput Graph Forum*, 31(3):1225–1234, June 2012. doi: 10.1111/j.1467-8659.2012.03115.x
- [98] B. Köhler, M. Grothoff, M. Gutberlet, and B. Preim. Visual and quantitative analysis of great arteries’ blood flow jets in cardiac 4D PC-MRI data. In *Comput Graph Forum*, vol. 37, pp. 195–204. Wiley Online Library, 2018. doi: 10.1111/cgf.13412
- [99] N. Kong, J. Heer, and M. Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Trans Vis Comput Graph*, 16(6):990–998, Nov./Dec. 2010. doi: 10.1109/TVCG.2010.186
- [100] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3D selection by progressive refinement. In *Proc. 3DUI*, pp. 67–74. IEEE CS, Los Alamitos, 2011. doi: 10.1109/3DUI.2011.5759219
- [101] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. p. 3–24, 2007. doi: 10.5555/1566770.1566773
- [102] D. Kouřil, T. Isenberg, B. Kozlíková, M. Meyer, M. E. Gröller, and I. Viola. HyperLabels—Browsing of dense and hierarchical molecular 3D models. *IEEE Trans Vis Comput Graph*, 27(8):3493–3504, Aug. 2021. doi: 10.1109/TVCG.2020.2975583
- [103] M. Krzywinski and N. Altman. Points of significance: Error bars. *Nature Methods*, 10(10):921–922, Oct. 2013. doi: 10.1038/nmeth.2659

- [104] 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*, 24(1):142–151, 2018. doi: 10.1109/TVCG.2017.2745085
- [105] J.-B. Lamy and R. Tsopra. RainBio: Proportional visualization of large sets in biology. *IEEE Trans Vis Comput Graph*, 26(11):3285–3298, 6 2020. doi: 10.1109/TVCG.2019.2921544
- [106] A. G. Landge, J. A. Levine, A. Bhatele, K. E. Isaacs, T. Gamblin, M. Schulz, S. H. Langer, P.-T. Bremer, and V. Pascucci. Visualizing network traffic to understand the performance of massively parallel simulations. *IEEE Trans Vis Comput Graph*, 18(12):2467–2476, 10 2012. doi: 10.1109/TVCG.2012.286
- [107] J. J. LaViola, Jr., E. Kruijff, R. P. McMahan, D. A. Bowman, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, Boston, 2nd ed., 2017.
- [108] B. Lee, C. S. Parr, D. Campbell, and B. B. Bederson. How users interact with biodiversity information using TaxonTree. In *Proc. AVI*, pp. 320–327. ACM, New York, 2004. doi: 10.1145/989863.989918
- [109] B. Lee, G. G. Robertson, M. Czerwinski, and C. S. Parr. CandidTree: Visualizing structural uncertainty in similar hierarchies. *Inf Vis*, 6(3):233–246, Dec. 2007. doi: 10.1145/1375939.1375944
- [110] B. Leggio, J. Laussu, A. Carlier, C. Godin, P. Lemaire, and E. Faure. MorphoNet: an interactive online morphological browser to explore complex multi-scale data. *Nature communications*, 10(1):1–8, 6 2019. doi: 10.1038/s41467-019-10668-1
- [111] B. Leggio, J. Laussu, E. Faure, P. Lemaire, and C. Godin. Multiscale mechanical model for cell division orientation in developing biological systems. *bioRxiv*, p. 785337, 2020. doi: 10.1101/785337
- [112] D. Legland, I. Arganda-Carreras, and P. Andrey. MorphoLibJ: integrated library and plugins for mathematical morphology with imagej. *Bioinformatics*, 32(22):3532–3534, 7 2016. doi: 10.1093/bioinformatics/btw413
- [113] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan. Tree Illustrator: Interactive construction of tree visualizations. In *CHI Extended Abstracts*, pp. 1–4. ACM, New York, 2020. doi: 10.1145/3334480.3383150
- [114] W. Li, M. Agrawala, B. Curless, and D. Salesin. Automated generation of interactive 3D exploded view diagrams. *ACM Transactions on Graphics*, 27(3):101:1–101:7, Aug. 2008. doi: 10.1145/1360612.1360700
- [115] J. Liang and M. Green. JDCAD: A highly interactive 3D modeling system. *Computers & Graphics*, 18(4):499–506, July/Aug. 1994. doi: 10.1016/0097-8493(94)90062-0

