Benchmarked approaches for cell lineage reconstructions of *in vitro* dividing cells and *in silico* models of *Caenorhabditis elegans* and *Mus musculus* developmental trees.

Wuming Gong[1]*, Alejandro Granados[2]*, Jingyuan Hu[3]*, Matthew G Jones[4,5]*, Ofir Raz[6]*, Irepan Salvador-Martínez[7]*, Hanrui Zhang[8]*, Ke-Huan K. Chow[2], Il-Youp Kwak[9], Renata Retkute[10], Alidivinas Prusokas[11], Augustinas Prusokas[12], Alex Khodaverdian[4], Richard Zhang[4], Suhas Rao[4], Robert Wang[4], Phil Rennert[13],Vangala G. Saipradeep[14], Naveen Sivadasan[14], Aditya Rao[14], Thomas Joseph[14], Rajgopal Srinivasan[14], Jiajie Peng[15], Lu Han[15], Xuequn Shang[15], Daniel J. Garry[1], Thomas Yu[16], Verena Chung[16], Michael Mason[16], Zhandong Liu[3], Yuanfang Guan[8], Nir Yosef [4], Jay Shendure[17,18,19,20], Maximilian J. Telford[7], Ehud Shapiro[6], Michael B. Elowitz[2], Pablo Meyer[21+]

[1] Lillehei Heart Institute, University of Minnesota, 2231 6th St S.E, 4-165 CCRB, Minneapolis, MN 55114, USA.

[2] California Institute of Technology, Pasadena, CA, USA

[3] Program in Quantitative and Computational Biosciences, Baylor College of Medicine Houston, TX, USA

[4] Department of Electrical Engineering & Computer Science, University of California, Berkeley, CA, USA

[5] Integrative Program of Quantitative Biology, University of California, San Francisco.

[6] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 761001, Israel.

[7] Centre for Life's Origins and Evolution, Department of Genetics, Evolution and Environment, University College London, Gower Street, London, WC1E 6BT, UK.

[8] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, Michigan, USA

[9] Department of Applied Statistics, College of Business & Economics, Chung-Ang University, 84, Heukseok-ro, Dongjak-gu, Seoul, Republic of Korea

[10] Department of Plant Sciences, University of Cambridge, Downing Street, Cambridge CB2 3EA, UK

[11] School of Natural and Environmental Sciences, Newcastle University, Newcastle, NE1 7RU, UK

[12] Department of Life Sciences, Imperial College London, London, SW7 2AZ, UK

[13] EC Wise Inc.

[14] TCS Research and Innovation, Tata Consultancy Services Ltd., Hyderabad, INDIA

[15] School of Computer Science, Northwestern Polytechnical University, Xi'an, China

[16] Sage Bionetworks, Seattle, WA, USA

[17] Department of Genome Sciences, University of Washington, Seattle, WA, USA

[18] Allen Discovery Center for Cell Lineage Tracing, Seattle, WA, USA

[19] Brotman Baty Institute for Precision Medicine, Seattle, WA, USA

[20] Howard Hughes Medical Institute, Seattle, WA, USA

[21] T.J. Watson Research Center, IBM, Healthcare & Life Sciences, 1101 Kitchawan road 10598, Yorktown Heights, NY, USA

*Contributed equally and sorted in alphabetical order

+Corresponding author

Abstract

The recent advent of new CRISPR-based and other molecular tools now enables the reconstruction of cell lineages based on DNA mutations induced by CRISPR and promises to solve the lineage of complex model organisms at single-cell resolution. To date, however, no lineage reconstruction algorithms have been rigorously examined for their performance and robustness across datasets, diverse molecular tools, and most importantly the number of cells in the lineage tree. In order to benchmark methods of cell lineage reconstruction we decided to organize the Allen institute lineage reconstruction DREAM challenge where we rigorously examined multiple methods using experimental and *in silico* data. On one hand, we took advantage of intMEMOIR recordings, a recently developed synthetic image-readable lineage tracing technology, and asked participants to reconstruct the lineages for 30 *in vitro*-grown mouse embryonic stem cell colonies. We also provided *in silico* datasets for a *C. elegans* lineage tree of about 1000 cells and a simulation of one year of *Mus musculus* development down-sampled to 10,000 cells upon which we simulated CRISPR-based GESTALT-like recordings. For these three lineage reconstruction tasks we provided training data with the ground true trees, as one of the goals of this challenge was to encourage machine-learning approaches different from the ones used for phylogenetics. The challenge was successful in its main goal of attracting a variety of successful approaches and teams: twenty-two full submissions were received and scored using two different metrics. The availability of a training set allowed not only the development of a successful machine-learning decision-tree based approach, but also the optimization of accurate distance-based algorithms and maximum parsimony approaches. This DREAM challenge was a first attempt to rigorously examine the performance and robustness of various reconstruction algorithms under varying conditions and underlies the importance of using several metrics when evaluating reconstruction accuracy. For the experimental dataset, we found that while some trees were reconstructed perfectly, the overall scores were far from the theoretical maximum, mainly due to the structural features of the trees and not the high degeneracy in recorded states across cells. On the other hand, the *in silico* results showed that using smaller subtrees as training sets is a good approach for tuning the algorithms to reconstruct larger trees. Together, these results and the availability of tools for generating and solving lineage trees delineate a potential way forward for solving larger cell lineage trees such as for mouse and human.

Introduction

Lineage inference for understanding Development

A fundamental challenge in biology is the reconstruction of the developmental histories of cells as they divide and progress through differentiation into different cell types. Indeed, multicellular organisms can be composed of billions or trillions of cells that derive from a single cell through repeated rounds of cell division. Knowing the lineage relationships between the cells of a fully developed organism -its cell lineage- would provide a framework to understand when, where and how cell fate decisions are made. Further, it can also be useful to understand the progression of disease such as in tumor subclonal reconstruction (Salcedo et al., 2020) or the development of an organ such as the brain (Evrony et al., 2015; Lodato et al., 2015). Historically, lineages of individual cells have only been fully reconstructed by their direct observation through microscopy as for the nematode *Caenorhabditis elegans* (Sulston and Horvitz, 1977). This direct observation approach is however not possible for most animals as the cells are not visible (Livet et al., 2007). In the 1980's new methods allowed marking all the descendants of a single cell by the injection of a dye or the expression of a marker gene. Since then, many new methods have been devised to improve cell lineage tracking, including inducible recombinases (Kretzschmar and Watt, 2012), fluorescent or genetic reporters (Kebschull and Zador, 2018; Weissman and Pan, 2015), or a combination of both (Garcia-Marques et al., 2020). However, these approaches come at the cost of resolution, meaning that lineage relationships of individual cells are not fully recovered.

Recent advances in sequencing technologies have enabled a variety of RNA-based methods to infer differentiation trajectories in multiple organisms and cell types by ordering the changes in single-cell gene expression along a pseudo-time axis representing the progression through differentiation (Wagner and Klein, 2020). However, these methods focus on the expression profiles of cells but do not have access to their genealogical relationships. In this regard, somatic mutations accumulated during normal development have been used to reconstruct genetic lineages (Behjati et al., 2014; Frumkin et al., 2005) and for example trace mosaicism in the brain (Evrony et al., 2015; Lodato et al., 2015). Deep sequencing of cDNA from T cell receptors has also been used to establish clonal development of T cells (Becattini et al., 2015). Cell lineage inference has also been done using copy-number variations, structural markers such as SNVs, indels, retrotransposon elements, microsatellite repeats, as well as epigenetic markers such as DNA methylation (Kester and Oudenaarden, 2018).

New lineage recording technologies

Recently, the advent of CRISPR-based molecular tools have produced a new generation of lineage reconstruction approaches inspired by principles of phylogenetic inference using naturally occurring DNA mutations. The DNA-editing technologies have been applied to introduce mutations in the genetic material of cells such that a registry of their genetic relationships is recorded and available for readout by sequencing (Alemany et al., 2018; Chan et al., 2019; McKenna et al., 2016a; Perli et al., 2016; Spanjaard et al., 2018). Indeed, the inserted synthetic construct can accumulate stochastic mutations upon induction of CRISPR-Cas9 activity as cells differentiate during development with the goal of resolving cellular lineages of complex model organisms (McKenna et al., 2016b; Wagner and Klein, 2020). Different versions of CRISPR-based methods such as scGESTALT, LINNAEUS and ScarTrace techniques have been successfully used to investigate cellular lineages in various animal models (Alemany et al., 2018; McKenna and Gagnon, 2019; Raj et al., 2018; Spanjaard et al., 2017). At the same time, other types of lineage recording techniques have been applied to allow readout by *in situ* imaging which enables lineage analysis through the maintenance of the spatial information (Chow et al., 2021; McKenna and Gagnon, 2019). Some of these approaches have applied phylogenetic reconstruction algorithms to infer the cell lineage, whilst others developed ad-hoc cell lineage reconstruction algorithms, but this explosion of lineage tracing technologies has increased the urgency for new reconstruction methods (Salvador-Martínez et al., 2019).

In principle, as in phylogenetic tree reconstruction (Frieda et al., 2017; McKenna et al., 2016a), the recorded mutations should encode enough information enabling inference of the likely tree structures that could represent the actual lineage relationships. However, there are significant challenges for tree-inference when applying standard phylogenetic methods to lineage recordings. The main limitations include noise from the experimental readout, restrictions in the total available 'DNA memory' for recording, and the random convergence of identical edit patterns in non-related cells, or homoplasy (Salvador-Martínez et al., 2019). It also remains unclear whether machine-learning algorithms that go beyond classical phylogenetic methods, such as Neighbor-Joining or Maximum Parsimony, could consistently reconstruct cell lineages with higher accuracy. While phylogenetic methods typically analyze a relatively small number of species and many more DNA sites, genes or even whole genomes (McKenna and Gagnon, 2019), CRISPR-based lineage recording aims to capture hundreds to thousands of cells with the compromise of limited numbers of editable sites. Additional limitations include variability in mutation rates for each site, large nucleotide deletions resulting in sequence dropouts, and single deletions that can erase previous mutations or ablate multiple targets. Although maximum parsimony-based methods have shown initial success when applied to lineage tracing (McKenna and Gagnon, 2019; McKenna et al., 2016a; Price et al., 2010), the key differences discussed above make it challenging to directly apply phylogenetic methods to lineage tracing data.

After having performed lineage tree inference one would ideally like to evaluate the reconstruction accuracy, however for most of these technologies the ground truth is inaccessible, meaning that we do not know the actual lineage relationships. Indeed, with rare exceptions (Sugino et al., 2019), to date no lineage reconstruction approach has been rigorously examined for its performance/robustness across diverse molecular tools, DNA-based recording methods, datasets, number of cells, topology of lineage trees and diverse metrics used for evaluation. Given the lack of benchmarking, there is still no agreement regarding the best practices for inferring cellular lineages from the recording datasets generated with these recently developped molecular tools.

The DREAM initiative

To catalyze the development of new methods to perform lineage reconstruction, we organized the Allen institute lineage reconstruction DREAM challenge, which ran from October 2019 through February 2020. DREAM challenges are a platform for crowdsourcing collaborative competitions where a rigorous evaluation of each submitted solution allows for objective comparison and assessment of their performance (Saez-Rodriguez et al., 2016). The value of DREAM resides not only in the acceleration of research through the participation of many teams while solving a common problem, but just as importantly, in the diversity of approaches used and the quality and reproducibility of each provided solution to problems in emerging areas of biology. The aggregation of the individual solutions, *i.e.*, the different approaches and insights to a common problem, namely the ‘wisdom of the crowds’, leads to a generally superior performance than any individual solution, from where collective insights can be garnered.

The DREAM challenge for lineage reconstruction

The lineage reconstruction DREAM challenge aimed to provide a new perspective on lineage inference by enabling participants from diverse fields to submit their reconstruction of trees for which the ground truth, *i.e.* the actual lineage, existed but was not provided. It consisted of three challenges with lineages of increasing numbers of cells. The first challenge leveraged a then unpublished experimental dataset of 106 trees recorded with intMEMOIR in mouse embryonic stem cell colonies of less than 100 cells (Chow et al., 2021). This technique was chosen as it has the key advantage of readout by imaging which can be coupled with a time-lapse movie of the cells as they divide to provide a ground truth lineage tree (**Fig 1A**). In the second challenge participants had to reconstruct an *in silico* tree of 1,000 cells, whose topology was derived from the *Caenorhabditis elegans* developmental cell lineage tree by removing a few clades in order to mask its identity to the participants. A general framework for simulation of CRISPR-based lineage recording (**Fig 1B**) (Salvador-Martínez et al., 2019) was used to simulate mutations in a recording array on top of the resulting topology (see **Fig 1C**). In the third challenge, participants had to infer the lineage of cells in a simulated tree of ~10,000 cells (**Fig 1D**) representing 11 different cell types after one year of *M. musculus* development (**Fig 1E**). Simulating such a large tree was made possible by applying the Environment-dependent Stochastic tree Grammars (eSTGt), a programming and simulation environment for population dynamics (Spiro and Shapiro, 2016) adapted to simulate cell lineages (see STAR methods). While the size of the actual simulated tree is estimated to be about 10^{12} or a trillion cells, the final sub-sampled lineage stored information for only 10,000 cells (see **Fig S1**).

Experimental *in vitro* dataset

intMEMOIR is a synthetic image-readable lineage recording system that has been recently developed and tested in mouse embryonic stem cells and the brain of *Drosophila melanogaster* (Chow et al., 2021). This technology builds upon a previously developed recording system named MEMOIR (Memory by Engineered Mutagenesis with Optical In situ Readout) (Frieda et al., 2017). In its current implementation, intMEMOIR consists of a multi-state memory DNA array that can be edited irreversibly by serine integrases and integrated at defined genomic sites. While MEMOIR's design enabled 2 different states for each recording unit in the memory array, intMEMOIR enables 3 different states. Upon induction by doxycycline, the serine integrase Bxb1 can bind to the editable character array elements or barcodes, and by DNA-recombination mutate the recording element ground state (represented as '1') into either two possible states, a deletion (represented as '0') or an inversion (represented as '2') of the DNA sequence. The recording process is fully stochastic and happens irreversibly at a constant rate, as any element in the array can be edited at any moment. On mouse embryonic stem cells, Chow *et. al* showed that lineage information can be recorded irreversibly and stored in the intMEMOIR array, while also read-out using microscopy. From the recorded data, the lineage history can then be inferred (**Fig 1A**).

In the experiment, the growth of 106 cell colonies was traced, each one started from an individual cell carrying an unedited 10-character array. Recording was induced for the first 36 hours of growth (approximately 3 cell divisions) and cells were then allowed to grow with no further recording for an additional 24 hrs. At this point the arrays for each cell in the colony were read-out using single molecule fluorescent *in situ* hybridization (smFISH). For each colony, the ground truth lineage was obtained from time-lapse movies. As cells grow at different speeds and some of them die, the resulting colonies had a distribution of sizes, from 4 to 39 cells (see **Table 1**).

