Cell Lineage Reconstruction using Felsenstein's and Markov Chain Monte Carlo

Vincent Chang, Zhonghao Dai, Weida Ma

Abstract

Determining the lineage of cells offers a foundation for understanding when, where, and how cell fate decisions are made. Molecular bioengineering techniques have advanced in recent years to the stage where it is possible to reconstruct cell lineages based on phylogenetic analysis of DNA mutations. To solve the challenging issue of determining the lineage of complex model organisms at single-cell resolutions, CRISPR/cas9 is used to synthetically introduce inheritable and irreversible point mutations. By reading the mutations on the latest generation of cells, a recording array can be used to reconstruct a phylogenetic tree that represents the evolutionary relationship between all cells. Cell-lineage reconstruction is a relatively new field of computational phylogenetics, so naturally there are associated challenges. Firstly, current lineage reconstruction algorithms have yet to be examined for their robustness across a variety of datasets ranging in nature and size. Secondly, phylogenetic algorithms are well-suited for analysis of data that are fundamentally structured differently compared to cell-lineage tracing data (more on this in the background). In essence, there are no current state-of-the-art algorithms for cell-lineage data in computational phylogenetics, and this challenge seeks to leverage the knowledge of the crowd to find one. The approach this project takes on involves a combination of Felsenstein's algorithm, local rearrangement swapping, and Markov Chain Monte Carlo sampling. The goal is to calculate a score for a randomly initialized tree using Felsenstein's algorithm, perform local rearrangement to generate a new tree, rescore the tree, and update the score based on a probabilistic model. The algorithm will iterate this process until it converges at the highest scoring tree. The algorithm is compared to a "sister-likelihood" adaptation of the classical neighbor-joining method of reconstructing trees using a score metric defined as 1-RF, where RF is the Robinson-Foulds distance between the reconstructed tree and the ground-truth tree. The proposed algorithm yields reconstructions with statistically significantly higher scores than the sister-likelihood method across data sets of varying number of cells. The inherent probabilistic nature of this approach and the results of this study imply the robustness of this algorithm across data sets of varying number of cells. Future studies may optimize the algorithms performance and runtime as well as investigate the performance and robustness of this algorithm across various molecular tools beyond CRISPR/cas9.

Availability and implementation

Source code of Cell Lineage Reconstruction using Felsenstein's and Markov Chain Monte Carlo, training sets containing readout for each cell, ground-truth datasets, and RFdist function provided by the DREAM 2019 Challenge used in this paper are available at:
https://github.com/vincentc122013/Cell-Lineage-Reconstruciton-Felsensteins-MCMC

Background

All multicellular organisms begin their lives as a single cell. That single cell goes through countless rounds of division, eventually growing into a collection of billions or trillions of cells. These cells, however, are not identical to one another or to their originator in general. Constructing the cell's lineage offers insight into how differentiation occurs and provides a framework for understanding when, where, and how cell fate decisions are made.

Recent advancements in molecular bioengineering has made the reconstruction of cell lineages based on the phylogenetic analysis of DNA mutations possible. Namely, CRISPR-based molecular tools can synthetically induce mutations during development, making them promising tools to help solve the lineage of complex model organisms at single-cell resolutions.[1] CRISPR/Cas9 is a system that acts on programmed sites to introduce mutations through insertions, deletions, or edits. CRISPR finds the targeted site and the Cas9 enzyme acts as the molecular scissors that initializes the intended changes. CRISPR-induced mutations occur stochastically starting in early embryogenesis. The stable inheritance of these mutations is ensured in most cases by the destruction of the match between the target

and sgRNA, making the target site immune to further change. At the end of development, the accumulated mutations can then be used as phylogenetic characteristics on a recording array that permits the reconstruction of a tree representing the evolutionary relationship between all cells.[2]

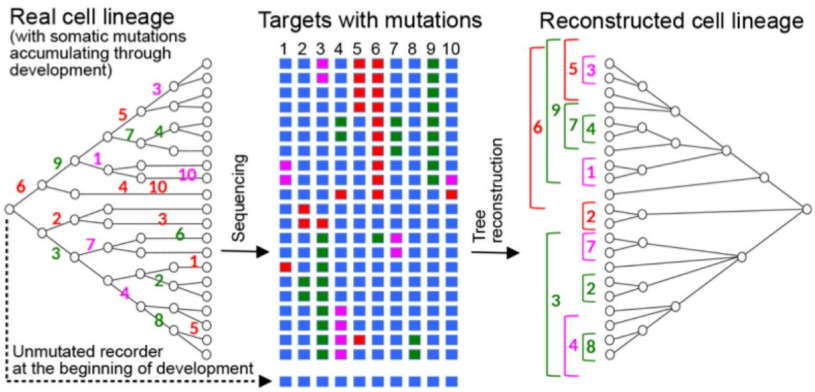

Figure 1. Schematic of reconstructing cell lineages from recording array of CRISPR-induced mutations
Left: Ground-truth of cell lineage with introduced mutations marked as number values during cell division. Center: Recording array of every cell after sequencing (blue boxes represent CRISPR targets, magenta, red, and green boxes represented introduced mutations). Right: Reconstructed cell lineage using recording array readout.[2]