- [116] D. Limberger, W. Scheibel, M. Trapp, and J. Döllner. Mixed-projection tree-maps: A novel approach mixing 2D and 2.5D treemaps. In *Proc. IV*, pp. 164–169. IEEE CS, Los Alamitos, 2017. doi: 10.1109/iV.2017.67
- [117] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu. Analyzing the training processes of deep generative models. *IEEE Trans Vis Comput Graph*, 24(1):77–87, 2018. doi: 10.1109/TVCG.2017.2744938
- [118] 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*, 23(1):91–100, 2017. doi: 10.1109/TVCG.2016.2598831
- [119] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48–56, 3 2017. doi: 10.1016/j.visinf.2017.01.006
- [120] M. Louveaux, J.-D. Julien, V. Mirabet, A. Boudaoud, and O. Hamant. Cell division plane orientation based on tensile stress in arabidopsis thaliana. *Proceedings of the National Academy of Sciences*, 113(30):E4294–E4303, 2016. doi: 10.1073/pnas.1600677113
- [121] Y. Lu, C. Yu, and Y. Shi. Investigating bubble mechanism for ray-casting to improve 3D target acquisition in virtual reality. In *Proc. VR*, pp. 35–43. IEEE CS, Los Alamitos, 2020. doi: 10.1109/VR46266.2020.00021
- [122] J. F. Lucas, D. A. Bowman, J. Chen, and C. A. Wingrave. Design and evaluation of 3D multiple object selection techniques. Report, Virginia Polytechnic Institute and State University, USA, 2005.
- [123] C. Lundström, T. Rydell, C. Forsell, A. Persson, and A. Ynnerman. Multi-touch table system for medical visualization: Application to orthopedic surgery planning. *IEEE Trans Vis Comput Graph*, 17(12):1775–1784, Dec. 2011. doi: 10.1109/TVCG.2011.224
- [124] P. Martinez, L. A. Allsman, K. A. Brakke, C. Hoyt, J. Hayes, H. Liang, W. Nether, Y. Rui, A. M. Roberts, A. Moradifam, et al. Predicting division planes of three-dimensional cells by soap-film minimization. *The Plant Cell*, 30(10):2255–2266, 2018. doi: 10.1105/tpc.18.00401
- [125] T. B. Masood, S. S. Thygesen, M. Linares, A. I. Abrikosov, V. Natarajan, and I. Hotz. Visual analysis of electronic densities and transitions in molecules. In *Comput Graph Forum*, vol. 40, pp. 287–298. Wiley Online Library, 2021. doi: 10.1111/cgf.14307
- [126] J. D. Mauseth. *Botany : an introduction to plant biology*. Jones & Bartlett Publishers, 2014.
- [127] K. McDole, L. Guignard, F. Amat, A. Berger, G. Malandain, L. A. Royer, S. C. Turaga, K. Branson, and P. J. Keller. In toto imaging and reconstruction