Simulated *in silico* datasets

To complement the challenge datasets, data from simulated recording arrays, with respectively 200 Cas9 targets in each cell for *C. elegans* and 1000 targets for *M. musculus*, were generated. Inspired by the GESTALT technique (McKenna et al., 2016a), in the simulations, every cell is represented as a vector of 200 (or 1000) characters, each character representing one Cas9 target. The simulations started with one cell, the fertilized egg, and all its targets in an unmutated ground state represented with "0" (see **Fig 1C**) had the possibility to change to either of 30 different mutational outcomes stochastically as cells divide (see **Box 1**). The initial cell then undergoes a series of cell divisions growing into a population of ~1,000 cells for *C. elegans* and about a trillion cells from which ~10,000 cells are preserved for *M. musculus* (see STAR Methods). The recording array accumulates independent and irreversible CRISPR-induced mutations with a constant probability per time unit, inherited in subsequent cell divisions (see **Box 1**).

When a Cas9-induced mutation occurs, the double strand of DNA is broken, which is eventually repaired by the cell. However, in cases where two or more relatively close double strands break before the cell repair machinery can act, the DNA between these breaks can be lost and such events are called an "inter-target deletions". To make these simulations more realistic, we included inter-target deletions affecting 5-10% of the mutation events (see STAR Methods and **Box 1**). We also introduced different probabilities for the different mutational outcomes, in agreement with experimental evidence (McKenna and Gagnon, 2019). Additionally, for the *M. musculus* simulations we implemented a 20% data acquisition dropout to reflect the fact that the data acquisition from single cells is rarely perfect (Qiu, 2020) (see **Box 1**). In summary, we introduced experimental parameters where possible in the simulation in order to approximate realistic recording assays.

Training data

As the goal of these challenges was not only to benchmark cell lineage reconstruction algorithms, but also to mobilize a larger community for evaluating new optimal tree-building methods, we provided training data for each challenge. In the *in vitro* challenge, participants were asked to reconstruct the test dataset consisting of 30 cell colonies using only the intMEMOIR array readout, as the ground truth for these lineages was not accessible to the participants. As training set, participants were given array readout data from 76 colonies along with the corresponding ground truth lineages (**Box 1**).

For the *in silico* challenges, the training data included the ground truth simulations of 100 lineage trees and their mutated array states. These trees comprised 100 cells for *C. elegans* and 1000 cells for *M. musculus* generated with the same simulation scheme as for the whole *C. elegans* and *M. musculus* trees. The rationale was to test whether training sets composed of smaller trees could still be helpful to fine-tune algorithms then used to reconstruct larger lineages. The *C. elegans* training set tree topology was generated by 100 iterations of pruning and regrafting sub-trees of 100 cells from the whole animal lineage tree (**Box 1**), to preserve some of the initial topology without giving away the origin of the tree. We indeed verified that the aggregation of the 100 trees given for training showed no direct similarity to the 1000 cells

C. elegans tree. The *M. musculus* training set was obtained using the same eSTG algorithm used for the test dataset but ran for a shorter time in order to obtain smaller trees of 1000 cells. Importantly, the *M. musculus* challenge also had an intermediate step where participants could submit solutions to a ~6000 cell tree and obtain their scoring results on a leaderboard in real time. The leaderboard encouraged participation through competition and provided a way of testing the scalability of the approaches. For scoring, the submitted lineage tree inferences for the test dataset were then compared to their corresponding ground truth using two different metrics (see **Box 2**).

Results

Best performing methods

Overall, the challenge was successful in its main goal to attract a variety of approaches and teams, as twenty-two submissions were received in total for the three challenges. **Figure 2A-C** shows the score rankings by both the RF and triplet distances. For the *in vitro* challenge, where nine teams participated, it is clear that the diverse set of approaches reached a plateau in performance for both metrics which suggests that participants successfully extracted and used all available information in the data (**Fig 2A, Fig S2** and **Fig S3A & B** fitted blue line to the medians). We found that the top three teams performed equally well even when calculating the Bayes Factor and an additional quartet metric (**Fig S2**). Interestingly, the two distance-metrics generated different rankings, showing that while correlated the two metrics are not identical. We noted that in general teams performed better on the RF distance compared to the triplet metric (**Fig 2A** and **Fig S3C**). This indicates that for trees less than 100 cells, the triplet metric is more stringent than the whole-tree partitions measured by RF.

Five teams submitted solutions for the *C. elegans* and three teams for the *M. musculus* challenge. In both challenges, the distance-based *DCLEAR* method outperformed all other participants. In general, *DCLEAR*'s performance in both challenges and under both metrics was excellent (**Fig 2B** and **2C**) and although the *M. musculus* tree was ten times larger, *DCLEAR* scored higher compared to the *C. elegans* tree.

Summary statistics for the *in vitro* challenge

Given that the *in vitro* challenge predictions consisted of 30 trees of different sizes, we were able to further analyze the results. When considering only perfectly reconstructed trees, defined by a distance value of 0, *AMbeRland** performed better as we see a larger number of perfect trees when considering triplets (28 trees across teams **Fig. 2D *top***) than when using RF (21 trees **Fig. 2D *bottom***). This discrepancy indicates that even when all triplets from a tree are correctly inferred, there might still be incorrect clades in the tree as measured by RF. We then asked whether the different teams performed better depending on the size of the tree, a main constraint for inference accuracy. Larger trees were defined as having more than 8 leaves/cells and small trees as having less or equal to 8 leaves/cells. Irrespective of the tree size, *AMbeRland** also performed better (see **Fig 2E**). To visualize that indeed tree size has an overall effect on reconstruction accuracy, we plotted the accuracy of individual trees in both metrics colored by the number of cells per tree (**Fig 2F**). Across all trees and submissions, the two metrics correlation is overall high *r*=0.77, but it becomes clear that larger trees generally have a larger triplet distance compared to RF. A total of six trees were reconstructed perfectly by at least one of the teams (**Fig S4**) and we noted that these perfect trees consisted of small trees of less than 9 cells. For these small trees, edit patterns can be slightly redundant without affecting accuracy (e.g. Tree 1 in **Fig S4**) indicating that the size of the tree is a dominant factor in reconstruction accuracy. The largest perfect tree (Tree 20, **Fig S4**) comprises 9 cells with redundant mutations in two array states across cells, despite this, the tree can still be perfectly resolved. More generally, higher redundancy in array states effectively decreases the information that can be used for lineage reconstruction and we indeed observed high levels of redundancy in several trees with an average of 65% ± 20% of cell arrays being unique (**Table S1**). However, tree reconstruction was not affected by this (**Fig S3D & E**). Considering non-perfect trees, the largest tree with the highest score was reconstructed by *AMberRland* (29 leaves/cells, 55% unique arrays RF distance = 0.44 and triplet distance = 0.40, **Fig S5**). The second largest tree with high score was reconstructed by *Cassiopeia* (23 leaves/cells, 71% unique arrays, RF = 0.48, Triplets = 0.70, **Fig S5**). In tree 29 we noted that some cells with identical array states were placed correctly in the reconstruction, this is due to the fact that *AMberLand** and *Jasper06* decided to leverage the biological restriction that lineage trees must be binary. Therefore, they imposed a binary structure even when cells had identical array states, reaching slightly higher accuracy (**Fig. S5**).

Methods summary

The best performing methods across challenges can be roughly divided into three groups: (1) distance-based methods such as the best performers *Liu*'s method, *Guan's* method and *DCLEAR* (2) a machine learning based method to predict probabilities of sister cells using a Gradient Boosting Machine *AMbeRland*, and (3) a maximum parsimony-based method *Cassiopeia-ILP* and *Cassiopeia-Greedy*. The distance-based methods reconstruct the lineage trees by first defining a distance to build a matrix between all pairs of cells as the distance between mutated characters in two cells'arrays should be proportional to the time since they split from a common ancestor. Therefore, distance matrices are commonly used in phylogenetic inference and clustering (Jones et al., 2020) or by hierarchical algorithms that represent the distance matrix as a tree such as in Neighbor-Joining (NJ)(1987). Conversely, the machine learning approach learns from the training set the importance of features/mutations to predict whether two cells are sisters. *Cassiopeia*'s maximum parsimony method reconstructed trees by minimizing the total number of steps required to explain a given configuration of the leaves.

Distance-based methods combined with hierarchical clustering overall performed well with the additional advantage of being scalable. Hamming distance is a metric used for phylogenetic analysis where the distance between sequences from two taxa (or cells in this case) is calculated as the number of different sites between the two sequences. While in the traditional Hamming distance, every mutation is assigned the same weight, in lineage recording technologies the editing rates of each array character are generally not uniform (**Fig 3A** and **Box1**), and so, mutations that occur with higher frequency are likely to arise independently in non-related cells, confounding the analysis. Conversely, some edit patterns are unlikely to happen independently and could be informative of a true inheritance event. Therefore, the uneven frequency of array edits suggests that each array element could potentially bring different information about the underlying lineage relations. To calculate the weighted Hamming distances between cells, several teams transformed the initial edited array sites of all cells in the lineage to their observed mutation frequencies and calculated the absolute difference between the arrays of two cells (**Fig 3B**). Tables S2 and S3 include a concise summary of all methods. For the *in vitro* challenge we included the type of parameters or features that different teams estimated from the data, how was the tree built from their estimations and how did they use the training dataset to estimate or learn the different features and parameters (**Table S3**). For the *in silico* challenges, given the larger scale of the trees, we also show the CPU running time as well as the code accessibility (**Table S3**).

***Liu*: Inference of internal states.**

In all three challenges Team *Liu's* method reconstructed internal nodes to represent the ancestral nodes that likely gave rise to the leave cells. For the *in vitro* challenge, the state of every internal node is inferred using the states of its children by applying the following rule for each site: the parent node gets the state of the children nodes if both children states are the same, alternatively it gets the unedited state if its two children states are different. Next, for each array element, the transition rate from state '1' to state '0' or '2' is calculated as the probability of parent node having state 1 and child node having the mutated state (**Fig 3C *top***). Finally, the pairwise distance between two cells is considered to be the probability of two cell states arising from independent events, that is, the product of the transition rate of shared states between the two cells. In a similar way for the *in silico* challenges, team *Liu* estimated the character array of the internal nodes based on the fact that a target can only mutate once (**Fig 3C *middle***). Deletions or d*ropouts* were replaced by the initial character "0". After inferring all the internal nodes, *Liu's* method derived the empirical transition probability from the ground state to the 30 possible mutated states, 'A-Z' and 'a-c' or deletion '-'. This empirical distribution was then used to calculate the probability of two cells arising from two independent events, assuming that each target was independent of the other. The log likelihood of the transition probability for shared states was considered as the cell-to-cell distance. Finally, the distance matrix was clustered using Unweighted Pair Group Method with Arithmetic Mean Algorithm (UPGMA) (**Fig 3C *bottom***). For the *M. musculus* challenge *Liu's* method added an extra step for clustering taking into consideration the 11 different types of cells.

***Guan:* weighted Hamming distance.**

For the *in vitro* challenge *Guan Lab*'s method first designed a rule-based hierarchical clustering method using weighted Hamming distances between cells **(Fig. S6A** for frequency and weight values**)**. *Guan Lab* transformed the initial edited array sites of all cells in the lineage to their observed mutation frequencies while retaining the mutation directions by mathematical signs (+/− see **Fig S6A**) and calculated the weighted distance as the absolute difference between the arrays of two cells. Finally, the lineage was reconstructed using a rule-based hierarchical clustering method (**Fig S6B**). For the *C. elegans* challenge they first replaced all gap mutations with the mutation types at both ends, since gaps even at the same sites could be the result of simultaneous mutation incidents (**Fig 3D**). The mutation weights were defined for each of the 200 characters in the *C. elegans* array as 1-$\log_{10}(P)$, where *P* is the observed probability of the mutation at that site. An iterative bifurcate clustering process was performed to combine the nearest cells based on matrix calibration, until there was only one pair of cells left and their parent cell was defined as the root of the tree (see **Fig 3D**).

***Cassiopeia*: Combinatorial optimization.**

Yosef Lab was the only team that did not opt for hierarchical clustering but instead, they used combinatorial optimization. For the *C. elegans* challenge, the team adapted the previously published *Cassiopeia-ILP* (Jones et al., 2020) an integer linear programming (ILP) which takes as input a "character matrix," summarizing the mutations seen at heritable target sites across cells (**Fig 3E *Top***). It then infers a Steiner Tree, finding the tree of minimum weight connecting all observed cell states across all possible ancestral states' histories and maximizes the parsimony over all possible trajectories that could have generated the observed barcode states which consistently finds a near-optimal solution. Importantly, the edges connecting cell states can be weighted by the number of mutations along that edge or the log-likelihood of these mutations. A derived method *Cassiopeia-Greedy* was implemented for the *M. musculus* challenge also adapted a different maximum parsimony-based strategy to infer the phylogeny from a set of observed character-states across all cells summarized in a cell's x cut-site "character-matrix" (Jones et al., 2020). To do so, the algorithm recursively applied a heuristic to split cells into two groups based on the frequency of a given state at a character and the likelihood of that state arising, taking into account mutations that occurred earlier in the tree (**Fig 3E *Bottom***). This procedure was applied until a full lineage tree was resolved.

Usage of the Ground Truth

For the *in vitro* challenge, several teams computed the calculated transition rates across the 76 trees in the training data and found striking variability across the array element identities and positions (**Fig 3A)**. It is possible to assess in several ways how much information regarding the correct lineage of a cell is contained in the transition rate of a particular mutation. For example, given a tree in the training set it is possible to assess whether cells having the same mutation in an array element are in the same subtree branch (see diagram **Fig 3F**). To obtain the percentage of correct branch positioning associated to this mutation, this process can be repeated for all trees. It can then be expanded to all ten elements in the arrays, and for the two types of mutations (1 to 0 or 1 to 2). This information was used to quantify how for a given mutation and array position there is a negative correlation between the state transition rate and how well it can establish the correct relationships between four cells in a subtree (R^2=0.58, see plot **Fig 3F**). This observation is in line with teams assigning the observed mutation frequencies to the Hamming distance weights of different array elements, but also shows that weight values can be further refined when training data is available.

Participant teams used this type of information differently as *Cassiopeia-ILP* (*Yosef Lab*) used the average across sites of the transition probabilities for each type of mutation to weight the edges of their Steiner-Tree search (**Fig 3E *top*)**. Additionally, for this team the training data also proved useful in choosing a model as they were able to compare the performance of different algorithms and select the one that performed the best (**Fig 3G)**. Team *Guan Lab* was able to use the ground truth for comparing several types of distance-based tree construction methods, including Neighbor-Joining (NJ) and UPGMA. This analysis showed that UPMGA performed similar to their rule-based hierarchical clustering whereas NJ was significantly outperformed (**Fig S6C**). Finally, *DCLEAR*(WHD) used the training set to weight the mutations for the *C.elegans* tree and *AMbeRland* used a Gradient Boosting Machine (GBM) to learn the relative importance of several features derived from the array states data and for determining the clustering thresholds for the tree reconstruction (see details below).

