LEARNING THE "GAME" OF LIFE: RECONSTRUCTION OF CELL LINEAGES THROUGH CRISPR-INDUCED DNA MUTATIONS

JIAXIAO CAI, DA KUANG, ARUN KIRUBARAJAN, MUKUND VENKATESWARAN,
{MIAJXCAI, KUANGDA, KIRUBA, MUKUNDV}*@SEAS.UPENN.EDU

ABSTRACT. Multi-cellular organisms are composed of many billions of individuals cells that exist by the mutation of a single stem cell. Using CRISPR-based molecular-tools, it is possible to induce DNA mutations, which allows us to study the mutation lineages of various multi-ceulluar organisms. However, to date no lineage reconstruction algorithms have been examined for their performance/robustness across various molecular tools, datasets and sizes of lineage trees. In addition, it is unclear whether classical machine learning algorithms, deep neural networks or some combination of the two are the best approach for this task. In this paper, we introduce 1) a simulation framework for the zygote development process to achieve a dataset size required by deep learning models, and 2) various supervised and unsupervised approaches for cell mutation tree reconstruction. In particular, we show how deep generative models (Autoencoders and Variational Autoencoders), unsupervised clustering methods (K-means), and classical tree reconstruction algorithms (UPGMA) can all be used to trace cell mutation lineages.

1. INTRODUCTION

Multi-cellular organisms are composed of billions or trillions of different interconnected cells that derive from a single cell through repeated rounds of cell division — knowing the cell lineage that produces a fully developed organism from a single cell provides the framework for understanding when, where, and how cell fate decisions are made. The recent advent of new CRISPR-based molecular tools synthetically induced DNA mutations and made it possible to solve lineage of complex model organisms at single-cell resolution.[1] Starting in early embryogenesis, CRISPR-induced mutations occur stochastically at introduced target sites, and these mutations are stably inherited by the offspring of these cells and immune to further change. At the end of development, only the recording array sequence has to be read rather than the whole genome; the accumulated mutations can then be used as phylogenetic characters allowing the reconstruction of a tree of relationships between all cells. The objective of the project is to simulate the cell developing process and reconstruct the lineage tree of a cell, given the possible ending states. To be more specific, a cell, start from the unmutated ground state, develops into 128 cells by seven divisions in silico. Similar with the late developmental stage of *Drosophila* embryo, there is a one-hour interval between divisions, and some irreversible and inherent mutations happen within the intermission.

Computational complexity of representing each state poses a problem since each entry of the target can take on 30 distinct values, in addition to the empty token for a total of 31^{200} possibilities. To solve this problem, a Variational Autoencoder was trained to learn a meaningful latent representation for developed cells in $\mathbb{R}^{16}$, given the task of reconstruction. Once obtaining representations of the cells, different tree construction strategies were explored in the latent space (which we expand upon in Section 4):

- (1) K-means (Equal Parition) on Latent Representations
- (2) Pair-wise distance and unweighted pair group method with arithmetic mean **(UPGMA)**.

1.1. Contributions.

In this paper, we introduce a zygote developing process simulation and demonstrate its encoding into a lineage tree (a vector with 200 mutable targets represents the each cell). Next, we introduce a Variational Auto-encoder architecture for encoding CIRSPR targets, that is trained to learn a meaningful latent representation for developed cells. Finally, several strategies were explored for tree reconstruction, and we hypothesize different criteria for the evaluation of tree reconstruction.

2. BACKGROUND

2.1. Phylogenetic of Species vs. Lineage of Cells.

Classic phylogenetics typically analyzes the evolutionary relationships among a small number of species by observing their genome sequence. Conversely, a lineage is a cellular level temporal series of development connected by a continuous line of descent from ancestor to descendant. It tracks hundreds of cells and their corresponding arrays of nucleic acid sequence. Cellular sequencing data gives detailed

genetic information but introduces more challenges to analysis at the same time. The difficulties are from the random process during cell division, namely mutation and deletion. Mutations cause variations in genetic information, while deletions lead to dropouts of nucleic acids. These differences make it challenging to apply phylogenetic methods to lineage-tracing data directly. To solve the problem, a new representation of cells can be learned while fitting the stochastic process.

2.2. **Classical Reconstruction Algorithms: UPGMA.** Unweighted Pair Group Method with Arithmetic mean (UPGMA) [3] is one of the most widely used distance based tree reconstruction algorithms. Let ν as the set of unprocessed nodes of the tree and distance matrix D represents the shortest path lengths between any two nodes. Suppose i and j are two nodes, then D_{ij} is their distance. In our project, the distance can be calculated in the latent space of auto-encoder. To construct a rooted tree, UPGMA finds the nearest nodes/clusters i and j and creates a central node as the new connection. Then updates the distance matrix D and iterates over the above steps until the tree is completely resolved and all branch lengths are known.