- of post-implantation mouse development at the single-cell level. *Cell*, 175(3):859–876, 2018. doi: 10.1016/j.cell.2018.09.031
- [128] M. J. McGuffin and J.-M. Robert. Quantifying the space-efficiency of 2D graphical representations of trees. *Inf Vis*, 9(2):115–140, June 2010. doi: 10.1145/1890886.1890889
- [129] S. McLellan, A. Muddimer, and S. C. Peres. The effect of experience on System Usability Scale ratings. *Journal of usability studies*, 7(2) :56–67, 2012.
- [130] M. Meuschke, S. Voss, O. Beuing, B. Preim, and K. Lawonn. Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms. *IEEE Trans Vis Comput Graph*, 23(1):761–770, 2017. doi: 10.1109/TVCG.2016.2598795
- [131] H. Miao, T. Klein, D. Kouřil, P. Mindek, K. Schatz, M. E. Gröllner, B. Kozlíková, T. Isenberg, and I. Viola. Multiscale molecular visualization. *Journal of molecular biology*, 431(6):1049–1070, 2019. doi: 10.1016/j.jmb.2018.09.004
- [132] R. Miller, V. Mozhayskiy, L. Tagkopoulos, and K.-L. Ma. EVEVis: A multi-scale visualization system for dense evolutionary data. In *2011 IEEE Symposium on Biological Data Visualization (BioVis)*, pp. 143–150, 2011. doi: 10.1109/BioVis.2011.6094059
- [133] N. Minc and M. Piel. Predicting division plane position and orientation. *Trends in cell biology*, 22(4):193–200, 2012. doi: 10.1016/j.tcb.2012.01.003
- [134] P. Mindek, D. Kouřil, J. Sorger, D. Toloudis, B. Lyons, G. Johnson, M. E. Gröllner, and I. Viola. Visualization multi-pipeline for communicating biology. *IEEE Trans Vis Comput Graph*, 24(1):883–892, 2018. doi: 10.1109/TVCG.2017.2744518
- [135] M. R. Mine. Virtual environment interaction techniques. Technical Report 95-018, Univ. of North Carolina at Chapel Hill, USA, Jan. 1995.
- [136] T. M. Mitchell and T. M. Mitchell. *Machine learning*, vol. 1. McGraw-hill New York, 1997.
- [137] N. J. Mitra, Y.-L. Yang, D.-M. Yan, W. Li, M. Agrawala, et al. Illustrating how mechanical assemblies work. *ACM Transactions on Graphics-TOG*, 29(4):58, July 2010. doi: 10.1145/1833349.1778795
- [138] H. Mohammed, A. K. Al-Awami, J. Beyer, C. Cali, P. Magistretti, H. Pfister, and M. Hadwiger. Abstractocyte: A visual tool for exploring nanoscale astroglial cells. *IEEE Trans Vis Comput Graph*, 24(1):853–861, 8 2018. doi: 10.1109/TVCG.2017.2744278
- [139] R. Mohammed, J. Rawashdeh, and M. Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experi-

- mental results. In *Proc. ICICS*, pp. 243–248. IEEE CS, Los Alamitos, 2020. doi: 10.1109/ICICS49469.2020.239556
- [140] M. Mohenska, N. M. Tan, A. Tokolyi, M. B. Furtado, M. W. Costa, A. J. Perry, J. Hatwell-Humble, K. van Duijvenboden, H. T. Nim, Y. M. Ji, et al. 3D-cardiomics: A spatial transcriptional atlas of the mammalian heart. *Journal of molecular and cellular cardiology*, 163:20–32, 2 2022. doi: 10.1016/j.yjmcc.2021.09.011
- [141] M. R. Morris, A. Danielescu, S. Drucker, D. Fisher, B. Lee, m. c. schraefel, and J. O. Wobbrock. Reducing legacy bias in gesture elicitation studies. *Interactions*, 21(3):40–45, May 2014. doi: 10.1145/2591689
- [142] J. Moukhtar, A. Trubuil, K. Belcram, D. Legland, Z. Khadir, A. Urbain, J.-C. Palauqui, and P. Andrey. Cell geometry determines symmetric and asymmetric division plane selection in arabidopsis early embryos. *PLOS Computational Biology*, 15(2):e1006771:1–e1006771:27, Feb. 2019. doi: 10.1371/journal.pcbi.1006771
- [143] T. Munzner. *Visualization analysis and design*. CRC press, 2014.
- [144] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans Graph*, 22(3):453–462, July 2003. doi: 10.1145/882262.882291
- [145] A. Olwal, H. Benko, and S. Feiner. SenseShapes: Using statistical geometry for object selection in a multimodal augmented reality. In *Proc. ISMAR*, pp. 300–301. IEEE CS, Los Alamitos, 2003. doi: 10.1109/ISMAR.2003.1240730
- [146] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Trans Graph*, 31(4):30:1–30:11, July 2012. doi: 10.1145/2185520.2185526
- [147] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim. An approach to supporting incremental visual data classification. *IEEE Trans Vis Comput Graph*, 21(1):4–17, 2015. doi: 10.1109/TVCG.2014.2331979
- [148] G. A. Pavlopoulos, S. I. O’Donoghue, V. P. Satagopam, T. G. Soldatos, E. Pafilis, and R. Schneider. Arena3D: visualization of biological networks in 3D. *BMC systems biology*, 2(1):1–7, 2008. doi: 10.1186/1752-0509-2-104
- [149] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vianova. DeepEyes: Progressive visual analytics for designing deep neural networks. *IEEE Trans Vis Comput Graph*, 24(1):98–108, 2018. doi: 10.1109/TVCG.2017.2744358
- [150] A. Pierre, J. Sallé, M. Wühr, and N. Minc. Generic theoretical models to predict division patterns of cleaving embryos. *Developmental cell*, 39(6):667–682, Dec. 2016. doi: 10.1016/j.devcel.2016.11.018