DCLEAR estimates k-mer replacement distances by simulation

DCLEAR (Distance based Cell LinEAge Reconstruction) implemented two best performing strategies to compute the cell distances. A weighted Hamming distance strategy (WHD) that requires a training set for optimizing each mutation weight for the *C. elegans* tree, and a *k*-mer replacement distance (KRD), that does not require training data, for the *M. musculus* tree. *DCLEAR* (KRD) first looks at mutations in the character arrays to estimate the parameters of the generative process associated with the tree to be reconstructed. With these parameters, they repetitively simulated trees with a size and mutation distribution similar to the *M. musculus* target tree (**Fig 4A**). The *k*-mer replacement distances were estimated from the simulated lineage trees and used to compute the distances between input sequences in the character arrays of internal nodes and tips. As a toy example, two cells in a simulated tree have respectively the character arrays A00A and E00C, their *1*-mer nodal distance will be the distance between A and C, their *2*-mer nodal distance will be the distance between 0A and 0C while the whole sequence nodal distance will be between A00A and E00C (see red cells in **Fig 4B**). Specifically, by examining the simulated lineage trees, *DCLEAR* (KRD) estimated the expected *1*-mer replacement distance between characters in the array (including ground state '0' and deletion state '-') in the lineage trees (**Fig 4C**) and the probability for a given nodal distance of replacing a character in a cell array (**Fig 4D and 4E**). To extend the *1*-mer replacement distance to the *k*-mer replacement distance, the posterior probability distributions of *k*-mer replacement distance were estimated by using a conditional model considering a dependance for the concurrence of mutations (**Fig 4F** and **4G**). They found that by considering the neighboring characters, the conditional model can more accurately estimate the nodal distance than an independent *1*-mer model. The cell distance can then be readily computed as the mean expected *k*-mer replacement distance (see STAR Methods). Similar to WHD, the lineage trees were reconstructed using the Minimum Evolution (FastME) or Neighbor-Joining (NJ) algorithms (Gascuel and Steel, 2006; Lefort et al., 2015). For both *DCLEAR* WHD and KRD, the deletions and dropouts were treated differently. In WHD, the weight for deletion, dropout, regular state and ground state are 0.9, 0.4, 3 and 1, respectively. In KRD, deletion and dropout are treated as two different characters.

***Amberland*, a decision tree-based method**

AMbeRland's approach relied on machine-learning to build a distance matrix between cells through the calculation of the relative importance of features derived from the states of the character arrays (**Fig 5**). In their approach for the *in vitro* challenge, they first defined four features for every pair of cells consisting of whether two cells are both unedited at a given array site (feature F1), a site has the same edits (feature F2), only one site is edited (feature F3) or if both sites have different edits (feature F4) (**Fig 5A *left***). Then, the prevalence of these four features was extracted for a group of ~500 pairs of sister cells (label 1) and ~3000 non-sister cells (label 0) using the 76 ground truth trees available in the training set. Finally, Gradient Boosting (Friedman, 2001) was applied to learn from this data the relative weights of each feature to predict whether two cells are actually sisters (see **Fig S7)**. For the *C. elegans* challenge *AMbeRland* applied a similar approach using the training set of 100 trees with 100 cells. They similarly determined weights for features selected by counting pairwise positions in two cell's arrays that were (1) not mutated, (2) had a single mutation, (3) both had different mutations, (4) both had a missing record, (5) one had a missing record and the other not mutated, etc. (**Fig 5A *right***).

In both challenges, *AMberland* applied a custom hierarchical clustering method for building the cell lineage tree from the predicted probabilities. During the tree construction, the ground truth was used to evaluate a set of decreasing thresholds corresponding to how any two individual clusters of cells were related at different levels of the lineage tree (see **Fig 5B *left***). The clustering starts at the lowest tree level, where all cell pairs are ordered according to the predicted probability that they are sister cells, from here, cells with a probability higher than the first threshold are assigned as pairs, while the rest are kept as a branch with a single cell. At each consecutive level, pairwise comparison are performed between each lower level cluster by calculating the maximum probability between any two elements of the two clusters. Pairs of clusters were ordered again according to this probability and were assumed to have the same parent node if their value was above the estimated threshold for this level. This process was repeated until one or two clusters were left. The values for the thresholds at each level were determined by performing a grid search minimizing the RF and triplet distance metrics (see results for tree 29 **Figure 5B *right*).** This procedure clearly helped obtaining better scores, particularly regarding the triplet metric (see **Fig S8** for all trees in the *in vitro* challenge and **Fig S9** for *C. elegans*).

Consensus trees

One advantage of having a set of different and diverse approaches trying to solve a common problem is that it is possible to aggregate the solutions and gather collective insight. Hence, we decided to test how a consensus tree of all teams would perform compared to individual methods (**Fig 6A&D**). For the *in vitro* challenge*, w*e constructed the consensus tree using the submissions from all teams (excluding *Bengal Tiger* because of their unusual number of low-accuracy outliers, **Fig 2A**) by applying the majority-rule algorithm (Felsenstein, 1985). Interestingly, we see that the consensus tree performs better than any individual team when considering the RF distance, but this is not the case according to the triplet distance (**Fig 6B**). To further understand this, we evaluated the agreement (or support) of each clade in a given tree across teams using the Felsenstein's Bootstrap Proportion (FBP), which has been traditionally used to assess the support of phylogenetic trees (Felsenstein, 1985). For FBP agreement, a branch must match a reference branch exactly to be accounted for in the score, so we define FBP as a strict agreement (**Fig 6A**). Alternatively, the Transfer Bootstrap Expectation (TBE) provides higher resolution estimates of branch support and can be used to assess phylogenetic similarity even when there is no strict majority consensus (Lemoine et al., 2018). The distribution of FBP and TBE support scores at different normalized depths across all 30 trees in the test dataset shows that the inference of earlier clades varies significantly across methods, whereas late splits are resolved correctly by the majority (**Fig 6A and Fig S10**). The divergence for earlier clades might explain the lower performance of the consensus tree under the triplet metric, given that for these small trees more triplets are prone to include early divisions with wrong clade relationships (see **Fig S3A&B**).

For the *in silico* challenges we also added for comparison the performance of the algorithm *FastTree2*, a fast and reliable approximately-maximum-likelihood method (Price et al., 2010) that performed better than neighbor joining or TripleMaxCut (Sevillya et al., 2016). Interestingly, we observed that in the *C. elegans* challenge, *DCLEAR* outperforms *Fastree2* by both metrics, which is not the case for the *M. musculus* challenge as *FastTree2* outperforms all methods, with *DCLEAR* as a close second (**Fig 6C**). We also see that for the *C. elegans* challenge, the consensus tree performs better than any individual team when considering the RF distance, but under the triplet distance the consensus is nevertheless equivalent to a random submission (**Fig 6C**). In the *M. musculus* challenge there were probably not enough submissions to see a "wisdom of the crowds" effect as the consensus tree does not outperform *DCLEAR*. To understand the difference between the RF and triplet distances, we evaluated the agreement of each clade in the *C. elegans* tree across teams. Overall, as in the *in vitro* challenge we observed a depth-dependent effect in the support between teams, as measured by TBE (**Fig 6D**) and the divergence for earlier clades might explain the lower triplet metric performance in the consensus tree solution but in this case probably due to the *C. elegans* tree topology having many internal nodes.

Discussion

The main goal of this DREAM challenge was to mobilize a larger community to generate new methods for cell lineage reconstruction. This goal was catalyzed through the generation of new *in silico* datasets and by the recent availability of *in vitro* datasets with an associated ground truth. This study represents the first attempt to rigorously examine the performance of various algorithms across diverse molecular tools and lineage trees. For the *in vitro* challenge a total of nine approaches were submitted for which the maximum performance plateaued (see **Fig 2A** and **Table S2**). While some trees were reconstructed perfectly, the scores were far from the theoretical maximum. We thought this could be mainly due to the high degeneracy in cell arrays where two or more cells show identical edit patterns, but further analysis showed that barcode degeneration did not affect the performance of the teams (**Fig S3E**). This problem could be in principle overcome by increasing the memory of the intMEMOIR system, as discussed by the authors (Chow et al., 2021). On the other hand, the degeneracy problem was non-existent for the *C. elegans* tree as all cells ended up with a different mutational character array and was minimal for *M. musculus* with only ~2.7% of sister cells sharing exactly the same character arrays. Indeed, the choice of the mutation rate and the diversity of mutations in the simulations has a strong effect on the accuracy of cell lineage reconstruction as low diversity of possible mutational outcomes generally gives poorer results. While too low mutation rates lead to more unedited and therefore non-informative targets, too high mutation rates lead to most targets being mutated during the early cell divisions, leaving few targets available for recording later events (Salvador-Martínez et al., 2019). Hence, we tuned our *in silico* mutation rates and array sizes in order to avoid cells having identical character arrays. As the performance of *DCLEAR* in the *in* *silico* challenges was as good or even better than the results of the *in vitro* challenge (see **Fig 2**), the limits of its performance must derive from the tree size or topology. We conclude that tree topology was the most important parameter given that *DCLEAR M. musculus* lineage reconstruction was more accurate than for the ten times smaller *C. elegans* tree. Given these great performance, we also consider the *in silico* challenges a success despite not having as many submissions, as the diversity and performance of the approaches was impressive (see **Fig 2** and **Table S3**).

The implementation of several metrics to evaluate the participants was also an original feature of the challenge as in general, lineage trees are evaluated with a single metric and no comparison between metrics is systematically performed (Salvador-Martínez et al., 2019). This aspect was essential not only to thoroughly evaluate participants (**Fig 2, S2** and **S3**) but also to better understand their solutions. One of the striking observations was the disconnection in all challenges of the performance as measured by the two metrics. Indeed, for the *in vitro* challenge *AMberRland* optimized post competition their algorithm for the triplet distance and had the overall best performance without compromising their RF performance (**Fig 3A**). Also, for larger trees, team *AMbeRland** had overall a similar performance than *Cassiopeia* relative to the triplet distance (average triplets = 0.55) but scores better in the RF metric (RF = 0.57 and 0.65 respectively, see **Fig 2E**). We see the opposite for team *philrennert* although now the difference for larger trees appears for the triplet distance (triplets = 0.57 and 0.72 respectively, see **Fig 2E**) as the RF distance is similar. Such dissociation between metrics was also observed for the majority-vote consensus solution which had the best score for RF but far from that for triplet distance (**Fig 6B**). The analysis of the overall agreement between individual solutions at different depths of the trees shows that indeed for earlier cell divisions agreement is low (**Fig 6A**). This observation provides a possible explanation for the divergence between triplet and RF distances, as in smaller trees such as the ones in the *in vitro* challenge, more triplets are prone to include early divisions with wrong clade relationships, bringing down the triplet performance. *AMberRland* was probably able to correct this by performing a grid search and changing the thresholds for hierarchical clustering at higher levels of the tree. As *AMberRland* was also the method that most consistently predicted smaller and larger trees (**Fig 2D, S4 and S5)**, this also explains why we observed that overall the triplet distance is higher than RF in larger trees as opposed to smaller trees (**Fig 2F** and **S3C**).

For the much larger trees in the *in silico* challenges the interpretation of the metrics is different as the number of triplets included in the triplet distance grows cubically with the size of the tree, while the number of partitions considered by the RF distance grows linearly. Hence, for larger trees, the triplet distance will be dominated by the higher number of triplets close to the tree leaves as the RF distance will be mostly measuring major branching events in the early cell division stages. As *DCLEAR* was consistently better in both metrics, but scored less favorably in RF distance, compared to the triplet distance, this suggests that *DCLEAR* is precisely having trouble detecting those major branching events. Indeed, both WHD and KRD in *DCLEAR* methods rely on the rare mutations to estimate the cell distances. During early cell division stages, however, the rare mutations are significantly less likely to be present in the sequences and result in difficulties for separating early branching events. Modeling the dependence between multiple non-adjacent mutations in the sequences, on top of the neighboring *k-mers*, may be necessary to more accurately evaluate the early branching events. It is also striking to see how the maximum parsimony approach of *Cassiopeia* scored much better for the triplet distance for larger trees in all challenges. Finally, the machine learning approach derived from the one applied in the *C. elegans* challenge by *AMberRland* was able to perform acceptably in the RF metric with much larger trees (see **Fig 2B**), but although the threshold optimization worked for the training set of 100 cell trees (see **Fig S9**), it did not do well with the triplet distance of the *C. elegans* tree probably due to the need to include many more thresholds given its 10 times larger size.

The final observation regarding the metrics discrepancy is related to the performances in the training and test sets of the *C.elegans* challenge, as all teams are similar regarding the RF distance but with the exception of *DCLEAR* and *Cassiopeia*, the triplets performance is worse for the test set than in the training set (see **Box 1**). Conversely, for the *M.musculus* challenge their performances in the leaderboard tree of ~6500 cells and the *M.musculus* tree of ~10,000 cells match for both metrics (**Fig 6C**). We conclude that when reconstructing a cell lineage tree, the results obtained with an algorithm for a training set of trees with a number of leaves an order of magnitude smaller than the test set are comparable, although the triplet distance is more unstable than the RF distance.

Regarding the generalization of the results obtained with the intMEMOIR technology which is difficult to compare at the molecular level to the sequence-based approaches for lineage reconstruction as it also shows differences such as the absence of accidental deletions or *dropouts*, we think that in conjunction with the results from the *in silico* approaches, the generalizable conclusions are the necessity of having well calibrated mutation rates to avoid too little mutations but also array degenerations, the utility of having a training set of smaller trees to optimize lineage reconstruction methods including distances and clustering, and allowing for a clear interpretation of the effect of the two different metrics with different tree sizes.

Overall, we think that the decisions taken while producing the datasets for the *in silico* challenges were the correct ones. We were able to pose a problem that we think is close enough to a biological situation and difficult enough so that the lessons learned and solutions generated can be implemented in other contexts. Indeed, it has been estimated that under ideal conditions of optimized mutation rates, uniform cell divisions and fully sequenced targets, 30 targets should be sufficient to reach a high level of accuracy for the lineage reconstruction of a tree of about 65,000 cells (Salvador-Martínez et al., 2019). In this situation 100 targets would theoretically yield almost perfect accuracy, far from the results obtained by the solutions submitted to both challenges.

Finally, as new DNA-editing-based molecular tools promise the reconstruction of single-cell lineages from complex model organisms, including the human cell lineage, an important question is whether the access to smaller trees and the molecular data from their cell lineages could help find solutions to be implemented for larger trees of the same origin. The *M. musculus* lineage tree being the current experimental frontier for lineage reconstruction(Bowling et al., 2020; Kalhor et al., 2018), our results show that indeed, in order to obtain an accurate full cell lineage for mouse or human, it could be possible to train algorithms on smaller trees obtained from organs (Bowling et al., 2020) or *in vitro* dividing cells and these can then be implemented for building algorithms that can then be applied to the reconstruction of much larger trees. This DREAM challenge was a first attempt to rigorously examine the performance and robustness of various algorithms under the same conditions. It took advantage of the unique opportunity to use unpublished datasets of molecular and simulated character arrays. We hope that showing that machine learning methods can indeed be successfully implemented will pave the way for other benchmarking efforts based on emerging technologies for monitoring cell lineages and the application of new algorithmic approaches, but also that the approaches described here will pave the way for the solution of the mouse and human cell lineages.