Accurate reconstruction of phylogenetic trees using only the recording array of introduced mutations is a challenging problem as it requires a reliable lineage reconstruction algorithm. Current lineage reconstruction algorithms have never been rigorously examined for their performance/robustness across diverse molecular tools, datasets, and number of cells/size of lineage trees. It also remains unclear whether or not innovative Machine-Learning algorithms that reach beyond the scope of classical phylogenetic algorithms such as Neighbor-Joining and UPGMA can consistently achieve higher levels of performance in cell lineage reconstruction.[3]

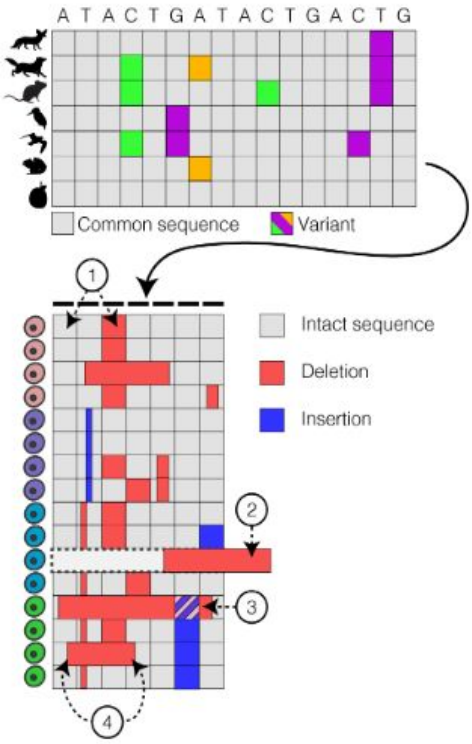

Figure 2. Schematic of data for "traditional" phylogenetic studies where there are few species and more sites/nucleotides involved (top) vs. cell lineage tracing phylogenetic studies where there are many more cells and only a few mutable sites (bottom)
Top: Colored squares indicate variants (nucleotide mutations) of the respective species. Bottom: 1 indicates variable mutability rates among different sites, 2 indicates large deletion creates dropouts, 3 indicates deletion erases previous mutations, and 4 indicates deletion that erodes multiple target sites (the data used by this report will only involve variable mutability rates)

Another challenge for phylogenetic studies based on nucleotide changes within cells is the unconventional set of input data. Traditional phylogenetic studies study small number of species over many more sites, genes, and sometimes even whole genomes. Data matrices for these studies are generally short and wide, where there are only a few proverbial “cells” (species) to work with, and a wealth of information available for each species that can be used for classification. Phylogenetic studies for cell lineage tracing is the opposite. Studies usually involve hundreds or thousands of cells and only a few mutable sites. This results in data matrices that are tall and narrow, where there are a massive number of cells that need lineage tracing and a limited amount of available information to work with. These differences make it difficult to directly apply phylogenetic methods designed for wide and short matrices to lineage-tracing data.[4]

Our project is an attempt at delivering a robust algorithm for constructing phylogenetic trees using lineage-tracing data. We implemented an algorithm that uses Felsenstein’s algorithm in conjunction with a local rearrangement swapping approach and Markov Chain Monte Carlo sampling to reconstruct *in vitro* lineages of cells from data obtained by the Elowitz Lab at the California Institute of Technology. Video-microscopy data captures inheritable and irreversible mutations that occur over a 48 hour period. This information is used to establish the ground-truth. To predict cell lineage, the only information available for phylogenetic tree reconstruction is the recording readout of the final state each cell is in after this incubation period. The recording readout is a barcode with 10 digits for each cell, signifying 10 mutable sites. We assume the root’s identity is a vector of ten 1s: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1], and there are two possible mutations the CRISPR-cas9 system can introduce: transitions from state 1 to state 0, or state 1 to state 2. As mutations are inherited and irreversible, subsequent daughter cells will hold all mutations from its parent (transitions out of state 0 and state 2 are impossible).[3]