- [151] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The Go-Go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proc. UIST*, pp. 79–80. ACM, New York, 1996. doi: 10.1145/237091.237102
- [152] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Comput Graph Forum*, 17(3):41–52, Sept. 1998. doi: 10.1111/1467-8659.00252
- [153] C. G. Rasmussen, J. A. Humphries, and L. G. Smith. Determination of symmetric and asymmetric division planes in plant cells. *Annual review of plant biology*, 62:387, 2011. doi: 10.1146/annurev-arplant-042110-103802
- [154] C. G. Rasmussen, A. J. Wright, and S. Müller. The role of the cytoskeleton and associated proteins in determination of the plant cell division plane. *The Plant Journal*, 75(2):258–269, 2013. doi: 10.1111/tpj.12177
- [155] H. Ro, S. Chae, I. Kim, J. Byun, Y. Yang, Y. Park, and T. Han. A dynamic depth-variable ray-casting interface for object manipulation in AR environments. In *Proc. SMC*, pp. 2873–2878. IEEE CS, Los Alamitos, 2017. doi: 10.1109/SMC.2017.8123063
- [156] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. CHI*, pp. 189–194. ACM, New York, 1991. doi: 10.1145/108844.108883
- [157] A. Rosset, L. Spadola, and O. Ratib. OsiriX: An open-source software for navigating in multidimensional DICOM images. *J Digital Imaging*, 17(3):205–216, Sept. 2004. doi: 10.1007/s10278-004-1014-6
- [158] D. Sacha, M. Kraus, D. A. Keim, and M. Chen. VIS4ML: An ontology for visual analytics assisted machine learning. *IEEE Trans Vis Comput Graph*, 25(1):385–395, 8 2019. doi: 10.1109/TVCG.2018.2864838
- [159] I. Salvador-Martínez, M. Grillo, M. Averof, and M. J. Telford. CeLaVi: an interactive cell lineage visualization tool. *Nucleic acids research*, 49(W1):W80–W85, 2021. doi: 10.1093/nar/gkab325
- [160] E. A. D. Santos, J. Hong, and T. Isenberg. A design space for linked 2D and 3D visual representations. In *Posters at the IEEE Conference on Visualization (IEEE VIS, October 16–21, Oklahoma City, USA)*, 2022.
- [161] M. Schedl, P. Knees, G. Widmer, K. Seyerlehner, and T. Pohle. Browsing the web using stacked three-dimensional sunbursts to visualize term co-occurrences and multimedia content. In *Posters of IEEE VIS*, 2007.
- [162] 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*, 9(7):676–682, July 2012. doi: 10.1038/nmeth.2019
- [163] W. Schroeder, K. Martin, and W. Lorensen. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *Proc. Visualization*, pp. 93–100. IEEE CS, Los Alamitos, 1996. doi: 10.1109/VISUAL.1996.567752
- [164] W. J. Schroeder, B. Lorensen, and K. Martin. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, New York, 2004.
- [165] H. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, Nov./Dec. 2011. doi: 10.1109/MCG.2011.103
- [166] H. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE Trans Vis Comput Graph*, 17(4):393–411, Apr. 2011. doi: 10.1109/TVCG.2010.79
- [167] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans Vis Comput Graph*, 18(12):2431–2440, Dec. 2012. doi: 10.1109/TVCG.2012.213
- [168] A. Semmo, M. Trapp, J. E. Kyprianidis, and J. Döllner. Interactive visualization of generalized virtual 3D city models using level-of-abstraction transitions. In *Comput Graph Forum*, vol. 31, pp. 885–894. Wiley Online Library, 6 2012. doi: 10.1111/j.1467-8659.2012.03081.x
- [169] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Comput*, 35(7):80–86, 2002. doi: 10.1109/MC.2002.1016905
- [170] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003. doi: 10.1101/gr.1239303
- [171] H.-W. Shen. Isosurface extraction in time-varying fields using a temporal hierarchical index tree. In *Proc. Visualization*, pp. 159–166. IEEE CS, Los Alamitos, 1998. doi: 10.1109/VISUAL.1998.745298
- [172] K. Shi, P. Irani, and B. Li. An evaluation of content browsing techniques for hierarchical space-filling visualizations. In *Proc. InfoVis*, pp. 81–88. IEEE CS, Los Alamitos, 2005. doi: 10.1109/INFVIS.2005.1532132
- [173] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pp. 364–371. Elsevier, 2003. doi: 10.1016/B978-155860915-0/50046-9
- [174] K. Siau and W. Wang. Building trust in artificial intelligence, machine learning, and robotics. *Cutter Bus Technol J*, 31(2):47–53, 2018.