Acknowledgements

Funding: The research was funded by the Paul G. Allen Frontiers Group and Prime Awarding Agency. **Author contributions:** AG, IS, OR,YG, ZL, NY, JS, MJT, ES,MBE, PM designed research, AG,OR, IS,WG, JH, HZ, RR, MGJ, PM analyzed data, AG, OR,IS, WG, JH, HZ, RR, MGJ, PM, wrote the manuscript **Competing interests:** The authors declare no competing interests.Worm image in Figure 1 was modified from Caenorhabditis elegans hermaphrodite adult-en.svg from Wikimedia Commons by K. D. Schroeder, CC-BY-SA 3.0. The schematic cell lineage of *C. elegans* in Figure 1 and Figure 6 was generated using the cell lineage web visualization tool *CeLaVi* available at http://celavi.pro (Salvador-Martínez et al., 2020).

References

Alemany, A., Florescu, M., Baron, C.S., Peterson-Maduro, J., and Oudenaarden, A. van (2018). Whole-organism clone tracing using single-cell sequencing. Nature *556*, 108–112.

Becattini, S., Latorre, D., Mele, F., Foglierini, M., Gregorio, C.D., Cassotta, A., Fernandez, B., Kelderman, S., Schumacher, T.N., Corti, D., et al. (2015). Functional heterogeneity of human memory CD4$^+$ T cell clones primed by pathogens or vaccines. Science *347*, 400–406.

Behjati, S., Huch, M., Boxtel, R. van, Karthaus, W., Wedge, D.C., Tamuri, A.U., Martincorena, I., Petljak, M., Alexandrov, L.B., Gundem, G., et al. (2014). Genome sequencing of normal cells reveals developmental lineages and mutational processes. Nature *513*, 422–425.

Bowling, S., Sritharan, D., Osorio, F.G., Nguyen, M., Cheung, P., Rodriguez-Fraticelli, A., Patel, S., Yuan, W.-C., Fujiwara, Y., Li, B.E., et al. (2020). An Engineered CRISPR-Cas9 Mouse Line for Simultaneous Readout of Lineage Histories and Gene Expression Profiles in Single Cells. Cell *181*, 1410-1422.e27.

Chan, M.M., Smith, Z.D., Grosswendt, S., Kretzmer, H., Norman, T.M., Adamson, B., Jost, M., Quinn, J.J., Yang, D., Jones, M.G., et al. (2019). Molecular recording of mammalian embryogenesis. Nature *570*, 77–82.

Chow, K.-H.K., Budde, M.W., Granados, A.A., Cabrera, M., Yoon, S., Cho, S., Huang, T., Koulena, N., Frieda, K.L., Cai, L., et al. (2021). Imaging cell lineage with a synthetic digital recording system. *Science* 327, eabb3099.

Evrony, G.D., Lee, E., Mehta, B.K., Benjamini, Y., Johnson, R.M., Cai, X., Yang, L., Haseley, P., Lehmann, H.S., Park, P.J., et al. (2015). Cell Lineage Analysis in Human Brain Using Endogenous Retroelements. Neuron *85*, 49–59.

Felsenstein, J. (1985). CONFIDENCE LIMITS ON PHYLOGENIES: AN APPROACH USING THE BOOTSTRAP. Evolution *39*, 783–791.

Frieda, K.L., Linton, J.M., Hormoz, S., Choi, J., Chow, K.-H.K., Singer, Z.S., Budde, M.W., Elowitz, M.B., and Cai, L. (2017). Synthetic recording and in situ readout of lineage information in single cells. Nature *541*, 107–111.

Frumkin, D., Wasserstrom, A., Kaplan, S., Feige, U., and Shapiro, E. (2005). Genomic Variability within an Organism Exposes Its Cell Lineage Tree. Plos Comput Biol *1*, e50.

Garcia-Marques, J., Espinosa-Medina, I., Ku, K.-Y., Yang, C.-P., Koyama, M., Yu, H.-H., and Lee, T. (2020). A programmable sequence of reporters for lineage analysis. Nat Neurosci *23*, 1618–1628.

Gascuel, O., and Steel, M. (2006). Neighbor-Joining Revealed. Mol Biol Evol *23*, 1997–2000.

Jones, M.G., Khodaverdian, A., Quinn, J.J., Chan, M.M., Hussmann, J.A., Wang, R., Xu, C., Weissman, J.S., and Yosef, N. (2020). Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. Genome Biol *21*, 92.

Kalhor, R., Kalhor, K., Mejia, L., Leeper, K., Graveline, A., Mali, P., and Church, G.M. (2018). Developmental barcoding of whole mouse via homing CRISPR. Science *361*, eaat9804.

Kebschull, J.M., and Zador, A.M. (2018). Cellular barcoding: lineage tracing, screening and beyond. Nat Methods *15*, 871–879.

Kester, L., and Oudenaarden, A. van (2018). Single-Cell Transcriptomics Meets Lineage Tracing. Cell Stem Cell *23*, 166–179.

Kretzschmar, K., and Watt, F.M. (2012). Lineage Tracing. Cell *148*, 33–45.

Lefort, V., Desper, R., and Gascuel, O. (2015). FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program. Mol Biol Evol *32*, 2798–2800.

Lemoine, F., Entfellner, J.-B.D., Wilkinson, E., Correia, D., Felipe, M.D., Oliveira, T.D., and Gascuel, O. (2018). Renewing Felsenstein's phylogenetic bootstrap in the era of big data. Nature *556*, 452–456.

Livet, J., Weissman, T.A., Kang, H., Draft, R.W., Lu, J., Bennis, R.A., Sanes, J.R., and Lichtman, J.W. (2007). Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. Nature *450*, 56–62.

Lodato, M.A., Woodworth, M.B., Lee, S., Evrony, G.D., Mehta, B.K., Karger, A., Lee, S., Chittenden, T.W., D'Gama, A.M., Cai, X., et al. (2015). Somatic mutation in single human neurons tracks developmental and transcriptional history. Science *350*, 94–98.

McKenna, A., and Gagnon, J.A. (2019). Recording development with single cell dynamic lineage tracing. Development (Cambridge, England) *146*, dev169730.

McKenna, A., Findlay, G.M., Gagnon, J.A., Horwitz, M.S., Schier, A.F., and Shendure, J. (2016a). Whole-organism lineage tracing by combinatorial and cumulative genome editing. Science *353*, aaf7907.

McKenna, A., Findlay, G., Gagnon, J.A., Horwitz, M., Schier, A.F.F., and Shendure, J. (2016b). Whole organism lineage tracing by combinatorial and cumulative genome editing.

Perli, S.D., Cui, C.H., and Lu, T.K. (2016). Continuous genetic recording with self-targeting CRISPR-Cas in human cells. Science *353*, aag0511.

Price, M.N., Dehal, P.S., and Arkin, A.P. (2010). FastTree 2--approximately maximum-likelihood trees for large alignments. Plos One *5*, e9490.

Qiu, P. (2020). Embracing the dropouts in single-cell RNA-seq analysis. Nat Commun *11*, 1169.

Raj, B., Wagner, D.E., McKenna, A., Pandey, S., Klein, A.M., Shendure, J., Gagnon, J.A., and Schier, A.F. (2018). Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain. Nat Biotechnol *36*, 442–450.

Robinson, D.F., and Foulds, L.R. (1981). Comparison of phylogenetic trees. Math Biosci *53*, 131–147.

Saez-Rodriguez, J., Costello, J.C., Friend, S.H., Kellen, M.R., Mangravite, L., Meyer, P., Norman, T., and Stolovitzky, G. (2016). Crowdsourcing biomedical research: leveraging communities as innovation engines. Nat Rev Genet *17*, 470–486.

Salcedo, A., Tarabichi, M., Espiritu, S.M.G., Deshwar, A.G., David, M., Wilson, N.M., Dentro, S., Wintersinger, J.A., Liu, L.Y., Ko, M., et al. (2020). A community effort to create standards for evaluating tumor subclonal reconstruction. Nat Biotechnol *38*, 97–107.

Salvador-Martínez, I., Grillo, M., Averof, M., and Telford, M.J. (2019). Is it possible to reconstruct an accurate cell lineage using CRISPR recorders? Elife *8*, e40292.

Salvador-Martínez, I., Grillo, M., Averof, M., and Telford, M.J. (2020). CeLaVi: An Interactive Cell Lineage Visualisation Tool. Biorxiv 2020.12.14.422765.

Sevillya, G., Frenkel, Z., and Snir, S. (2016). Triplet MaxCut: a new toolkit for rooted supertree. Methods Ecol Evol *7*, 1359–1365.

Spanjaard, B., Hu, B., Mitic, N., and Junker, J.P. (2017). Massively parallel single cell lineage tracing using CRISPR/Cas9 induced genetic scars. BioRxiv 205971.

Spanjaard, B., Hu, B., Mitic, N., Olivares-Chauvet, P., Janjuha, S., Ninov, N., and Junker, J.P. (2018). Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars. Nat Biotechnol *36*, 469–473.

Spiro, A., and Shapiro, E. (2016). eSTGt: a programming and simulation environment for population dynamics. Bmc Bioinformatics *17*, 187.

Sugino, K., Garcia-Marques, J., Espinosa-Medina, I., and Lee, T. (2019). Theoretical modeling on CRISPR-coded cell lineages: efficient encoding and optimal reconstruction. Biorxiv 538488.

Sulston, J.E., and Horvitz, H.R. (1977). Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. Dev Biol *56*, 110–156.

Wagner, D.E., and Klein, A.M. (2020). Lineage tracing meets single-cell omics: opportunities and challenges. Nat Rev Genet *21*, 410–427.

Weissman, T.A., and Pan, Y.A. (2015). Brainbow: New Resources and Emerging Biological Applications for Multicolor Genetic Labeling and Analysis. Genetics *199*, 293–306.

(1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol.

Figure Legends

Box 1 Training set

One of the main goals of this challenge was to provide participants with a training set composed of several trees, their cell's character arrays and the gold standard tree solution. This allowed participants to train or optimize their methods.

A. In the *in vitro* experiments to obtain mouse stem cell lineages, mutations were induced for the first 36 hrs of growth (approximately 3 cell divisions) and cells were then allowed to grow with no further changes in the recording arrays for an additional 24 hrs. For all these cells the final values (unmodified encoded as 1, inverted encoded as 2 or deleted encoded as 0) of the 10 character arrays were obtained by smFISH, while cell divisions were tracked by video-microscopy (see **Table S1**). Two partitions were created from the original unpublished dataset containing 106 lineages, which represent sufficient experimental data to extract a training set: the training partition composed of 76 trees was provided for the teams along with the corresponding ground truth lineages, for the test partition composed of 30 trees only the cells character arrays were provided without ground truth. The partitions were defined to have similar tree size distribution, given that the lineages were composed of a different number of cells depending on the cell division and survival rates, shown in *middle* histogram panel. Also, a similar median RF score distribution between the two data sets when using a maximum-likelihood method described in *Chow et at* was used as partition criteria, see *bottom* panel. **B.** For the *in silico* challenges, both character arrays for the training and test sets were simulated in a similar way. The type of Cas9-induced mutations consisted of 32 characters 'A' to 'Z' and 'a' to 'e' and character deletion '-'. The characters represent DNA targets for Cas9 but no specific relationship with actual DNA sequences was established. The starting character was '0' and the probability of mutating to one of the 30 characters or of being deleted (insertions were not considered) followed in alphabetical order the Gamma probability distribution used to sample the mutations, shown in blue, and in red a fit on the histogram of the actual results. Mutations are irreversible, once a target is mutated, it can no longer change, either to revert to the unmutated state or to transit to a new state. **C.** Inter-target deletions were simulated for both *in silico* challenges where *C. elegans* arrays were composed of one hundred characters and *M. musculus* of one thousand characters. When a Cas9-induced mutation occurs, the double strand of DNA is broken, which is eventually repaired by the cell. However, in cases where 2 or more relatively close double strands break before the cell repair machinery can act, the DNA between these breaks can be lost this is known as an "inter-target deletion". We implemented these so that when two mutations occur in close targets (less than 20 targets apart in the recording array) within a short interval of time during a given cell division, all the targets between them are removed. In these simulations, 5-10% of targets are missing due to inter-target deletions. **D.** Acquisition dropout distributions were implemented only for the *M. musculus* challenge. In order to capture the variability of the signal quality in both the individual samples and the different sites we modeled the 'sequencing dropout' of single cell samples by assigning distinct coverage factors for each sample and for each locus. The density of cell coverage factors P = (*pi*: i = 1 to M) is the probability of obtaining a signal in each sample or and the density of site coverage factors Q = (*qj*: j = 1 to N) as the probability of obtaining a signal in each locus. The probability of obtaining a signal in sample i and locus j thus equals *pi.qj.r.* Those are multiplied to get the individual coverage factor of a specific site in a specific cell, finally deriving the acquisition dropout status as a factor of a global coverage parameter *r*. **E.** We provided 100 training cell lineage trees of 100 cells for *C. elegans* and of ~1000 cells for *M. musculus.* As the *C. elegans* tree has been experimentally solved, its topology was used to generate the training set. The *M. musculus* tree being completely synthetically generated, the training set was obtained by simply running shorter simulations to obtain ~1000 cells trees instead of the ~10,000 cells tree for the test set. **F.** *Top*. We extracted the *C. elegans* training set from its tree topology by cutting and pasting subsets of tree branches. We followed the indicated schematic of cutting and pruning 100 times subsets of the whole tree. Note only one prune and regraft event is shown in red in the diagram. From the obtained topology, the mutation arrays were generated from the Gamma distribution and then 100 cells were sampled. This process was repeated 100 times to obtain a full training set. *Bottom* The boxplots show the performance of each submitted method for inferring the lineage trees from 100 training lineages used in the *C. elegans in vitro* challenge. The similarity between the inferred trees and the ground truth trees was measured by Robinson-Foulds distance *left* and Triplet distance *right*. Red stars indicate the score for the *C. elegans* 1000 cell tree. The values for the *M. musculus* training set, were not established due to excessive computational time required.

Box 2 Scoring approach

We applied two widely used metrics for tree comparison: the Robinson-Foulds distance and the triplets distance. While both metrics are applied to assess tree similarities there is no clear agreement as to which one is more relevant for lineage trees. We decided to use both metrics as a way of evaluating their correlation and the insight they provide about the lineage relationships. The Robinson-Foulds distance is commonly defined as the number of partitions shared by a pair of trees across all possible partitions. A partition refers to any cut in the internal branches of a tree that would generate two sub-trees containing complementary leaves. Since the ground-truth and the inferred lineage contain in total the same set of leaves, we can define a shared partition if there is a way to cut both the inferred and ground-truth trees such that the resulting sub-trees share the same sets of leaves. We obtain the RF distance by normalizing to the maximum possible distance of 1, when there are no shared partitions by the trees (Robinson and Foulds, 1981). On the other hand, the triplet distance enumerates all possible combinations of three leaves and their corresponding lineage relationship in both the ground truth and the inferred trees. One then counts the number of shared triplets and normalizes by the total possible number of triplets to obtain the triplet distance. For both metrics, a distance value of 0 means that the ground truth and inference trees are identical under the specific criteria while a distance value of 1 means that the inference is comparable to a random guess on the tree structure. Overall, the Robinson-Foulds metric detects main branching events, while the triplet metric is a better measure of local branching events.