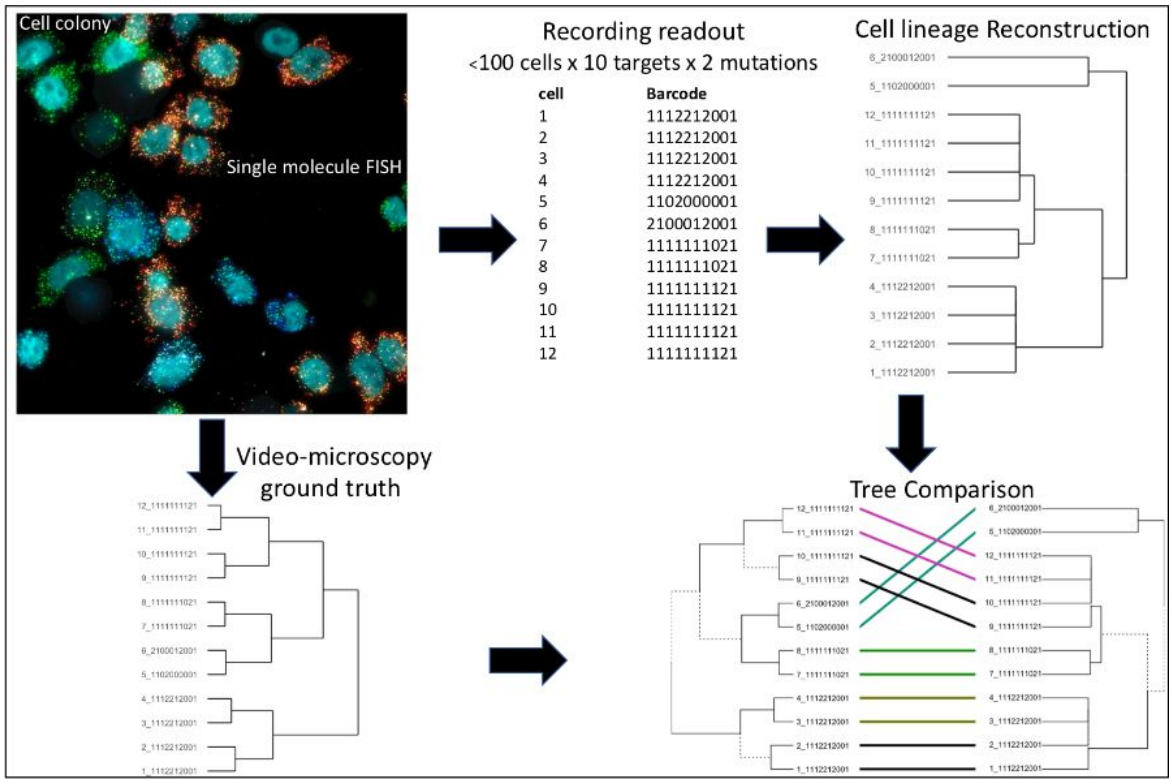

Figure 3. Schematic representation of the project’s overall goal.[3]

Recording readout of the final state of each cell is used to build an algorithm that reconstructs a predicted phylogenetic tree. The predicted phylogenetic tree is compared with the ground-truth tree obtained from video-microscopy and scored using the Normalized Robinson-Foulds distance.

The performance of the algorithm will be assessed by comparing the output prediction with ground-truth video-microscopy data, as well as a current algorithms in phylogenetics such as UPGMA and sister-likelihood. Our scoring metric will be based on the Normalized Robinson-Foulds distance, defined by 1-RF (higher RF value signifies larger discrepancies between the predicted tree and the ground-truth tree). This metric compares the quality of each algorithm's prediction relative to the ground-truth, where a value of 1 signifies a perfect reconstruction and 0 represents a random guess.

Successful completion of this project can offer novel insight on accurately determining the phylogeny of cells for lineage-tracing data. This information can play a crucial role in determining where specific mutations are introduced into a cell's lineage and determining the key points where cell fate decisions are made.

Results & Discussion

Selection of methodology

In order to reconstruct cell lineages from only final recording readouts, we chose to implement a Markov Chain Monte Carlo approach to sample various tree topologies and determine the optimal solution. The data provided by the DREAM 2019 Challenge consists of over 70 recording readouts of cell barcodes. Each data set contains a varying number of cells. Ground-truth trees in Newick format and results from a "sister likelihood" reconstruction method are also provided for evaluation of performance. The sister likelihood method is an adaptation of the neighbor-joining method that evaluates pairwise probabilities of cells being sisters and compares these probabilities to the probability of observing the barcode independently via random mutation. Sisters which are highly likely to be paired together (i.e. minimal distance from each other) are paired together to reconstruct the tree from the bottom up. Our proposed algorithm instead evaluates probabilities spanning over the entire tree topology.

Our algorithm is implemented primarily in MATLAB and utilizes the phylogenetic analysis tools of the Bioinformatics Toolbox. The first step in our project was to simply understand the data structure of phylogenetic objects in MATLAB. Each "phytree" object is composed of three key features: pointers, branch lengths, and node names. Branch lengths and node names are simply lists containing the branch lengths and names of each node within the tree. Pointers are an n-1 by 2 matrix containing information about the structure of the tree. Since n represents the number of leaves in the tree, there are n-1 internal nodes. Therefore, each row of the pointers matrix represents an internal node and its elements are the children of that internal node.