- [175] B. Speckmann and K. Verbeek. Necklace maps. *IEEE Trans Vis Comput Graph*, 16(6):881–889, 2010. doi: 10.1109/TVCG.2010.180
- [176] 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*, 26(1):1064–1074, 8 2020. doi: 10.1109/TVCG.2019.2934629
- [177] A. Steed. Towards a general model for selection in virtual environments. In *Proc. VR*, pp. 103–110. IEEE CS, Los Alamitos, 2006. doi: 10.1109/VR.2006.134
- [178] F. Steinicke, T. Ropinski, and K. Hinrichs. Object selection in virtual environments using an improved virtual pointer metaphor. In K. Wojciechowski, B. Smolka, H. Palus, R. S. Kozera, W. Skarbek, and L. Noakes, eds., *Computer Vision and Graphics*, vol. 32 of *Computational Imaging and Vision*, pp. 320–326. Springer, Dordrecht, 2006. doi: 10.1007/1-4020-4179-9_46
- [179] G. Stent. The role of cell lineage in development. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 312(1153) :3–19, Dec. 1985.
- [180] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. CHI*, pp. 265–272. ACM, New York, 1995. doi: 10.1145/223904.223938
- [181] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Trans Vis Comput Graph*, 20(12):1653–1662, Dec. 2014. doi: 10.1109/TVCG.2014.2346574
- [182] H. Strobel, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush. Seq2Seq-Vis: A visual debugging tool for sequence-to-sequence models. *IEEE Trans Vis Comput Graph*, 25(1):353–363, 2019. doi: 10.1109/TVCG.2018.2865044
- [183] H. Strobel, 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*, 24(1):667–676, 2018. doi: 10.1109/TVCG.2017.2744158
- [184] N. Sultanum, S. Somanath, E. Sharlin, and M. C. Sousa. “Point it, split it, peel it, view it” Techniques for interactive reservoir visualization on tabletops. In *Proc. ITS*, pp. 192–201. ACM, New York, 2011. doi: 10.1145/2076354.2076390
- [185] M. Sun, A. Namburi, D. Koop, J. Zhao, T. Li, and H. Chung. Towards systematic design considerations for visualizing cross-view data relationships. *IEEE Trans Vis Comput Graph*, 28(12):4741–4756, Dec. 2022. doi: 10.1109/TVCG.2021.3102966

- [186] B. Szigeti, P. Gleeson, M. Vella, S. Khayrulin, A. Palyanov, J. Hokanson, M. Currie, M. Cantarelli, G. Idili, and S. Larson. OpenWorm: An open-science approach to modeling *Caenorhabditis elegans*. *Frontiers in Computational Neuroscience*, 8:137:1–137:7, Nov. 2014. doi: 10.3389/fncom.2014.00137
- [187] E. Talevich, B. M. Invergo, P. J. Cock, and B. A. Chapman. Bio. Phylo: a unified toolkit for processing, analyzing and visualizing phylogenetic trees in biopython. *BMC bioinformatics*, 13(1):1–9, 8 2012. doi: 10.1186/1471-2105-13-209
- [188] G. K. Tam, V. Kothari, and M. Chen. An analysis of machine-and human-analytics in classification. *IEEE Trans Vis Comput Graph*, 23(1):71–80, 2017. doi: 10.1109/TVCG.2016.2598829
- [189] D. Tan, G. Smith, B. Lee, and G. Robertson. AdaptiviTree: Adaptive tree visualization for tournament-style brackets. *IEEE Trans Vis Comput Graph*, 13(6):1113–1120, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70537
- [190] Y. Tanaka, Y. Okada, and K. Nijjima. Treecube: Visualization tool for browsing 3D multimedia data. In *Proc. IV*, pp. 427–432. IEEE CS, Los Alamitos, 2003. doi: 10.1109/IV.2003.1218020
- [191] M. Tatzgern, D. Kalkofen, and D. Schmalstieg. Multi-perspective compact explosion diagrams. *Computers & Graphics*, 35(1):135–147, Feb. 2011. doi: 10.1016/j.cag.2010.11.005
- [192] E. Tejada, R. Minghim, and L. G. Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization*, 2(4):218–231, 12 2003. doi: 10.1057/palgrave.ivs.9500054
- [193] M. Tennekes and E. de Jonge. Tree colors: color schemes for tree-structured data. *IEEE Trans Vis Comput Graph*, 20(12):2072–2081, 2014. doi: 10.1109/TVCG.2014.2346277
- [194] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *nature*, 381(6582):520–522, 6 1996. doi: 10.1038/381520a0
- [195] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Trans Vis Comput Graph*, 18(12):2565–2574, 10 2012. doi: 10.1109/TVCG.2012.265
- [196] E. Truernit, H. Bauby, B. Dubreucq, O. Grandjean, J. Runions, J. Barthélemy, and J.-C. Palauqui. High-resolution whole-mount imaging of three-dimensional tissue organization and gene expression enables the study of phloem development and structure in arabidopsis. *The Plant Cell*, 20(6):1494–1503, 2008. doi: 10.1105/tpc.107.056069
- [197] Y. Tu and H. Shen. Visualizing changes of hierarchical data using treemaps. *IEEE Trans Vis Comput Graph*, 13(6):1286–1293, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70529