We here present an illustrative example with left the ground truth and *right* the predicted tree. In this case, the tree has three possible partitions *top right* and ten possible triplets *bottom left*. Since 1 out of three partitions was incorrect the RF distance is 1/3 or 0.66. Similarly, 4 out of 10 triplets were incorrect for a triplet distance of 4/10 or 0.4. Higher distance implies more differences between the ground truth and the inference and therefore a lower score. As observed in the results of this challenge, the relationship between the two metrics will depend on the tree topology but also on the tree size. Indeed the number of triplets will size as the cube of the number of nodes, while the RF partitions will scale linearly with the number of nodes.

Figure 1. Three challenges for lineage reconstruction from experimental and in silico generated character arrays. A. Challenge consisting of reconstructing *in vitro* growing cell lineages. The lineage tracing intMEMOIR system consists of a character array of editable DNA elements –or barcodes- and the integrase enzyme Bxb1. A mouse stem cell line was engineered with both components. A recording event happens when the integrase stochastically edits one of the 10 elements in the array, resulting in two possible outcomes, deletion and inversion (blue and red squares). As cells divide, each individual daughter cell acquires unique edit patterns (right panel). Finally, *in situ* readouts by smFISH enables the extraction of recorded data for individual cells. Since the whole experiment is done under a microscope, a ground truth lineage tree is also generated which we use as our ground truth. **B.** Diagram showing the simulations performed to generate the character arrays for the two *in silico* datasets, an initial cell with N multiple targets (200 or 1000 for the *C. elegans* or *M. musculus* challenge respectively) accumulates one of the 30 independent mutations with a given probability, which are inherited in subsequent cell divisions. The pattern of mutations accumulated in each cell is used to infer the lineage tree. **C.** *In silico* challenge consisting of reconstructing the ~1000 cells *C. elegans* cell lineage from the simulated cell character arrays. For visualization purposes the ground truth cell lineage shows only the first 9 cell divisions. **D.** Challenge consisting of reconstructing ~10,000 cells from a simulated *M. musculus* cell lineage developmental tree generated using Stochastic Tree Grammar (STG). The tree simulation describes the early stages of mouse development up to the three germ layers (Mesoderm, Ectoderm and Endoderm are highlighted with colors in the equations and resulting tree), those in turn continue to differentiate to the final populations of about 10^{12} cells and 11 cell types simulated in the challenge. **E.** Displayed is a simulation example of the ground truth tree for a subset of cells from the Mesoderm and Ectoderm, highlighted with the respective colors, throughout 1 year of development. The edges width and color reflect the hypergeometric score of its descending leaves.

Figure 2. **Analysis of challenge results. A.** Average performance across 30 lineages of all teams by both triplets and RF metrics for the *in vitro* challenge. **B.** Average bootstrapped performance of all teams by both triplets and RF metrics for the *C. elegans in silico* challenge. **C.** Average bootstrapped performance of all teams by both triplets and RF metrics for the *in silico M. musculus* challenge **D.** Number of perfectly reconstructed lineages for each team in the *in vitro* challenge. **E.** We partitioned the *in vitro* challenge test data into large (more than 8 cells) and small (less or equal to 8 cells) trees, to assess performance by tree size. **F.** The scores for the two metrics of all 30 trees for all 9 teams for the *in vitro* challenge are plotted against each other and color coded depending on the size of the tree. Deep blue dots, small trees #cells<10, gray blue dots, trees with 10<cells<20, light blue dots, trees with #cells>20. Scores show a general correlation r=0.77 between the two metrics, but also significant dispersion especially for larger trees.

Figure 3. Different approaches for solving lineage trees and using the training data. **A.** In the *in vitro* challenge, the transition rates from the unedited state (*1)* to either of the two edited states (*0, 2)* can be learned directly from the training data, the probabilities for all possible transitions at each of the ten array positions are shown as extracted from the training set. **B.** The schematic shows that when computing the sequence distances, instead of assigning equal weight to different character replacement as in Hamming distance, the weighted Hamming distance assigns different weights to different character replacements. **C.** Description of *Liu lab*'s method in all 3 challenges. First, for the *in vitro* challenge, the transition probability is calculated by counting the frequency of every state transition from parent node to child node. For the *in silico* challenges the transition probability for all character arrays is extracted. Next, the pairwise cell distance is defined as the likelihood of two cells' states arising from two independent events. Finally, the cell lineage is reconstructed from the distance matrix using the UPGMA method. **D.** This schematic shows the *Guan Lab*'s method used to reconstruct the *C. elegans* tree. First, all gap mutations are remarked based on mutation types at both ends, since gaps, even at the same sites, could be the results of different mutation incidents from simultaneous mutations at both ends. Then the mutation weights are generated for each mutation state at each of the 200 sites in the array and are given by 1-log10(p), where p is the observed probability of the mutation on that site. The weights define how important characters should be considered when comparing the mutation states between cells. Then bifurcate clustering of nearest cells was carried out based on matrix calibration. In the training set, the characters of all cells at all sites will be presented as *n* 200 by 100 matrices, where *n*=30 is the number of array characters (0, A, B, ...). The inner product of the matrices, which is *n* 100 by 100 matrices, reveals the relationship between the 100 cells in each tree of the training set themselves according to the 200 states, and the sum of *n* product matrices gives the overall pairwise similarity relationship of the 100 cells, where we can extract the most similar cell pair by the maximum value in that matrix (denoted as dark red, and the indices of the cells are denoted as *i* and *j*). Then a parent cell, generated based on the shared mutations of the two cells, replaces the two cells and is sent back to next iterations of bifurcate clustering, until only one pair of cells is left and their parent cell will become the tree root. **E.** *Top.* For the *in vitro* challenge Cassiopeia-*ILP* (*Yosef Lab*) takes as input a "character matrix," summarizing the mutations seen at heritable target sites across cells and infers a Steiner Tree, finding the tree of minimum weight connecting all observed cell states across all possible evolutionary histories using integer linear programming (ILP). Importantly, the edges connecting cell states can be weighted by the number of mutations along that edge or the log-likelihood of these mutations. *Bottom.* For both *in silico* challenges *Cassiopeia-Greedy* infers a phylogeny from the observed character-states across all cells, which can be summarized in a cell's x cut-site "character-matrix". To do so, the algorithm recursively applies a heuristic to split cells into two groups based on the frequency of a given state at a character, *n(i, s)*, and the likelihood of that state arising, *p(s)*. This procedure is applied until a full phylogeny is resolved. **F.** Using the 76 trees in the training set of the *in vitro* challenge to compare the relationships between cells that share a particular state, *Liu lab* quantified how rarer states are more predictive of the true

relationship between pairs of cells. As observed in the plot, these relative rates can vary by both identity and for each of the ten positions in the target array. **G.** *Cassiopeia-ILP* (*Yosef Lab*) is able to incorporate learned state priors by weighting evolutionary transitions by their log-likelihoods and find a Weighted Parsimony solution. Performance on the training data can inform whether Weighted or Unweighted Parsimony is better suited.

Figure 4. ***DCLEAR*** **Learning k-mer replacement distances by simulation. A.** The input sequences were first used to estimate the summary statistics such as mutation rate (μ), outcome probability of each character, number of targets and number of tips. These estimated parameters, combined with the pre-defined parameters such as cell divisions, were used to simulate multiple lineage trees from the root node. The k-mer nodal distances were estimated from these simulated lineage trees and then used to compute the distances between input sequences. **B.** The schematic shows a simulated lineage tree with one root, two internal nodes and three tips. The nodal distance is defined as the distance between any two nodes on the lineage tree. The expected nodal distance can be estimated from the replacement of individual characters (e.g. between A and C), the replacement of k-mers (e.g. between 0A and 0C), or sequences (e.g. between A000A and E00C). **C.** The heatmap shows the expected nodal distance of the replacement of the most frequent individual characters. **D.** The heatmap shows the probability of replacement of the most frequent individual characters at a nodal distance of 15. **E.** The histogram shows the posterior distribution of nodal distance of two sequences when having the same characters A or C at any specific position. **F-G.** The histograms show the observed distribution (red bars) and estimated posterior distribution of nodal distance of two sequences **F** with the replacement of C- by CC, or **G** with BBBB at the same position. The posterior distributions were estimated by using an independent model (blue bars) and a conditional model (green bars). In both cases, the posterior distribution estimated by the conditional model is more consistent with the observed distribution. **H.** The simulated trees were used to compare the performance of lineage reconstruction by using Hamming distance and k-mer replacement distances with different k's. We simulated 1,000 lineage trees with cell division of 16, mutation probability of 0.1, 200 targets and 200 tips. The outcome probability was sampled from a Gamma distribution with shape of 0.1 and rate of 2. For both k-mer replacement distances and Hamming distance, we used a balanced minimum evolution (ME) algorithm with tree rearrangement (nearest neighbor interchange, subtree pruning and regrafting, and tree bisection and reconnection) to infer the tree topology. The similarity between the inferred tree and the simulated tree was measured by the Robinson-Foulds (RF) distance.

Figure 5. ***AMbeRland*** **A decision tree based approach for reconstruct cell lineages**
A. After selecting manually different model features for *left* the *in vitro* challenge (F1 to F4) and *right* the *C. elegans* challenge, *AMbeRland* learns the features importance represented by histograms of the weights, for predicting phylogenetic relationships directly from the training data using a Gradient Boosting Machine (GBM) *middle*. These learned weights are then used to predict the probability of sister-cell relationships on the hold out test data creating a probability matrix used for hierarchical reconstruction *bottom*. **B.** *Left* Trees are reconstructed from probability matrices by performing a grid search to obtain the clustering thresholds at each tree level while maximizing the RF and triplets metrics. *Right* Example of differences when establishing thresholds for Tree 29, the largest correctly reconstructed tree in the *in vitro* challenge. See also detailed examples in **Fig S7 & S8**.

Figure 6. Consensus methods and agreement in tree reconstruction. A. Depth-dependent agreement between reconstructed trees calculated by Felsenstein Bootstrap Proportion and Transfer Bootstrap Expectation. Both metrics assess the degree of agreement that different trees have on specific splits (or cell divisions). High agreement indicates that most teams resolved splits correctly at that depth. The distribution is computed across all 30 trees in the *in vitro* test sets. **B.** We computed the consensus trees by majority rule using the *consensus* function from the *R package* **ape** v5.3. The consensus performance in the *in vitro* challenge is higher than any individual team by RF distance but not by triplets (red dotted line indicates the best performed by each metric). **C.** Scores summarizing all participating methods for the *in silico* challenges, including the PHYLIP consensus and for reference *FastTree2*. **D.** Annotated subtree of *C. elegans* challenge, edges are marked with tables listing the agreement of each of the 5 individual submissions and the consensus in Transfer Bootstrap Distance where 1 is high agreement. Colors refer to the table in **C**.

Supplementary figures and tables

Figure S1. *Mus musculus in silico* challenge A. Simulation of the Mouse lineage, "token" cells whose lineage are stochastically chosen to be followed as the lineage tree is formed, are shown in blue, in white are represented cells whose lineage is not followed. At the end of the simulation for the mouse lineage information for about 10,000 blue cells is stored, but it is estimated that the size of the tree is about 10^{12} or a trillion cells. **B.** Visualization of the 10,000 cell Mouse tree with 11 types of cells encoded by different colors.

Figure S2. *In vitro* challenge rankings for all teams according to multiple metrics. The ranks for each team were evaluated by calculating the ranksum values (left boxplots) for the Robison-Foulds (middle boxplots) and the triplet metric (right boxplots) sampled 1000 times with replacement from the scores for the 30 individual trees. The 9 teams were ordered by average ranksum and the Bayes Factor (BF) was calculated, yellow boxes show teams that are considered to be tied as they have a 1/3<BF<3 and a BF>3 against all the other teams in grey. Implementation of a third metric calculating quartets could not differentiate the top 3 teams: Yosef Lab (*Cassiopeia*) 0.4200, Guan Lab 0.4232, Jasper06 0.4243.

Figure S3. *In vitro* challenge results with Robinson-Foulds and triplets metrics. The participant teams'distribution of scores across 30 reconstructed lineage trees is shown for **A.** triplets metric **B.** Robinson-Foulds metric **C.** Histogram showing the difference between the Robinson-Foulds and triplets metrics for all 30 trees across all teams. Median of zero indicates that overall the metrics agree but dispersion suggests a small bias for higher distance values in triplets. **D.** The histogram of scores of all 30 trees for all 9 teams are for *left* Robinson-Foulds and *right* triplets metrics, color coded depending on the percentage of unique barcode arrays in the tree. Deep blue dots trees with 25-50% unique arrays, gray blue dots trees with 50-75% unique arrays, light blue dots, trees with 75-100% unique arrays. **E.** Comparison of team performance depending on whether cells with degenerate barcodes are merged (gold boxes) or not (blue boxes). *Left* Boxplots represent the triplet distances, *Right* RF distances, of trees where for both predictions and ground truth, cells with the same barcodes were merged into a single

leaf. The procedure followed for each tree a 100x bootstrap choosing each time a different cell with the same barcode as distances were recalculated for each fold.

Figure S4. *In vitro* challenge list of trees that were reconstructed perfectly by at least one team. Ground truth lineages are shown along with the array state for each cell.

Figure S5. *In vitro* challenge largest trees with high reconstruction scores. Two examples of large trees with 29 and 23 cells respectively and their RF and Triplets distance. These large but accurate trees were reconstructed by **A**) *AMberLand* and **B**) *Yosef Lab* (*Cassiopeia*).

Figure S6. *Guan Lab* approach for *in vitro* challenge A. Probability of mutations for the array sites and their corresponding weights for the Hamming distance. When calculating the weights for the Hamming distance, the mutation direction preference is set as reciprocal of the mutation frequency so that the rarer the mutation type, the more weight it is given to the distance between cells. **B.** A rule-based Hierarchical clustering approach was used to generate the trees. The cells character arrays final states were transformed by weights according to the observed probability of mutations, and the transformed states were used to calculate the distance between cells. The hierarchical clustering was done using a rule-based method to reconstruct parent cells, based on the fact that the editions from initial states (1) to edited states (0 and 2) are irreversible. **C.** Comparison of different clustering methods for the distance matrices including Rule-based hierarchical clustering, UPGMA and Neighbor Joining. The performance is shown for both triplets and RF distances. The Distribution across the 30 lineages in the test set and the average of the two tree measurements is shown by the violin plots. The rule-based hierarchical clustering method and UPGMA have similar performance on reconstructing cell lineage trees.