2.3. **Deep Generative Model: Variational Auto-encoder (VAE).** In just few years, VAE has emerged as one of the most popular approaches to unsupervised learning of complicated distributions and have already shown promise in bioinformatics. VAE modeled the distribution of high-dimensional original data $P(x)$, by a set of latent variables z. Its objective was to find the optimal z in terms of capturing the intrinsic information of the input data by designing another common distribution family $Q(z|x)$ to approximate $P(z|x)$. The minimization of the Kullback–Leibler (KL) divergence between the two distributions is usually used for the approximation. In our project, the variational distribution $Q(z|x)$ should have enough representation capacity to model the complex information of $P(z|x)$ in the dataset.

2.4. **Unsupervised Machine Learning Clustering: K-means.** K-means is a clustering method that adjusts the classification of the observations into clusters and updates the cluster centroids until the position of the centroids is stable over successive iterations. Sci-Kit Learn's implementation of K-means was used in the project, and the stability of the centroids is determined by comparing the absolute value of the change in the average Euclidean distance between the observations and their corresponding centroids against a threshold (1e-05). Suppose VAE constructs a good representation, then intuition tells us that cells would gather into crowds in the latent space. Moreover, within each cluster, cells should be relatives in the lineage tree.

3. Related Work

3.1. **Allen Institute DREAM Challenge.** The new CRISPR-based molecular tools from [1] introduce a simple and elegant mutation mechanism into the particular location of a cell's DNA. It not only adds irreversible and inherent information between off-springs but also decreases the sequencing cost in experiments. Now it is possible to construct a lineage tree by only detect the sequence at particular locations instead of the whole genome. To date, however, no lineage reconstruction algorithms have been rigorously examined for their performance/robustness across various molecular tools, datasets, and some cells/size of lineage trees. It also remains unclear whether new Machine-Learning algorithms that go beyond the classical ones developed for reconstructing phylogenetic trees (such as Neighbor-Joining or Parsimony methods) could consistently reconstruct cell lineages to a high degree of accuracy. To mobilize for evaluating new optimal tree-building methods, Allen Institute started a current ongoing DREAM challenge about lineage tree reconstruction. Inspired by this challenge, the idea of our project is to use a combination of deep learning models and classical machine learning algorithms to solve the linage tree reconstruction task. However, such models excel with large datasets, and the data provided by the challenge is not sufficient for training a deep learning model. Encouraged by Professor Pratik Chaudhari, we decided to generate our own data based on an approach given by a paper written by Dr. Salvador-Martinez, as well as develop our own tree construction algorithm.

3.2. **Variational Auto-encoder in Latent Representation.** In 2018, Yosef Lab introduced single-cell variational inference (scVI) for the probabilistic representation and analysis of gene expression in single cells [2]. It uses a Variational Auto-encoder to aggregate information across similar cells and genes and to approximate the distributions that underlie observed expression values while accounting for batch effects and limited sensitivity.

4. Approach

4.1. **Simulation.** In this paper, we introduce a simulation framework for the zygote development process to achieve a dataset size required by deep learning models In the simulation, every cell is represented as a vector of 200 characters,

each character representing one Cas9 target. The simulation begins with one cell, the fertilized egg, that has all its targets in an unmutated state. The unmutated ground state is noted as "0".

Mutation events are implemented as following a Poisson distribution. Under the Poisson model, given a mutation rate $\mu_t = 0.0014$ (per unit time), the probability that a site remains unmutated after t minutes is:$e^{-\mu_t t}$. Each unmutated target can mutate to one of 30 possible mutated states with probabilities sampled from a random gamma distribution (shape parameter $k = 0.1$ and scale parameter γ = 2). Once a target is mutated, it can no longer change, either to revert to the unmutated state or to transit to a new state. The mutation process is shown in Figure 1.(a).

Besides mutation, inter-target deletions are possible to happen during the development. In experiment, where 2 or more relatively close mutation will introduce double strands break and lead to a lost of DNA between these breaks. This events have been implemented in the simulation. If 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 our dataset, one simulation records the process that a zygote cell develops into 128 cells by seven divisions with a one-hour interval in-between. Such simulation were processed 3000 times so that we have 2500 lineage trees for training, 200 for validation and 300 for testing. The frequency for each variation has been counted in Table 2.