- [198] H. van de Wetering, N. Klaassen, and M. Burch. Space-reclaiming icicle plots. In *Proc. PacificVis*, pp. 121–130. IEEE CS, Los Alamitos, 2020. doi: 10.1109/PacificVis48177.2020.4908
- [199] M. Veit, A. Capobianco, and D. Bechmann. Influence of degrees of freedom’s manipulation on performances during orientation tasks in virtual reality environments. In *Proc. VRST*, pp. 51–58. ACM, New York, 2009. doi: 10.1145/1643928.1643942
- [200] A. Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural computing and applications*, 32(24):18069–18083, 12 2020. doi: 10.1007/s00521-019-04051-w
- [201] O. Viader-Llargués, V. Lupperger, L. Pola-Morell, C. Marr, and H. López-Schier. Live cell-lineage tracing and machine learning reveal patterns of organ regeneration. *Elife*, 7:e30823, 2018. doi: 0.7554/eLife.30823.001
- [202] P. Villoutreix. What machine learning can do for developmental biology. *Development*, 148(1):dev188474, 2021. doi: 10.1242/dev.188474
- [203] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Comput Graph Forum*, vol. 30, pp. 1719–1749. Wiley Online Library, 2011. doi: 10.1111/j.1467-8659.2011.01898.x
- [204] R. Wang, Y. Perez-Riverol, H. Hermjakob, and J. A. Vizcaíno. Open source libraries and frameworks for biological data visualisation: A guide for developers. *Proteomics*, 15(8):1356–1374, 12 2015. doi: 10.1002/pmic.201400377
- [205] X. Wang, L. Besançon, D. Rousseau, M. Sereno, M. Ammi, and T. Isenberg. Towards an understanding of augmented reality extensions for existing 3D data analysis tools. In *Proc. CHI*, pp. 528:1–528:13. ACM, New York, 2020. doi: 10.1145/3313831.3376657
- [206] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE Trans Vis Comput Graph*, 26(1):895–905, Jan. 2019. doi: 10.1109/TVCG.2019.2934398
- [207] G. J. Wills. Selection: 524,288 ways to say “This is interesting”. In *Proc. InfoVis*, pp. 54–60. IEEE CS, Los Alamitos, 1996. doi: 10.1109/INFVIS.1996.559216
- [208] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Trans Vis Comput Graph*, 24(1):1–12, 8 2018. doi: 10.1109/TVCG.2017.2744878
- [209] L. Woodburn, Y. Yang, and K. Marriott. Interactive visualisation of hierarchical quantitative data: An evaluation. In *Proc. VIS*, pp. 96–100. IEEE CS, Los Alamitos, 2019. doi: 10.1109/VISUAL.2019.8933545