Figure S7. Representation of the decision tree and weights obtained by *Amberland* using GBM for the training set in the *in vitro* challenge. For each decision tree leaf are indicated: on top the feature's weight, the number of cells *n* and the percentage of the training set cells they represent, and in bold is the criteria of the feature used for selecting the next leaf *i.e* number of times the feature is present when comparing the 2 cells character arrays. Features in this case are:

F1-both not mutated, F2-both same mutation F3-one mutation F4-different mutations. This figure was made using the *R package* “rattle”.

Figure S8. Reconstructing trees by clustering probability matrices as implemented by AMbeRland for the training set of the *in vitro* challenge. Seventy six trees of different number of cells were used to optimize the tree reconstruction thresholds from the probability matrix of cells being sisters obtained from training a GBM algorithm **A.** Performance of the algorithm for four sets of thresholds: **set_A**=(0,0,0,0,0) results in mean RF=0.512 and triplets=0.389; **set_B**=(0.5,0,0,0,0) results in mean RF=0.519 and triplets=0.380; **set_C**=(0.8,0.4,0.2,0.1,0.05) results in mean RF=0.512 and triplets=0.433; and **set_D**=(0.3,0.1,0.05,0.01,0.005) results in mean RF=0.502 and triplets=0.375. The numbers shown in the scatter plots represent the tree ID and the color represents the number of cells in the tree. Threshold **set_D** was used to reconstruct the test dataset for submission. **B.** A perfectly reconstructed tree with 3 thresholds (tree ID 70 from the training set, RF=0 and triplets=0) has 7 pairs joined into clusters at level 1, 4 pairs joined at level 2 and 2 pairs joined at level 3. **C.** Probability matrices for tree 70 are plotted for each level. From here it can be seen that cells 7 and 8 have the highest probability so they are first joined into cluster C1, the next pair with highest probability comprises cells 12 and 13 which joined into cluster C2 and so on. Once all pairs are defined, the algorithm moves to Level 2, where clusters C2 and C3 have the highest pairwise probability (cells on these two clusters can be seen on top right corner of level 1 probability matrix) so they are joined into a new cluster C1. The algorithm proceeds until all cells are joined into a single lineage.

Figure S9. Clustering of cells into trees performed by *AMbeRland* for the training set in the *C. elegans in silico* challenge. One hundred trees of a hundred cells each were used to optimize the tree reconstruction thresholds from the probability matrix of cells being sisters obtained from training a GBM algorithm **A.** Comparing performance of the algorithm for two sets of thresholds: **set_A**={0} gives mean RF distance=0.78 and triplets=0.59; **set_B**=(0.07, 0.04, 0.01, 0.05, 0, 0, 0,0) gives mean RF distance=0.71 and triplets=0.49. Threshold **set_B** was used to reconstruct the test sample. **B.** Ground truth and reconstructed tree for training sample 100, with RF distance = 0.48 and triplets=0.44. **C.** Probability matrices for training sample 100 are plotted

for each level. Clusters identified letters C. by Four clusters for level 7 (C1-C4) are indicated on the reconstructed tree in **B**.

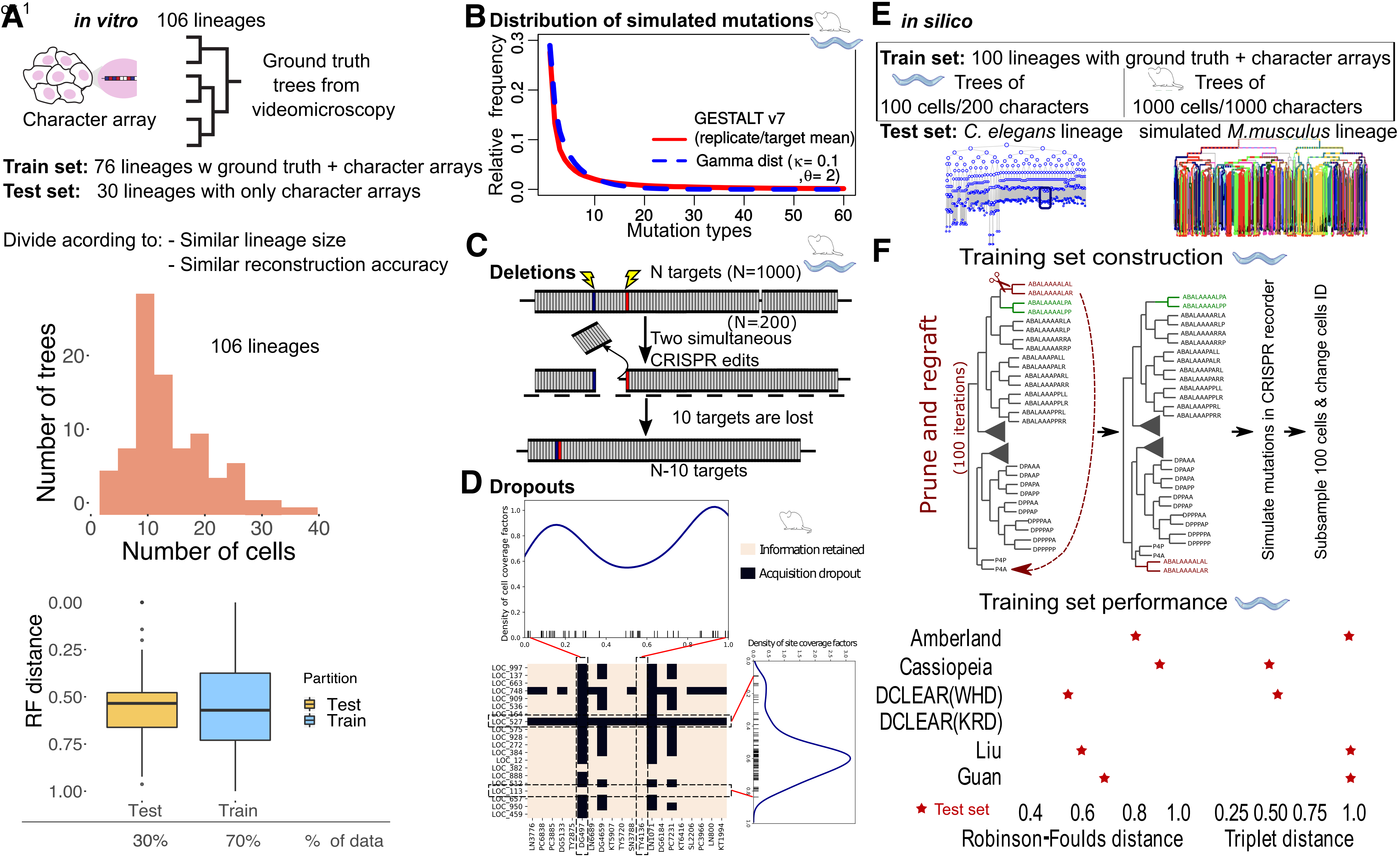
Figure S10. Agreement distribution across all reconstructed trees at different normalized tree depths for the *in vitro* challenge. A depth of 0 represents the root of the tree whereas a depth of 1 corresponds to the leaves and therefore the depth of cell divisions within the lineage fall between [0,1]. *Top* For a given ground truth lineage, The Felsenstein Bootstrap Support is calculated across all reconstructed trees submitted by the teams corresponding to that lineage. We obtain a distribution by computing the FBS score for all 30 ground truth lineages. *Bottom* The Transfer Bootstrap Expectation is calculated in an analogous way.

Table S1. Training and test datasets for the *in vitro* challenge.

Table S2. Comparing machine learning approaches for reconstruction of the *in vitro* cell lineage trees.

Table S3. Comparing machine learning approaches for reconstruction of the *in silico* large cell lineage trees (for comparison all methods were implemented in a two Intel(R) Xeon(R) CPUs @ 2.20GHz).

Box 1


A
in vitro
106 lineages
Character array
Ground truth trees from videomicroscopy
Train set: 76 lineages w ground truth + character arrays
Test set: 30 lineages with only character arrays
Divide acording to: - Similar lineage size
- Similar reconstruction accuracy
106 lineages
Number of trees
0
10
20
0
10
20
30
40
Number of cells
RF distance
0.00
0.25
0.50
0.75
1.00
Test
Train
Partition
Test
Train
30%
70%
% of data
B
Distribution of simulated mutations
Relative frequency
0.0
0.1
0.2
0.3
GESTALT v7 (replicate/target mean)
Gamma dist (κ= 0.1 ,θ= 2)
10
20
30
40
50
60
Mutation types
C
Deletions
N targets (N=1000)
(N=200)
Two simultaneous CRISPR edits
10 targets are lost
N-10 targets
D
Dropouts
Density of cell coverage factors
0.0
0.2
0.4
0.6
0.8
1.0
Information retained
Acquisition dropout
Density of site coverage factors
LOC_997
LOC_137
LOC_663
LOC_748
LOC_909
LOC_536
LOC_164
LOC_527
LOC_575
LOC_928
LOC_272
LOC_384
LOC_12
LOC_382
LOC_888
LOC_512
LOC_113
LOC_657
LOC_950
LOC_459
LN3776
PC6838
PC3885
DG5133
TY2875
DG497
LN6689
DG4659
KT5907
TY5720
SN3788
TY4136
LN1071
DG6184
PC7231
KT6416
SL2206
PC3966
LN800
KT1994
E
in silico
Train set: 100 lineages with ground truth + character arrays
Trees of 100 cells/200 characters
Trees of 1000 cells/1000 characters
Test set: C. elegans lineage
simulated M.musculus lineage
F
Training set construction
Prune and regraft
(100 iterations)
ABALAAAALAL
ABALAAAALAR
ABALAAAALPA
ABALAAAALPP
ABALAAAARLA
ABALAAAARLP
ABALAAAARRA
ABALAAAARRP
ABALAAAPALL
ABALAAAPALR
ABALAAAPARL
ABALAAAPARR
ABALAAAPPLL
ABALAAAPPLR
ABALAAAPPRL
ABALAAAPPRR
DPAAA
DPAAP
DPAPA
DPAPP
DPPAA
DPPAP
DPPPAA
DPPPAP
DPPPPA
DPPPPP
P4P
P4A
Simulate mutations in CRISPR recorder
Subsample 100 cells & change cells ID
Training set performance
Amberland
Cassiopeia
DCLEAR(WHD)
DCLEAR(KRD)
Liu
Guan
Test set
0.4
0.6
0.8
1.0
Robinson-Foulds distance
0.25
0.50
0.75
1.0
Triplet distance

Box 2

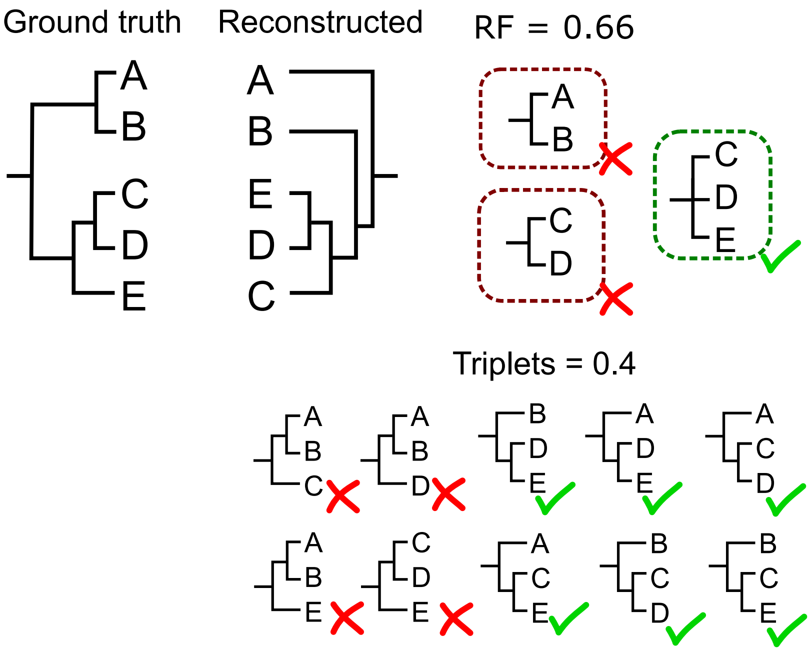

Ground truth
A
B
C
D
E
Reconstructed
A
B
E
D
C
RF = 0.66
A
B
C
D
C
D
E
Triplets = 0.4
A B C
A B D
B D E
A D E
A C D
A B E
C D E
A C E
B C D
B C E