4.2. **Representation.** For each developed cell, a vector of 200 characters tracks all the mutations during the divisions. Given the ground state as 0, 30 possible mutations are coded as A to Z and a to z. Moreover, '-' represents a deleted target. Representing variations as unique discrete characters leads to data sparsity and difficulties for training a neural network. Therefore, "mutations embedding" is used to map variations into real number as Table 1 in appendix. Since probabilities of each mutation are obey a gamma distribution, a larger real numbers in this table are used to represent less likely mutations. Based on mutation embedding, cells from one lineage tree are projected in 3D-PCA space and 2D-t-SNE space for visualization in Figure 1. In (a) and (c), cells are colored by their lineage after three divisions. In (b), cells are colored by lineage after two divisions in (b). From (b) and (c), one can observe that graphical-based clustering has its limitation to classify the cell's lineage. For example, Cluster 3 spreads at the two sides of the plot in (b) even if they both belong to the sub-cluster 6 in (c). The 3D-PCA plot supports the observation since cells become mixed after three divisions. It is because cells' diverseness decrease with the increase of division, and deletion introduces uncertainty by hiding variations.

4.3. **Autoencoder Model.** After experimenting with an autoencoder and a variational autoencoder model, we decided to continue with the regular autoencoder model as enforcing two constraints on the latent space with our loss function turned out to be difficult to train. So, in order to constrain our latent encodings to represent pairwise distances between cells, we experimented with the loss function used to train the autoencoder model. Ultimately, we implemented a mean squared error term between the latent representations of two drawn examples in addition to a mean-squared error reconstruction term. Our loss function to minimize for a given set of n cells was then the following:

$$L = \frac{1}{n}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}(||y_i - x_i||_2^2 + ||y_j - x_j||_2^2) + (||z_i - z_j||_2 - p_{ij})^2$$

Where x_i represents an input datum, z_i represents the latent representation of x_i, and y_i represents the generated representation by the decoder from z_i. Intuitively, this task is representative of encoding pairwise distances between cells in a tree. This is significant information to encode since it would allow us to construct a pairwise distance matrix between all cells in the set for use in tree reconstruction methods.

Our autoencoder model was comprised of three fully connected encoder layers and three symmetric fully connected decoder layers. The first layer takes the 200 dimensional input into a size 128 vector with a ReLU activation. The next layer takes this 128 dimensional vector into a size 64 vector with tanh activation. The final encoding layer creates a size 16 latent representation of our cell. As previously stated, the decoder is symmetric to the encoder and takes this latent representation to a generated positive real-valued vector of size 200.

In order to train the model, we used SGD with mini-batches of size 100. Each update, we would draw a tree at random and randomly select 100 pairs of cells from the tree to train the above loss function.

4.4. **Reconstruction.** We propose two different methods of reconstructing the lineage tree. The first is using an Autoencoder to generate latent representations for each cell, and running the same sized 2-means clustering algorithm on the representations to partition our cell space into two equally sized centroids. Then, we separate the two partitions and recurse on each 2-means cluster until we yield leaves (single datum). Each iteration of this will take us a further depth in the tree, with the mean of the centroid being representative of the ancestors of the leaves in that level of the tree. Since K-means takes $O(n^2)$ time to compute two centroids, we yield the recurrence $T(n) = 2T(\frac{n}{2}) + O(n^2)$. By

the master theorem, this algorithm runs in $O(n^2)$ time (which is comparable to most tree reconstruction algorithms). In our second method of tree reconstruction, a pairwise distance matrix is calculated directly on counting the different targets between any two mutation embedding and a lineage tree is reconstructed by UPGMA. Figure 2(a) represents the reconstructed tree while (b) is the real lineage tree.

4.5. **Evaluation.** In a rooted tree, a clade is a group of leaves that have a common ancestor that is not a common ancestor for any other leaf in the tree. Percentage of clade similarity, in other words, the number of branches in one tree that is present in another), is a measurement of the topological distance between two trees [4]. In our project, the percentage of clade similarity is calculated by an R package: Analyses of Phylogenetics and Evolution (APE). For instance, regarding to the lineage trees in Figure 2, there are 127 clades in each tree, and 57 clades in the reference tree are not identical to the reconstructed tree. Therefore the percentage of clades similarity is $\frac{127-57}{127} = 62.99\%$ for the reconstruction.

5. Experimental Results

1. Auto-encoder results and talk about PCA of the latent space yielding decent results of splitting the different parts of the tree. Put *good* plot of PCA in the appendix section and reference it here.

During the simulation, to follow the DREAM challenge setup, we set the length of vector as 200 representing for mutable CRISPER targets. To

Paragraph about simulation data, how more deletions appeared than expected and how this messed up our reconstruction and edit-distance methods. Calculate the frequency of deletions in the dataset.

3. Paragraph on K-means reconstruction accuracies, why K-means might not be a good suit for this problem (because we don't expect our data of tree partitions to be organized in n-spheres which would be ideal for K-means).

4. Paragraph on UPGMA reconstruction accuracies from latent representations and evaluation of pairwise distance methods.

6. Discussion

1. We tried to encode absolute distances, when we are dealing with a contextual problem. Why this might not have been the best approach and recommendations for contextual models such as a modified beam search.