Tree topology scoring

The ability to choose an optimal tree topology relies on being able to score different topologies. The basis for our scoring methodology is a substitution model of evolution at each barcode mutation site. In order to determine the parameters of our evolutionary model, we loaded in all of the barcodes available to us from every data set and counted the number of 0s, 1s, and 2s in the final readouts. We then normalized these counts in order to estimate the probabilities of mutations occuring in order to generate a substitution matrix for each mutation site.
The next step in our algorithm is to generate an initial tree topology and score it. In order to generate the initial tree, we used the built-in MATLAB functions seqdistp, which calculates pairwise distances between sequences, and seqlinkage, which constructs a phylogenetic tree using pairwise distances and the UPGMA method. The algorithm we chose to use to score our tree topologies is Felsenstein's algorithm. Felsenstein's algorithm is a dynamic programming approach which calculates the probability of sub-trees in order to calculate the probability of larger trees. One of the greatest advantages to this algorithm is that the scores of subtrees can be stored within a matrix, which we will call V. V(i,j) contains the score of the tree under node j given that it takes the identity i. In a model of nucleotide substitution, this matrix will contain 4 rows for each possible identity the nucleotide can assume, A,C,G and T. However, in our problem, each node can assume one of 3^{10} possible identities because each cell is composed of 10

mutation sites where there are 3 possible states. The computational expense of computing and storing the scores for all 3^{10} possible identities renders this approach infeasible. Alternatively, we chose to instead write a function that returns all of the possible parents of an input barcode given the irreversibility of mutations. Using this approach, we can initialize a smaller V matrix and update it dynamically as our list of possible parents grows as long as we are careful in how the matrix is indexed.

With the data infrastructure in place, the scoring algorithm works as follows. For any leaf node, it's score is initialized as 1. For each internal node, if the children of the node are both leaves, the algorithm iterates over the intersection of all of the possible parents of each barcode, scoring the probability of a substitution at each site from the parent into the leaves. This score is then stored in the V matrix under the row corresponding to the parent identity and the column corresponding to the parent node number and all of the possible identities of the internal node (i.e. the intersection of the parents of the leaves) are stored in a list. If one of the children is an internal node and one is a leaf, the algorithm retrieves all of the possible identities of the child internal node in order to calculate all of its possible parents. The intersection of this list and the possible parents of the child is taken and iterated over. The score tree under the parent internal node is calculated by retrieving and summing the scores under the child internal node from the V matrix and multiplied by the substitution probabilities from the parent internal node to the child internal node and the leaf. These scores are then placed in the V matrix under the column corresponding to the parent node number and the row corresponding to its identity. If both of the children of the internal node are internal nodes themselves, the intersection of the list of all possible parents of each internal node are calculated similarly to the cases above. The score under the parent node is calculated by retrieving the relevant scores from the V matrix and multiplied by the substitution probabilities, similarly to the other cases. Finally the score is stored in the V matrix as in the other cases. The algorithm terminates once it reaches the root node.

While the rationale of scoring internal nodes whose children are either one internal node and one leaf or two internal nodes remain grounded, it was quickly realized that scoring an internal node whose children are both leaves by scoring the probability of a substitution at each site from all of their possible parent combination is undesirable since this approach does not account for the similarity of the two leaves' barcodes. It is evident from analysis on several high scoring tree topologies that cases exist where two leaves have very different barcodes, but score highly because two different nodes have more potential parents and subsequently the scores stored in the V matrix cascaded to upper nodes, eventually leading to a higher score than topologies which are more similar to ground-truth. To address this issue, alignment of the two barcode from leaves is scored via the Needleman Wunsch algorithm.[6] Subsequently this alignment score is multiplied to score of parent probability before being stored in the V matrix. This algorithm favors tree topologies which contain higher number of two-leaves internal nodes, and two leaves being similar to one another—both are biologically rational.

Generating new tree topology

After the initial tree is scored, a new tree topology must be generated to compare to the initial topology. One strategy for generating a new tree topology is Local Rearrangement.[5] In the Local Rearrangement strategy, one internal node is chosen at random to be the target node. One of the two children of the target node is then randomly swapped with the sister of the target node, generating a new tree topology. This strategy is theoretically capable of generating any tree topology in a finite number of steps, making it a good candidate for our Markov Chain Monte Carlo sampling approach. However, the nature of this approach in generating trees within a "local neighborhood" of the initial tree potentially limits the range of tree topologies we are able to sample given our computational and time constraints. Iteration of our algorithm is rather slow, and thus only a few hundred to a few thousand samples are possible, potentially excluding many tree topologies.

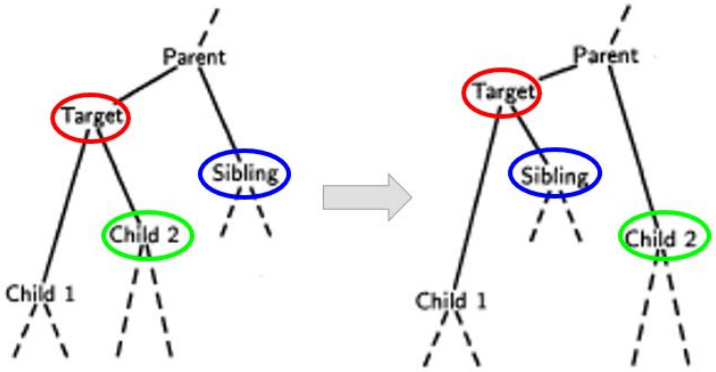

Figure 4. Schematic representation of the Local Rearrangement process.
The generation of a new tree topology follows follows three major steps as follows:
Step 1: Select an interior node at random to be the target node.
Step 2: Randomly select one of its children nodes.
Step 3: Swap the chosen child with the target's sister.