Figure 1

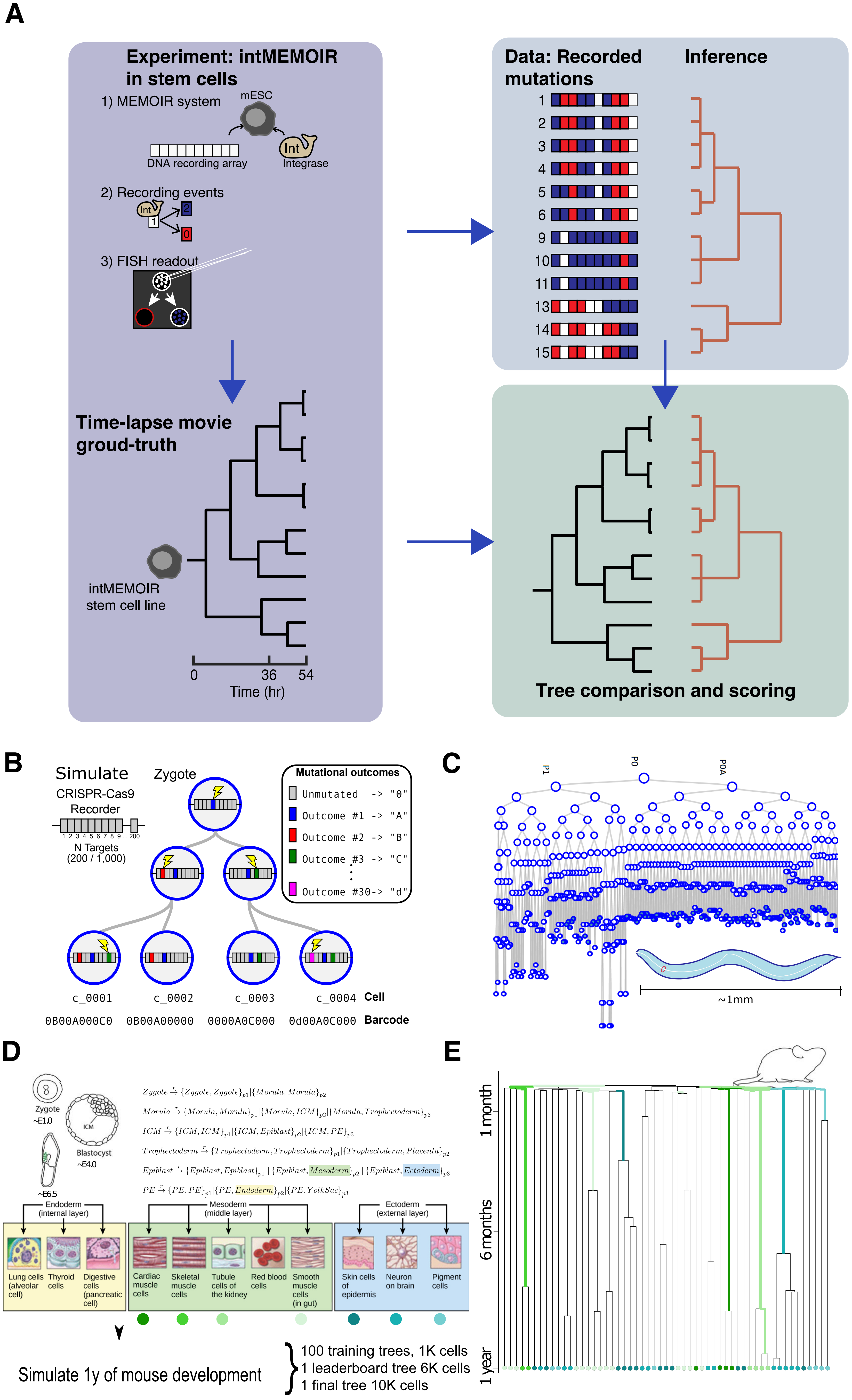

A
Experiment: intMEMOIR in stem cells
1) MEMOIR system
mESC
DNA recording array
Int
Integrase
2) Recording events
Int
1
2
0
3) FISH readout
Time-lapse movie groud-truth
intMEMOIR stem cell line
0
36
54
Time (hr)
Data: Recorded mutations
Inference
1
2
3
4
5
6
9
10
11
13
14
15
Tree comparison and scoring
B
Simulate
CRISPR-Cas9 Recorder
1 2 3 4 5 6 7 8 9 ... 200
N Targets (200 / 1,000)
Zygote
Mutational outcomes
Unmutated -> "0"
Outcome #1 -> "A"
Outcome #2 -> "B"
Outcome #3 -> "C"
Outcome #30-> "d"
c_0001
c_0002
c_0003
c_0004
Cell
0B00A000C0
0B00A00000
0000A0C000
0d00A0C000
Barcode
C
P1
P0
P0A
c
~1mm
D
Zygote ~E1.0
ICM
Blastocyst ~E4.0
~E6.5
$Zygote \xrightarrow{r} \{Zygote, Zygote\}_{p1}|\{Morula, Morula\}_{p2}$
$Morula \xrightarrow{r} \{Morula, Morula\}_{p1}|\{Morula, ICM\}_{p2}|\{Morula, Trophectoderm\}_{p3}$
$ICM \xrightarrow{r} \{ICM, ICM\}_{p1}|\{ICM, Epiblast\}_{p2}|\{ICM, PE\}_{p3}$
$Trophectoderm \xrightarrow{r} \{Trophectoderm, Trophectoderm\}_{p1}|\{Trophectoderm, Placenta\}_{p2}$
$Epiblast \xrightarrow{r} \{Epiblast, Epiblast\}_{p1} \mid \{Epiblast, Mesoderm\}_{p2} \mid \{Epiblast, Ectoderm\}_{p3}$
$PE \xrightarrow{r} \{PE, PE\}_{p1}|\{PE, Endoderm\}_{p2}|\{PE, YolkSac\}_{p3}$
Endoderm (internal layer)
Mesoderm (middle layer)
Ectoderm (external layer)
Lung cells (alveolar cell)
Thyroid cells
Digestive cells (pancreatic cell)
Cardiac muscle cells
Skeletal muscle cells
Tubule cells of the kidney
Red blood cells
Smooth muscle cells (in gut)
Skin cells of epidermis
Neuron on brain
Pigment cells
Simulate 1y of mouse development
100 training trees, 1K cells
1 leaderboard tree 6K cells
1 final tree 10K cells
E
1 month
6 months
1 year

Figure 2

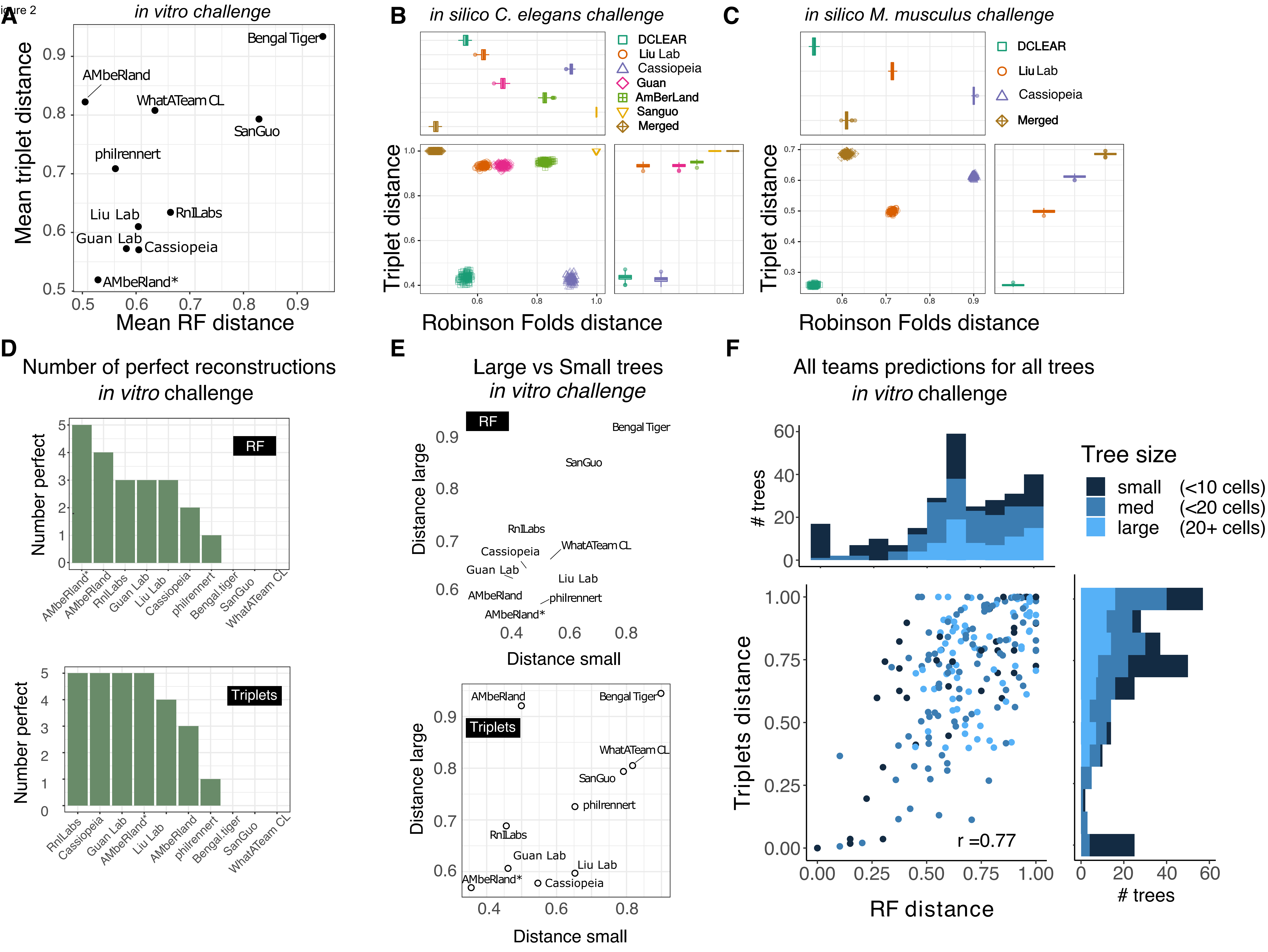
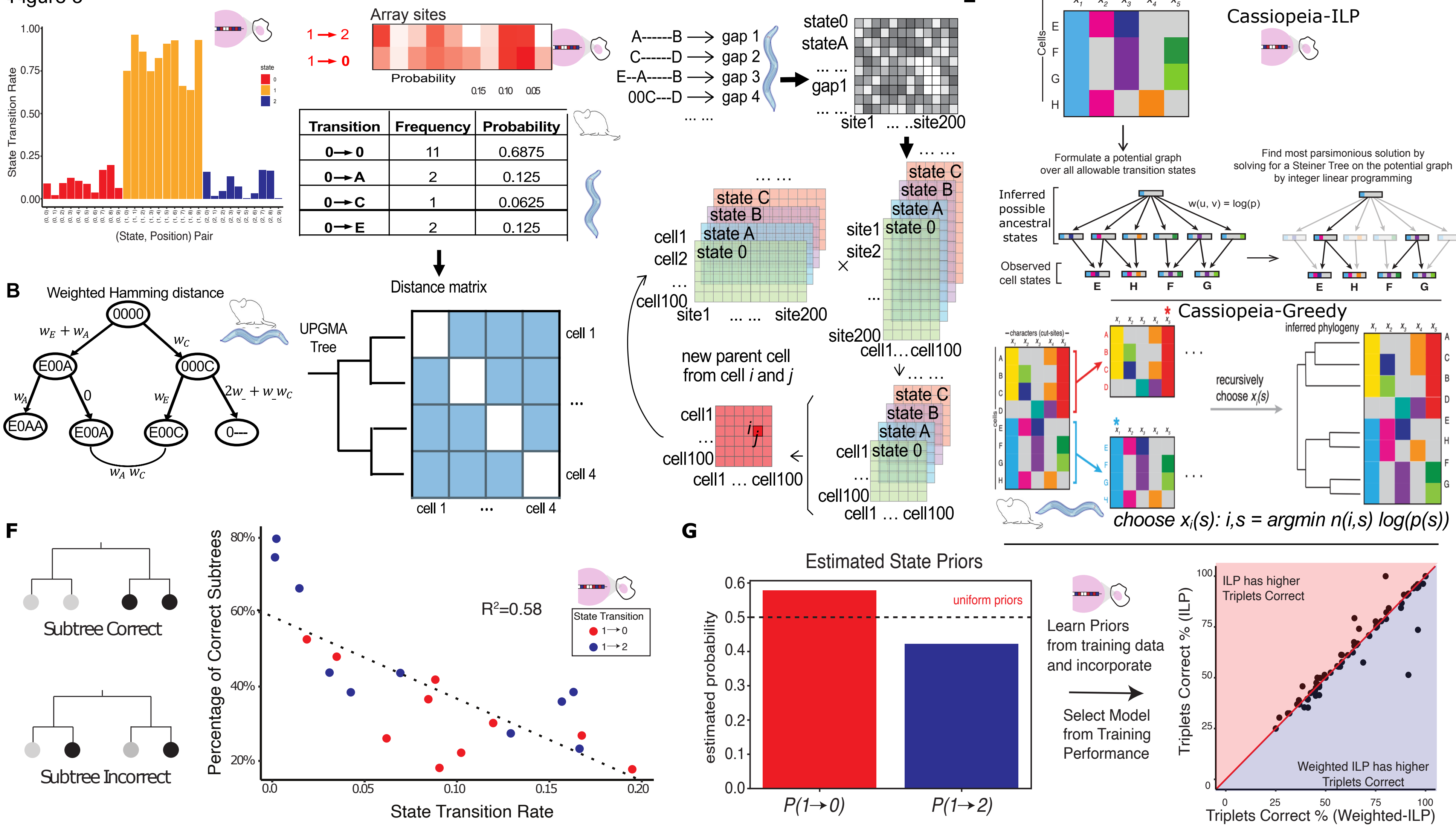

A
in vitro challenge
Mean triplet distance
Mean RF distance
Bengal Tiger
AMbeRland
WhatATeam CL
SanGuo
philrennert
Liu Lab
RnILabs
Guan Lab
Cassiopeia
AMbeRland*
0.5
0.6
0.7
0.8
0.9
B
in silico C. elegans challenge
DCLEAR
Liu Lab
Cassiopeia
Guan
AmBerLand
Sanguo
Merged
Triplet distance
Robinson Folds distance
C
in silico M. musculus challenge
DCLEAR
Liu Lab
Cassiopeia
Merged
Triplet distance
Robinson Folds distance
D
Number of perfect reconstructions
in vitro challenge
Number perfect
RF
AMbeRland*
AMbeRland
RnILabs
Guan Lab
Liu Lab
Cassiopeia
philrennert
Bengal.tiger
SanGuo
WhatATeam CL
Number perfect
Triplets
RnILabs
Cassiopeia
Guan Lab
AMbeRland*
Liu Lab
AMbeRland
philrennert
Bengal.tiger
SanGuo
WhatATeam CL
E
Large vs Small trees
in vitro challenge
RF
Distance large
Distance small
Bengal Tiger
SanGuo
RnILabs
Cassiopeia
WhatATeam CL
Guan Lab
Liu Lab
AMbeRland
philrennert
AMbeRland*
Triplets
Distance large
Distance small
AMbeRland
Bengal Tiger
WhatATeam CL
SanGuo
philrennert
RnILabs
Guan Lab
Liu Lab
AMbeRland*
Cassiopeia
F
All teams predictions for all trees
in vitro challenge
# trees
Tree size
small (<10 cells)
med (<20 cells)
large (20+ cells)
Triplets distance
RF distance
r =0.77
# trees

Figure 3



Transition	Frequency	Probability
0→0	11	0.6875
0→A	2	0.125
0→C	1	0.0625
0→E	2	0.125