2-1. The large dimensionality of the dataset: there are 200 possible mutation targets for each cell. We choose the length based on the dataset of the DEAM challenge. But to solve the lineage tree problem, the length of mutations targets is actually a hyper parameter. Increase the number of targets,....; decrease the number of targets, ...

2-2. Talk about the large dimensionality of the vectors and the rarity of some mutations making it difficult to model the cell DNA properly. The state space of cell DNAs is also far too large to model as one-hot encodings as we would word embeddings in NLP. This caused us to drift to the autoencoder model to reduce dimensionality.

3. Recommendations for some kind of batch normalization or normalization of distances as these may have been hard for the autoencoder to properly model in the latent space.

REFERENCES

[1] I. Salvador-Martınez, M. Grillo, M. Averof, and M. J. Telford, "Is it possible to reconstruct an accurate cell lineage using CRISPR recorders?," Developmental Biology, p. 23.

[2] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, "Deep generative modeling for single-cell transcriptomics," Nat Methods, vol. 15, no. 12, pp. 1053–1058, Dec. 2018.

[3] C. D. Michener and R. R. Sokal. A quantitative approach to a problem in classification. Evolution, vol.11, pp.130-162, 1957.

[4] Jaime Huerta-Cepas, Francois Serra and Peer Bork. ETE 3: Reconstruction, analysis and visualization of phylogenomic data. Mol Biol Evol 2016; doi: 10.1093/molbev/msw046

7. APPENDICES

TABLE 1. Mutation Embedding Table

0	-	A	B	C	...	Z	a	b	c	d
0	1	2	3	4	...	27	28	29	30	31

TABLE 2. Variation Frequency in Simulated Data

Variation	0	-	A	B	C	D	E	F	G	...
Freq	0.352	0.535	0.0329	0.023	0.0148	0.0171	0.0069	0.004206	0.003204	...

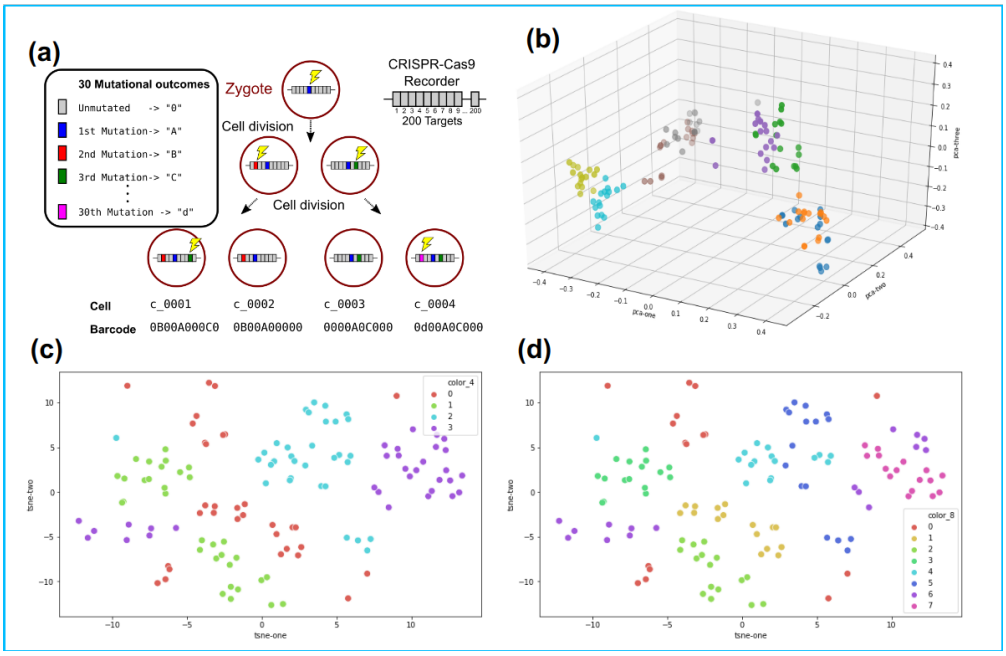

FIGURE 1. Visualization of Mutation Embedding

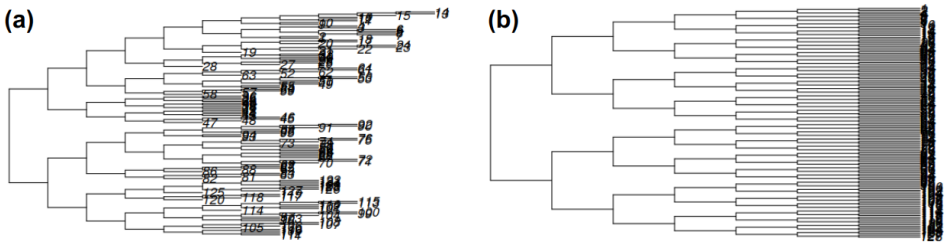

FIGURE 2. UPGMA Based on Mutation Embedding