The newly generated tree via Local Rearrangement is then scored using the scoring algorithm described above. A ratio of scores between the new tree and the old tree, R = S(new)/S(old) is calculated. Using a Metropolis MCMC approach, if R ≥ 1, the new tree is accepted as the sample and if R < 1, R becomes the probability by which the new tree is accepted as the sample.

Selecting the optimal tree topology

After iterating this process several hundred to a thousand times, a histogram of sampled trees can be generated. We infer that the tree that has been sampled the most maintains the best score and is therefore the optimal topology. Due to the principle of Metropolis MCMC, the score of this optimal tree topology is usually also the highest score. Still, We have observed occasionally that the highest score is not the most frequent tree configuration sampled. An in-depth inspection reveals that this phenomenon is most likely to be resulted from the semi-stochastic nature of generating tree topology, since there still exists a probability of keeping a lower score tree as sample for the subsequent round of topology remodelling.

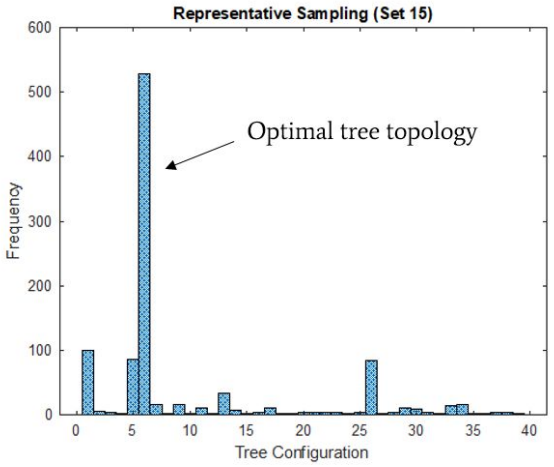

Figure 5. Representative sampling result from one training dataset.

Comparing generated topology with ground-truth

After selecting the optimal tree topology using Metropolis MCMC approach, this topology is exported into Newick format and imported into R. The R phangorn package includes the *RFdist* function, which calculates the normalized Robinson-Foulds distance between two trees. A RF distance of "1" represents that the two tree topologies are completely different with one another while "0" represents identical topology. The RF distance between our reconstruction and the provided ground-truth tree is calculated and qualitatively examined with their built-in topology graphing tool as shown in Fig 6. Two RF results are provided from the *RFdist* function, one being for unrooted tree and the other being rooted tree. For this figure in particular, RF=0 for unrooted, and RF=0.1343 for rooted. Rooted results are selected for comparison purposes since cell lineage is a rooted tree. One noticeable aspect of scoring using RF distance is that the branch length of the tree topology is not taken into consideration.

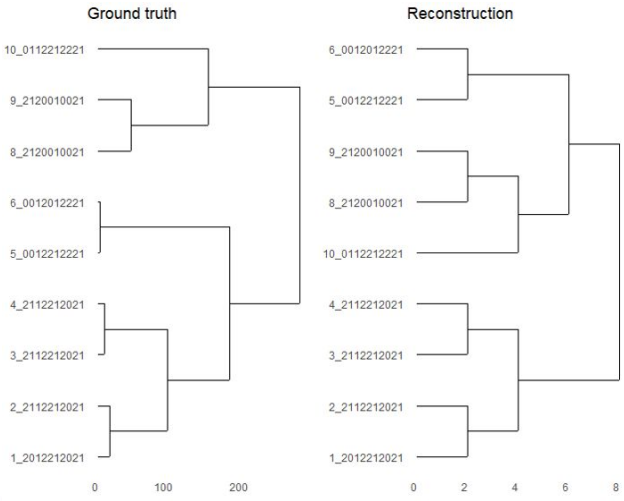

Figure 6. Representative comparison between Ground truth and Reconstructed tree

Performance evaluation

In order to evaluate the performance of our cell lineage reconstruction method, we used the first 50 training datasets provided by the DREAM 2019 Challenge, which contained the readout of cell barcodes, as well as the ground-truth. Optimal tree topology selected based on our method is scored using RF distances described in previous section. These RF distances are also compared to the RF distances between the provided results of the sister-likelihood method and the ground-truth.