Figure 4

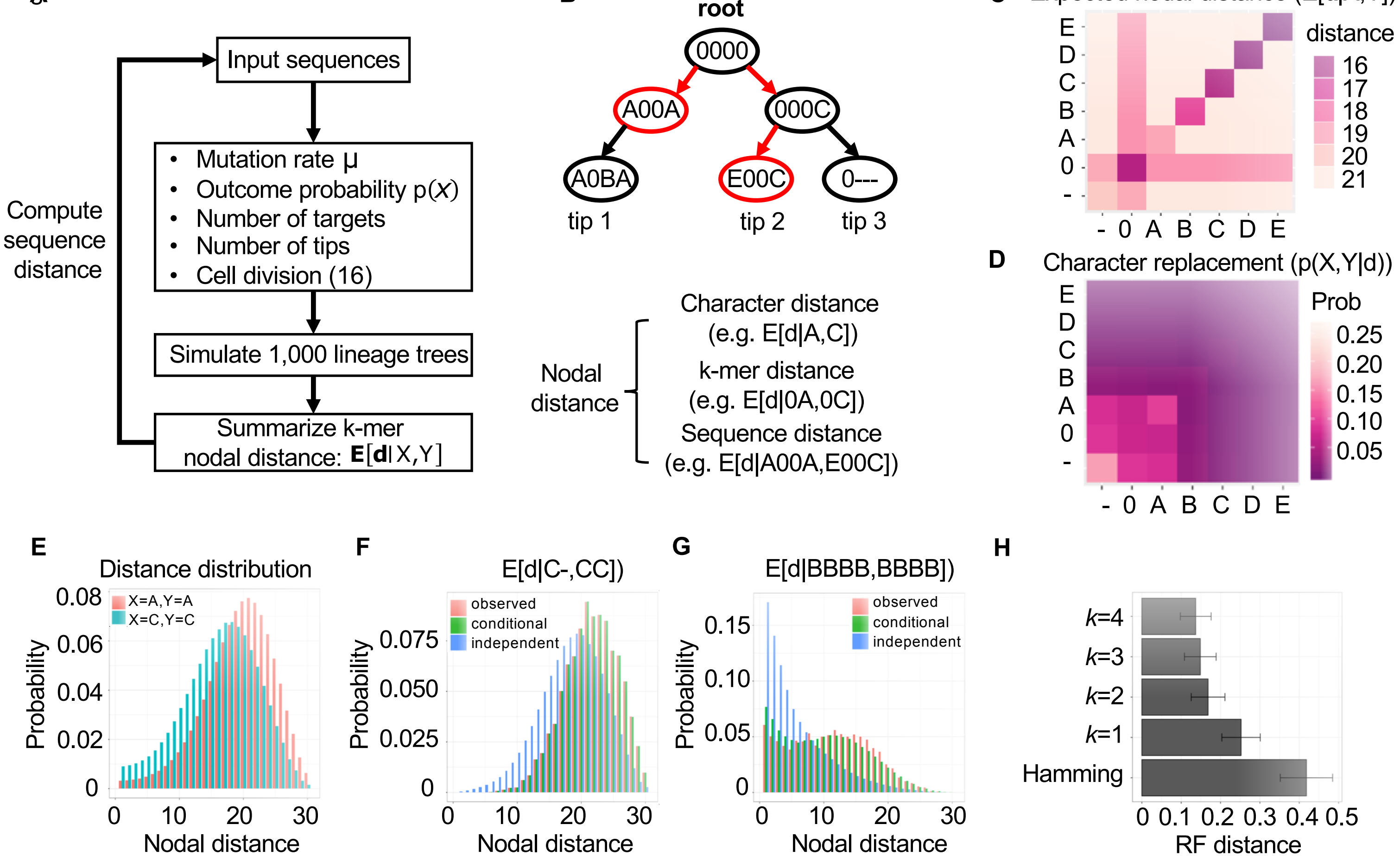

A
Input sequences
Mutation rate μ
Outcome probability p(X)
Number of targets
Number of tips
Cell division (16)
Simulate 1,000 lineage trees
Summarize k-mer nodal distance: E[d| X,Y]
Compute sequence distance
B
root
0000
A00A
000C
A0BA
E00C
0---
tip 1
tip 2
tip 3
Nodal distance
Character distance (e.g. E[d|A,C])
k-mer distance (e.g. E[d|0A,0C])
Sequence distance (e.g. E[d|A00A,E00C])
C
Expected nodal distance (E[d|X,Y])
distance
16
17
18
19
20
21
E
D
C
B
A
0
-
- 0 A B C D E
D
Character replacement (p(X,Y|d))
Prob
0.25
0.20
0.15
0.10
0.05
E
D
C
B
A
0
-
- 0 A B C D E
E
Distance distribution
X=A,Y=A
X=C,Y=C
Probability
0.08
0.06
0.04
0.02
0
0
10
20
30
Nodal distance
F
E[d|C-,CC])
observed
conditional
independent
Probability
0.075
0.050
0.025
0
0
10
20
30
Nodal distance
G
E[d|BBBB,BBBB])
observed
conditional
independent
Probability
0.15
0.10
0.05
0
0
10
20
30
Nodal distance
H
k=4
k=3
k=2
k=1
Hamming
0
0.1
0.2
0.3
0.4
0.5
RF distance

Figure 5

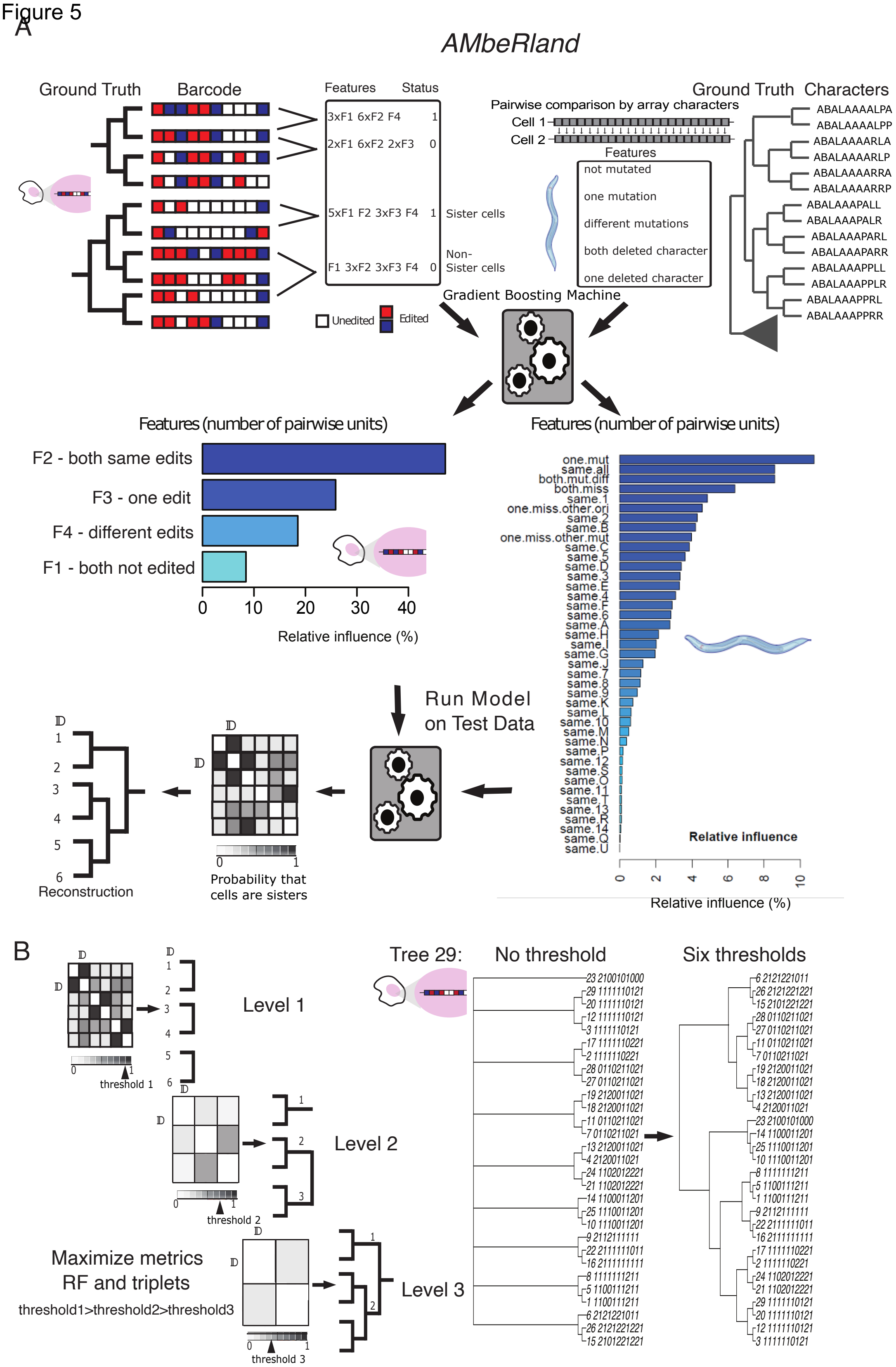
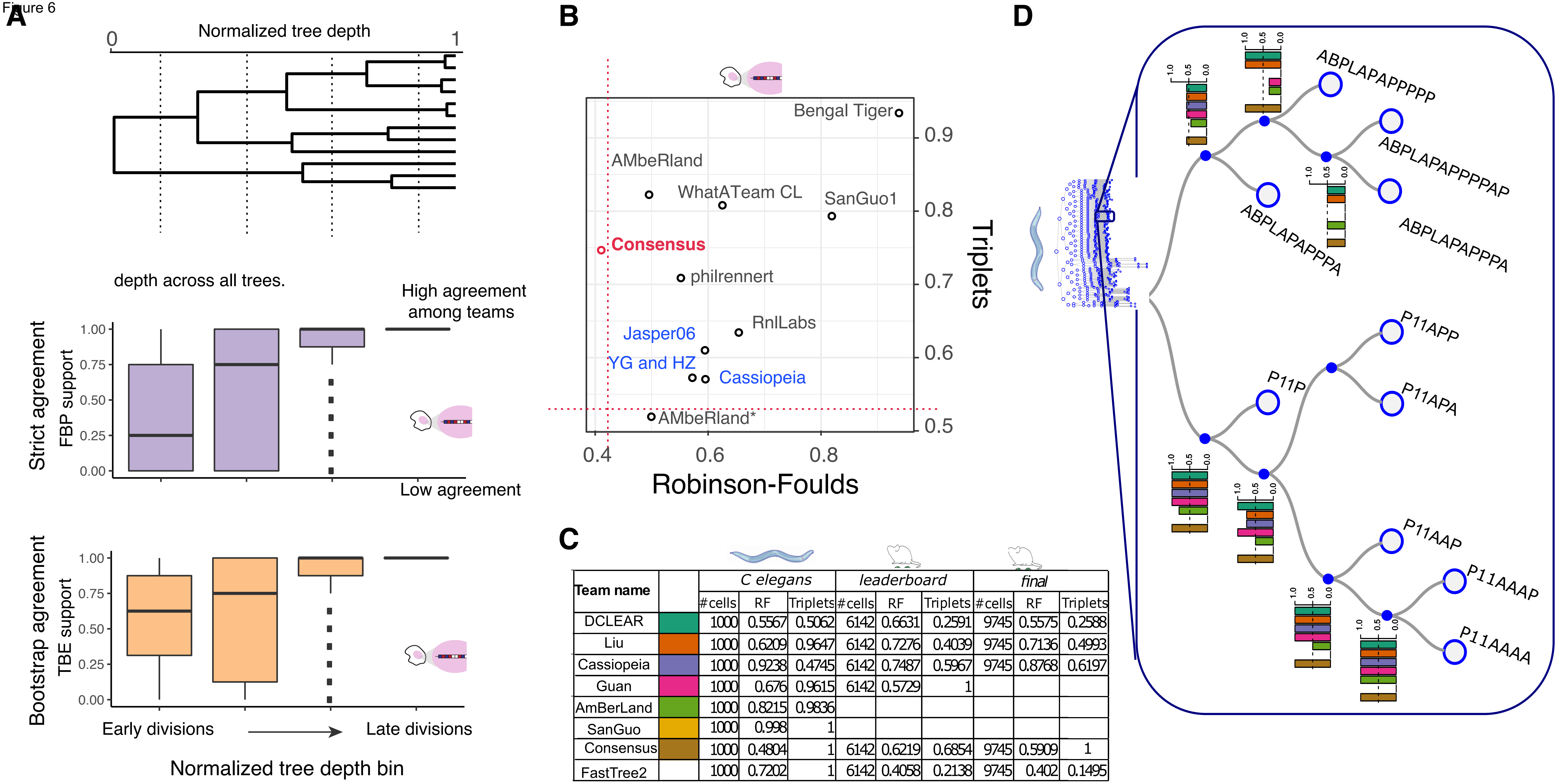

A
AMbeRland
Ground Truth
Barcode
Features
Status
3xF1 6xF2 F4 1
2xF1 6xF2 2xF3 0
5xF1 F2 3xF3 F4 1
F1 3xF2 3xF3 F4 0
Sister cells
Non-
Sister cells
Unedited
Edited
Pairwise comparison by array characters
Cell 1
Cell 2
Features
not mutated
one mutation
different mutations
both deleted character
one deleted character
Ground Truth
Characters
ABALAAAALPA
ABALAAAALPP
ABALAAAARLA
ABALAAAARLP
ABALAAAARRA
ABALAAAARRP
ABALAAAPALL
ABALAAAPALR
ABALAAAPARL
ABALAAAPARR
ABALAAAPPLL
ABALAAAPPLR
ABALAAAPPRL
ABALAAAPPRR
Gradient Boosting Machine
Features (number of pairwise units)
F2 - both same edits
F3 - one edit
F4 - different edits
F1 - both not edited
0
10
20
30
40
Relative influence (%)
Features (number of pairwise units)
one.mut
same.all
both.mut.diff
both.miss
same.1
one.miss.other.ori
same.2
same.B
one.miss.other.mut
same.C
same.5
same.D
same.3
same.E
same.4
same.F
same.6
same.A
same.H
same.I
same.G
same.J
same.7
same.8
same.9
same.K
same.L
same.10
same.M
same.N
same.P
same.12
same.S
same.O
same.11
same.T
same.13
same.R
same.14
same.Q
same.U
Relative influence
0
2
4
6
8
10
Relative influence (%)
Run Model
on Test Data
ID
ID
0
1
Probability that
cells are sisters
ID
1
2
3
4
5
6
Reconstruction
B
ID
ID
0
1
threshold 1
ID
1
2
3
4
5
6
Level 1
ID
ID
0
1
threshold 2
1
2
3
Level 2
ID
ID
0
1
threshold 3
1
2
Level 3
Maximize metrics
RF and triplets
threshold1>threshold2>threshold3
Tree 29:
No threshold
23 2100101000
29 1111110121
20 1111110121
12 1111110121
3 1111110121
17 1111110221
2 1111110221
28 0110211021
27 0110211021
19 2120011021
18 2120011021
11 0110211021
7 0110211021
13 2120011021
4 2120011021
24 1102012221
21 1102012221
14 1100011201
25 1110011201
10 1110011201
9 2112111111
22 2111111011
16 2111111111
8 1111111211
5 1100111211
1 1100111211
6 2121221011
26 2121221221
15 2101221221
Six thresholds
6 2121221011
26 2121221221
15 2101221221
28 0110211021
27 0110211021
11 0110211021
7 0110211021
19 2120011021
18 2120011021
13 2120011021
4 2120011021
23 2100101000
14 1100011201
25 1110011201
10 1110011201
8 1111111211
5 1100111211
1 1100111211
9 2112111111
22 2111111011
16 2111111111
17 1111110221
2 1111110221
24 1102012221
21 1102012221
29 1111110121
20 1111110121
12 1111110121
3 1111110121

Figure 6



Team name		C elegans			leaderboard			final		
		#cells	RF	Triplets	#cells	RF	Triplets	#cells	RF	Triplets
DCLEAR	🟩	1000	0.5567	0.5062	6142	0.6631	0.2591	9745	0.5575	0.2588
Liu	🟧	1000	0.6209	0.9647	6142	0.7276	0.4039	9745	0.7136	0.4993
Cassiopeia	🟪	1000	0.9238	0.4745	6142	0.7487	0.5967	9745	0.8768	0.6197
Guan	🟥	1000	0.676	0.9615	6142	0.5729	1			
AmBerLand	🟩	1000	0.8215	0.9836						
SanGuo	🟨	1000	0.998	1						
Consensus	🟫	1000	0.4804	1	6142	0.6219	0.6854	9745	0.5909	1
FastTree2		1000	0.7202	1	6142	0.4058	0.2138	9745	0.402	0.1495