- [210] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *Proc. VAST*, pp. 57–68. IEEE CS, Los Alamitos, 2019. doi: 10.1109/VAST47406.2019.8986943
- [211] W. Yang, X. Wang, J. Lu, W. Dou, and S. Liu. Interactive steering of hierarchical clustering. *IEEE Trans Vis Comput Graph*, 27(10):3953–3967, Oct. 2020. doi: 10.1109/TVCG.2020.2995100
- [212] Y. Yang, B. Jenny, T. Dwyer, K. Marriott, H. Chen, and M. Cordeil. Maps and globes in virtual reality. In *Comput Graph Forum*, vol. 37, pp. 427–438. Wiley Online Library, 7 2018. doi: 10.1111/cgf.13431
- [213] Y. Yao, Q. Dong, A. Brozena, J. Luo, J. Miao, M. Chi, C. Wang, I. G. Kevrekidis, Z. J. Ren, J. Greeley, G. Wang, A. Anapolsky, and L. Hu. High-entropy nanoparticles: Synthesis-structure-property relationships and data-driven discovery. *Science*, 376(6589):eabn3103, Apr. 2022. doi: 10.1126/science.abn3103
- [214] S. Ye, Z. Chen, X. Chu, Y. Wang, S. Fu, L. Shen, K. Zhou, and Y. Wu. ShuttleSpace: Exploring and analyzing movement trajectory in immersive visualization. *IEEE Trans Vis Comput Graph*, 27(2):860–869, 2021. doi: 10.1109/TVCG.2020.3030392
- [215] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. *IEEE Trans Vis Comput Graph*, 18(12):2245–2254, Dec. 2012. doi: 10.1109/TVCG.2012.217
- [216] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds. *IEEE Trans Vis Comput Graph*, 22(1):886–895, Jan. 2016. doi: 10.1109/TVCG.2015.2467202
- [217] C. Zhang, M. W. Caan, T. Höllt, E. Eisemann, and A. Vilanova. Overview + detail visualization for ensembles of diffusion tensors. In *Comput Graph Forum*, vol. 36, pp. 121–132. Wiley Online Library, 2017. doi: 10.1111/cgf.13173
- [218] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 57–64. IEEE, 10 2005. doi: 10.1109/INFVIS.2005.1532129
- [219] B. Zheng and F. Sadlo. On the visualization of hierarchical multivariate data. In *Proc. PacificVis*, pp. 136–145. IEEE, 2021. doi: 10.1109/PacificVis52677.2021.00026

Titre : Visualisation interactive, soutenue par l'apprentissage automatique, de données hybrides 3D et 2D ; l'exemple de la spécification du lignage cellulaire en biologie végétale

Mots clés : Données 3D, visualisation scientifique, visualisation abstraite 2D, interaction/HCI, apprentissage automatique

Avec le développement des technologies d'imagerie, les données spatiales peuvent être mieux visualisées dans leur environnement 3D afin que les spectateurs puissent observer clairement les formes et les positions 3D. Parallèlement, les visualisations abstraites en 2D peuvent présenter des informations résumées, visualiser des données supplémentaires et contrôler la vue 3D. La combinaison de ces deux représentations en une seule interface peut aider les utilisateurs à entreprendre des tâches complexes, en particulier dans les domaines scientifiques, bien qu'il y ait un manque de directives générales de conception pour l'interaction. En général, les experts doivent analyser de volumineuses données scientifiques pour mener à bien des tâches difficiles. Par exemple, dans le domaine biologique, les biologistes doivent construire l'arbre de lignage cellulaire d'un embryon contenant plus de 200 cellules. Dans ce cas, le travail manuel peut être long et fastidieux, et les algorithmes d'apprentissage automatique ont le potentiel d'alléger certains des processus manuels fastidieux en fournissant des annotations ou prédictions initiales aux experts. Dans le cas du lignage cellulaire, ces prédictions contiennent toutefois des informations hiérarchiques et multicouches, et il est essentiel de les visualiser de manière séquentielle ou progressive. De plus, les représentations 3D et 2D, ainsi que les prédictions d'apprentissage automatique, doivent être connectées visuellement et interactivement dans le système.