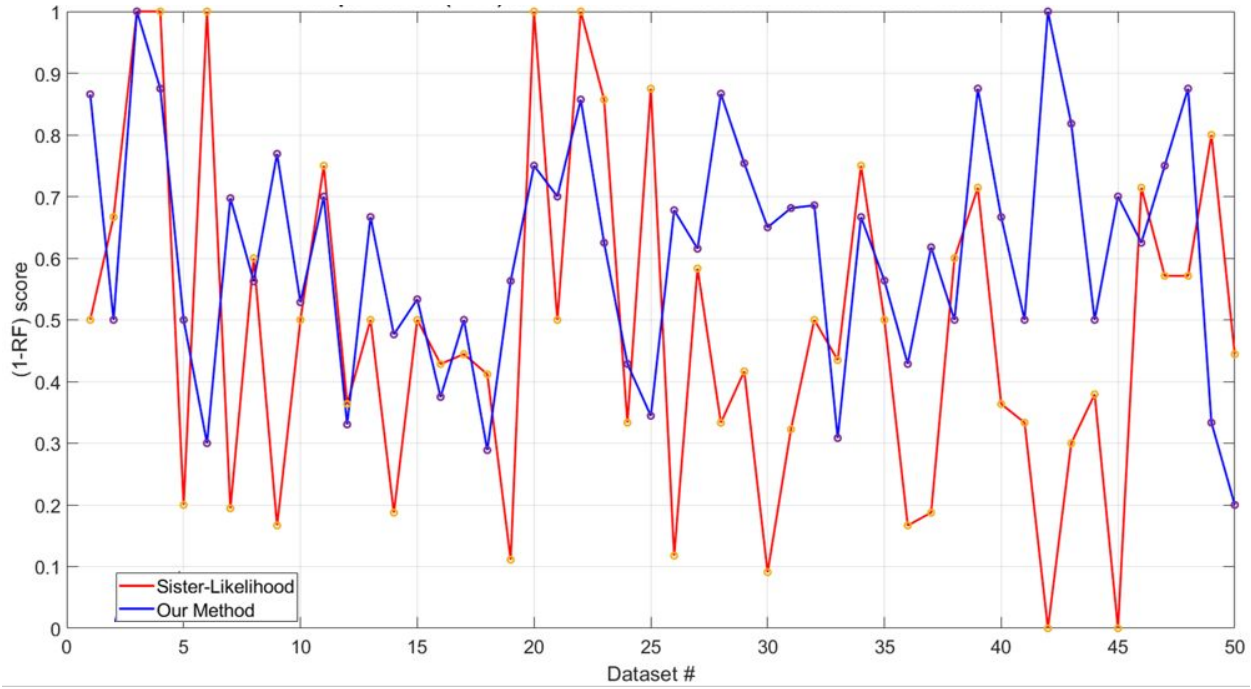

Figure 7. Comparison of (1-RF) score for optimal tree topology generated by Sister-Likelihood method vs. optimal tree topology generated from our method

The average (1-RF) score of tree topology generated for the first 50 dataset is as follows:

Sister-Likelihood method: 0.4857 ± 0.2741

Our method: 0.6049 ± 0.1838

A paired-sample t-test is performed using MATLAB, with one-tailed P value equals 0.0250. By conventional criteria, the difference in score is considered to be statistically significant, meaning that our method outperformed traditional tree building method in terms of accuracy when compared with ground-truth. It also suggests that our implemented scoring method is in agreement with the biological rationale of cell lineage mutations induced by CRISPR; the cell lineage topology that leads to the formation of end-result cell barcodes will have a high probability of branching subtrees that also contain high probability from their children nodes, and so on until the branching that have two leaf nodes. Conversely, a convergent reconstructed tree topology using a scored-based Markov Chain Monte Carlo sampling leads to the selection of tree topology in close proximity to topology of ground-truth. It is also important to mention that the number of cell barcodes each dataset contain is not uniform but ranging from 4 cells to as many as 35 cells. It is consequently shown that our algorithm is able to perform better across varying number of cells, an important characteristic that contributed to the overall robustness of our approach. In addition, one would observe that the (1-RF) score using our method shows a relatively lower standard deviation of ±0.1838 compared to that of Sister-Likelihood, which has formed some awful reconstructions with 0.2 or lower (1-RF) score from time to time. Therefore, while our method still lacks the ability to reconstruct almost-perfect topology, the comparatively higher score and increased robustness implies that researchers can more confidently rely on the tree topology selected

based on our method while gathering additional cellular features that might lead to marginal branch swapping before reaching ground-truth topology, instead of being wary of the probability of reconstructing a completely wrong lineage as starting ground using Sister-Likelihood and other traditional approaches.

As robust as our method appears, it is noticeable from Fig 7 that not every cell lineage reconstruction generated using our method is more accurate than by Sister-Likelihood. In fact, Sister-likelihood results provided by the challenge contain a higher number of perfectly reconstructed trees. We therefore analyze these tree topologies and barcodes, attempting to explain why our method is incapable of reconstructing the perfect tree topology. It is quickly observed that except for training set #4, which ironically contains four cells with identical barcodes, the imparity of topology from our method is mainly contributed to by branching of incorrect internal nodes. By diagnosing the score stored in V matrix, we found out that there exists issue analogous to that arises from scoring two-leaves nodes; For the sister-likelihood method, internal nodes whose children are two internal nodes are frequently generated from 4 leaves, with 2 sharing the same barcodes, and the other 2 sharing another barcodes that isn't identical to the first, but in close resemblance. This ultrametric distance approach is biologically rational for this challenge since rate of barcode mutation is roughly constant with time, therefore time is proportional with the degree of divergence. This aspect is ignored in our method, in which we only take in consideration of potential parent barcodes and the scoring of them by the substitution probabilities. Therefore a similar situation is found for those internal nodes that the same sum of parents' score can be obtained when it's four grand-children nodes have distinct, or similar barcode. Yet due to the limitation of time as well as the coding complexity of incorporating ultrametric distance approach in nodes whose branches are internal nodes, we are unable to resolve this issue but we certainly hope to as future goals.

Another aspect of cell lineage reconstruction we latter become aware of by observing many ground-truth data is the rare appearance of leaf-internal node branching, or stair-shaped topology structure, something that is rather commonplace in reconstruction generated by both our method and Sister-Likelihood methods. Biologically speaking, leaf-internal node branching should be rare to observe since cell mitosis is a process of forming parent daughter cells from one parent cell, where the CRISPR-induced mutations will be phenotypically exhibited. Meanwhile, the stair-shaped topology resembles a process more similar to a subspecies diverged from main species, or in analogous, a parent cell "give birth" to subsequent daughter cells. For the context of this study, while leaf-internal node branching is possible if the signal of one end cell is lost due to apoptosis and only leaving its "sister" cell observable for barcode reading, stair-shaped topology should be highly unfavorable. One potential adjustment to be made is adding a penalty factor when forming stair-shaped topology, or take branch length into scoring consideration, since stair-shaped structure would result in substantially longer branch length between leaf nodes and ancestor nodes, subsequently making the tree topology unfavorable.

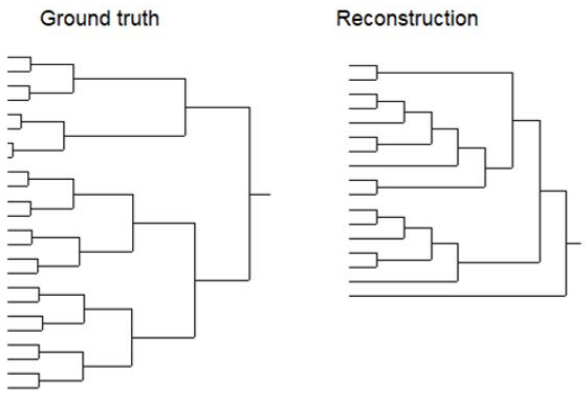

Figure 8. Comparison of ground-truth cell lineage and reconstructed cell lineage with presence of Stair-shaped substructures

One last concern of our current method is the high computational expense of scoring the probability of the tree topology. In our problem, each node can assume one of 3^{10} possible identities because each cell is composed of 10 mutation sites where there are 3 possible states. While the computational expense of computing and storing the scores for all of the possible parents of an input barcode given the irreversibility of mutations is comparatively less than storing the probabilities of all 3^{10} possible identities, we have encountered the presence of two-leaves branching that results in the formation of 2^{10} possible identities. The fact that this computation needed to be applied to all

internodes and cascaded bottom-up to ancestor node, the complexity of the problem increases with higher number of cells in one dataset. Additionally, since the presence of more barcode could theoretically allow better reconstruction of cell lineage, we envision that the number of barcode per cell would be much higher than 10 digits, or more than 3 possible states, which would subsequently leads to exponential growth of computational cost. The computational cost can be alleviated if the probability of X^Y possible identities is first calculated and stored in a data matrix, subsequently our algorithm can directly refer to the probability of each potential parents' barcode with excessive and repetitive computation. An alternative would also be that we only store a number of highest possible parents' probability in the V matrix; this would allow cascaded computation to upper nodes much simpler. But still, the approach of neglecting parents with low probability might result in none of the potential parents node matches at nodes closer to ancestor node. An in-depth evaluation would be crucial before finding an optimal tradeoff between computational cost and loss of probability score.

Future Goals

As comprehensively described in our Results & Discussion section, we consider our approach of reconstructing cell lineage using Felsenstein's and Markov Chain Monte Carlo to be biologically rational, and computationally robust. By comparing with ground-truth using Robinson-Foulds distance metric, we have shown that our algorithm allow reconstruction of cell lineage topology with higher accuracy and greater robustness compared with that of traditional methods. To take this project forward, we hope to address several apparent issues still existed in our algorithm before it is ready for prime-time. Some future works include but are not limited to:

- Include ultrametric distance metric when computing branching probability
- Incorporate a penalty for very long cell lineages composed of many leaf-internodes branching and stair-shaped topology structure / Include branch length as a parameter when scoring topology
- Reduce computational expense

We sincerely hope that by resolving these identified issues, the subsequent sampling of tree topology would allow us to reconstruct cell lineage with much higher accuracy than the result we currently present. We would like to note that the approach introduced in our paper is by no means the only—and not necessarily the optimal—way of reconstructing such cell lineage. We expect that the overall accuracy of the cell lineage reconstruction may be further improved in the future by topology-scoring and topology-editing models networks that are further enriched with additional information that may be useful in predicting the accurate cell lineage. Devising and incorporating novel Machine Learning models that are optimized for predicting cellular mutation and heredity may potentially enhance the speed and accuracy of the cell lineage reconstruction algorithm even further.