Dans cette thèse, le problème du lignage cellulaire des embryons de plantes a été le leitmotiv pour concevoir et étudier un système de visualisation qui utilise des combinaisons de représentations 3D et 2D ainsi que des visualisations pour l'apprentissage automatique. Nous avons d'abord étudié les techniques d'interaction pour la sélection 3D au sein d'un embryon de plante. Les cellules d'un embryon de plante sont jointives et

constituent un ensemble d'objets 3D dense dans toutes les dimensions spatiales. Nous avons mené une étude pour évaluer trois techniques de sélection différentes, et nous avons montré que la combinaison de la technique de Sélection par Explosion et de la technique de Sélection par Liste fonctionne bien pour désigner et observer les cellules d'un embryon. Ces techniques peuvent également être étendues à d'autres données 3D denses et similaires. Deuxièmement, nous avons conçu un système de visualisations et d'interaction combiné afin que les biologistes puissent examiner les cellules de l'embryon et enregistrer l'histoire du développement dans l'arbre de lignage hiérarchique. Nous prenons en charge la construction de la hiérarchie dans deux directions, à la fois en construisant l'historique de haut en bas de l'arbre en utilisant la sélection lasso dans l'environnement 3D et de bas en haut selon le flux de travail traditionnel pour construire un arbre de lignage cellulaire hiérarchique. Nous avons également ajouté un modèle de réseau neuronal pour fournir aux biologistes des prédictions initiales sur les filiations. Nous avons réalisé une évaluation avec des biologistes ; celle-ci a montré que les représentations 3D et 2D facilitent les prises de décisions et que l'outil peut enrichir leur vision des embryons. Cependant, la performance du modèle d'apprentissage automatique n'était pas idéale. Aussi, pour faciliter le processus et améliorer les performances du modèle, dans une version plus aboutie de notre système, nous avons entraîné cinq modèles de classification différents, visualisé les prédictions et leurs incertitudes associées. Nous avons réalisé une évaluation auprès des utilisateurs ; les résultats ont indiqué que les représentations des classificateurs que nous avons conçues sont faciles à comprendre, et que le nouvel outil peut améliorer significativement les prises de décision pour la validation du lignage cellulaire.

Title : Machine Learning Supported Interactive Visualization of Hybrid 3D and 2D Data for the Example of Plant Cell Lineage Specification

Keywords : 3D data, scientific visualization, 2D abstract visualization, interaction/HCI, machine learning

Abstract : As computer graphics technologies develop, spatial data can be better visualized in the 3D environment so that viewers can observe 3D shapes and positions clearly. Meanwhile, 2D abstract visualizations can present summarized information, visualize additional data, and control the 3D view. Combining these two parts in one interface can assist people in finishing complicated tasks, especially in scientific domains, though there is a lack of design guidelines for the interaction. Generally, experts need to analyze large scientific data to finish challenging tasks. For example, in the biological field, biologists need to build the hierarchy tree for an embryo with more than 200 cells. In this case, manual work can be time-consuming and tedious, and machine learning algorithms have the potential to alleviate some of the tedious manual processes to serve as the basis for experts. These predictions, however, contain hierarchical and multi-layer information, and it is essential to visualize them sequentially and progressively so that experts can control their viewing pace and validation. Also, 3D and 2D representations, together with machine learning predictions, need to be visually and interactively connected in the system.

In this thesis, we worked on the cell lineage problem for plant embryos as an example to investigate a visualization system and its interaction design that makes use of combinations of 3D and 2D representations as well as visualizations for machine learning. We first investigated the 3D selection interaction techniques for the plant embryo. The cells in a plant embryo are tightly packed together, without any space in between. Traditional

techniques can hardly deal with such an occlusion problem. We conducted a study to evaluate three different selection techniques, and found out that the combination of the Explosion Selection technique and the List Selection technique works well for people to get access and observe plant cells in an embryo. These techniques can also be extended to other similar densely packed 3D data. Second, we explored the visualization and interaction design to combine the 3D visualizations of a plant embryo with its associated 2D hierarchy tree. We designed a system with such combinations for biologists to examine the plant cells and record the development history in the hierarchy tree. We support the hierarchy building in two directions, both constructing the history top-down using the lasso selection in 3D environment and bottom-up as the traditional workflow does in the hierarchy tree. We also added a neural network model to give predictions about the assignments for biologists to start with. We conducted an evaluation with biologists, which showed that both 3D and 2D representations help with making decisions, and the tool can inspire insights for them. One main drawback was that the performance of the machine learning model was not ideal. Thus, to assist the process and enhance the model performance, in an improved version of our system, we trained five different ML models and visualized the predictions and their associated uncertainty. We performed a study, and the results indicated that our designed ML representations are easy to understand, and that the new tool can effectively improve the efficiency of assigning the cell lineage.