Project Summary of Process

Comparison with Original & Revised Proposal

In the original proposal, the plan was to reconstruct the phylogenetic trees using a Viterbi modification on Felsenstein's algorithm. The thought process is that, by taking the max instead of the sum when applying Felsenstein's algorithm to the leaves, we will be able to effectively determine all the leaves ancestors. This is precisely the issue with the original approach, as decoding ancestor identity did not provide further insight towards tree topology. To work around this issue, we needed a method of sampling from multiple different topologies and scoring each one (using Felsenstein's algorithm) to determine the "best" or most probable path the mutations had occurred in. This step was the biggest change between the algorithm's design from the final version to the original proposal.

Regarding how we addressed our reviewer's suggestions, the biggest concern that came up was the fact that Aim 2 (assessing the performance of our algorithm by comparing its predictive power vs. traditional phylogenetic algorithms like sister-likelihood and UPGMA) is heavily dependent on the success of Aim 1 (establishing an algorithm that can predict/reconstruct tree topology). We recognized the risk involved with establishing aims that happen chronologically within the proposal, but as our project stemmed around the challenge of predicting cell lineage, getting a working

algorithm that can perform the task is pretty much a mandatory first step. We completely understood the concern around our aims being established in this manner, but didn't see an option to remedy the issue. Other concerns from reviewers that involved introducing organizations and concepts more thoroughly, or rearranging information to better suit the section they were supposed to be in, were considered and implemented in the final proposal and the report.

Commentary on Experience

The biggest challenge we faced throughout this project was understanding the current algorithms for phylogenetic tree reconstruction, their limitations, and how to adapt promising algorithms to our specific challenge's constraints. The most valuable piece of advice for us would've been to start the implementation of the Viterbi approach on Felsenstein's earlier. Having a theoretical understanding of what an algorithm should be able to do versus programming it ourselves and recognizing its weaknesses and limitations are two very different things. If we had understood our misinterpretations for the potential of our original algorithmic design earlier, it would've given us more time to redesign the algorithm and potentially resolve a few of the issues we bring up in the future goals section of the report.

Looking back at the datasets we used for the project, we realized it wasn't ideally suited for Felsenstein's algorithm. A unique aspect of our input barcode/data is that mutations from state 1 are irreversible. The irreversibility of mutations has severe implications on the number of potential parents between two barcodes. Two barcodes that should be "good fits" can end up scoring lower than two barcodes that are "bad fits" if the previous case had fewer possible parents than the ladder case. As a result, the results were heavily skewed the first few times we ran the algorithm. This issue was remedied by adjusting the weights on barcodes that had high levels of similarity in our algorithm.

Overall, we felt like we struck a good balance between doing and writing for this project. We were 100% dedicated to the "doing" aspect before we concerned ourselves with the writing, as without solid results and a thought-out design process, there wouldn't be a lot of meaningful content to write. By focusing purely on the coding, we were able to achieve decent results within the time frame of the course, which was definitely both the most challenging and rewarding part of the project.

Commentary on the Peer-Review Process

The review process helped us frame issues within our proposal that we otherwise would have missed. The biggest modifications we made based off the feedback we got was to introduce concepts (algorithms, molecular bioengineering techniques, etc.) and the organization that proposed the challenge we are working on more thoroughly. Other comments that were particularly helpful to us were suggestions for rearranging where we introduced content in the proposal to better fit the intended goals of the specific aims, significance, innovation, and the research strategy. Overall, we found the feedback to be consistently useful in helping us refine our writing.

Division of Labor

The contributions from each member for project brainstorming and writing the original/final proposal were split evenly. All members took part in the algorithmic design, but Weida Ma took charge in the research of potentially helpful ideas as well as the implementation of the program's first draft. All members contributed towards debugging the algorithm, with Zhonghao Dai taking charge in this aspect of the project. Finally, all members were a part of the scoring process as well, but Vincent Chang took on the responsibility of automating the scoring process. The work involved in creating the final presentation slides and writing the final report was evenly split between the members. Overall, the collaboration between this project was effective, and it was a positive experience for all members of the group.

References

1. McKenna, Aaron, et al. "Whole-organism lineage tracing by combinatorial and cumulative genome editing." *Science* 353.6298 (2016): aaf7907.
2. Salvador-Martínez, Irepan, et al. "Is it possible to reconstruct an accurate cell lineage using CRISPR recorders?." *Elife* 8 (2019): e40292.
3. Allen Institute Cell Lineage Reconstruction Challenge Homepage Link:(https://www.synapse.org/#!Synapse:syn20692755/wiki/595096)
4. McKenna, Aaron, and James A. Gagnon. "Recording development with single cell dynamic lineage tracing." *Development* 146.12 (2019): dev169730.
5. Li, Shuying, Dennis K. Pearl, and Hani Doss. "Phylogenetic tree construction using Markov chain Monte Carlo." *Journal of the American Statistical Association* 95.450 (2000): 493-508.
6. Likic, Vladimir. "The Needleman-Wunsch algorithm for sequence alignment." *Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne* (2008): 1-46.
7. Metropolis, Nicholas, et al. "Equation of state calculations by fast computing machines." *The journal of chemical physics* 21.6 (1953): 1087-1092.