The Alexander Kofkin Faculty of Engineering
Bar-Ilan University

PREDICTION OF FUTURE GENE EXPRESSION IN CELLS USING COMPUTATIONAL MODELLING

Submitted by

Dotan Sadka
(ID: 318474657)

Adi Shrem
(ID: 208279570)

Under the supervision of

Dr. Shahar Alon
Tal Goldberg

Table of Contents

1 Abstract 4

2 Acknowledgment 5

3 Introduction 6

4 Research Goals 8

5 Theoretical Background 9

5.1 Tumor-Immune Cell-Cell Interactions 9

5.2 T-Cells 9

5.3 In Situ Spatial Genomics 10

5.4 RNA Velocity 10

5.5 Dimensionality Reduction 11

5.5.1 PCA 11

5.5.2 UMAP 11

5.6 Hyper Parameters 12

5.7 Linear Regression 12

5.8 KNN Algorithm 13

5.9 Precision, Recall and F1-Score 13

5.10 Permutation Test 14

6 Methods 16

6.1 Preprocessing 16

6.2 Hyperparameters tuning 18

6.2.1 Impact of Quantile Size and Number of Neighbors (K) on Gamma Calculation 19

6.2.2 Filtering Genes for Reliable Gamma Values 19

6.2.3 Identifying the Time Parameter (t) for Predicting Future Expression States 20

6.3 Hypothesis Testing: Association Between T Cell Proximity to Tumor Cells and Their Future States 20

7 Results 21

7.1 RNA velocity implementation 21

7.1.1 Hyperparameter Results 22

7.2 Proximity to Tumor Cells and Permutation Test 27

8 Supplementary Figures 29
9 Block diagram 32
10 Summary and Conclusions 34
11 Future Directions 36
12 References 38

1 Abstract

One of the critical challenges in modern oncology is improving the effectiveness of immunotherapy, a treatment that has revolutionized cancer care by utilizing the immune system to target tumors. Despite its promise, the success of immunotherapy remains limited, with only a fraction of patients, typically between 20% and 40% responding positively, and even fewer achieving long-term remission[1,2]. These variations in efficacy are partly due to the dependency of the treatment on the precise interaction between immune and tumor cells. Surprisingly, current clinical practices often administer immunotherapy without verifying the presence or activity of immune cells in the tumor microenvironment. Even when immune cells are present, their interaction with cancer cells is not guaranteed, which can result in ineffective treatment outcomes.

The recent advent of spatial genomics (like Expansion Sequencing -ExSeq) provides a new lens to explore the T-cells proximity to tumor. By mapping the spatial distribution of immune and tumor cells alongside their gene expression profiles, researchers can gain a more nuanced understanding of how immune cells behave when they are in proximity to cancer cells. This insight allows for the investigation of whether immune cells undergo gene expression changes when near tumor cells, a key indicator of potential interaction.

In this study, we employed a computational algorithm called RNA velocity designed to predict immune-tumor cell interactions based on gene expression data. This tool combined with Expansion Sequencing not only identifies interactions but also holds the potential to guide future therapeutic decisions. By enabling more precise selection of patients who are likely to benefit from immunotherapy, our approach could significantly improve the treatment's success rate. Furthermore, this method lays the groundwork for future research into the diverse immune responses across different cancer types, potentially leading to more tailored and effective treatment strategies.

2 Acknowledgment

We would like to extend our deepest gratitude to several individuals whose support and contributions played a pivotal role in the successful completion of this project.

We are sincerely thankful to our supervisors, Dr. Shahar Alon and Tal Goldberg, for their unwavering guidance and expert insights. Their extensive knowledge and continuous encouragement were vital throughout every stage of this project. Their mentorship not only provided us with direction but also greatly influenced the progress, development, and refinement of our work.

Their thoughtful feedback and dedicated involvement allowed us to overcome challenges, stay focused on our objectives, and significantly improve the overall quality of the project.

3 Introduction

The goal of our project is to utilize RNA velocity, a computational method for modelling gene expression dynamics, combined with Expansion Sequencing for detailed measurement in situ, to explore how immune cells, such as T cells, interact with cancer cells in a tumor microenvironment. RNA velocity predicts the future state of gene expression in individual cells by analyzing the ratio between spliced and unspliced RNA, providing insight into cellular trajectories.

This project expands upon the foundational work done last year on predicting future gene expression in cells using RNA velocity. Previously, the focus was on analyzing a single tissue sample, from a breast cancer patient. Last year's project team encountered limitations with the La Manno technique and the Velocyto[3] Python package, which restricted the flexibility needed to adapt to different dataset types. The pre-built algorithm used by Velocyto lacked the ability to modify certain hyperparameters that the team believed could influence the results. Given that these parameters are crucial for adapting the model to different datasets and optimizing the analysis, the team decided to develop their own code.

Building on the progress of that project, this year we scaled up the analysis significantly. We applied the refined and updated code to five different tissue samples, increasing the dataset to 3.5 times the size of last year's. This substantial increase allows for a more comprehensive exploration of how immune cells interact with cancer cells across different tissues. By broadening the scope of our dataset, we aim to improve the robustness of our findings and enhance the generalizability of our model to different biological contexts.

The project aims to address the snapshot problem inherent in single biopsy studies. Traditional biopsy analysis captures a static moment in time, limiting our understanding of the dynamic changes happening within tissues. RNA velocity, by incorporating unspliced and spliced RNA data, (the data is the result of the process called Expansion Sequencing, a technique conducted in Shahar Alon's lab that capture high resolution molecular data), this provides an element of temporal prediction, allowing us to infer future gene expression states. This temporal aspect is critical, as it gives us insights into how cellular interactions may evolve, rather than just offering a single snapshot of gene expression at one moment. By doing so, we hope to better understand how physical proximity between T cells and tumor cells influences changes in gene expression within the immune cells. The goal remains the same: to gain insights into the mechanisms of immune-tumor cell interactions and to potentially improve the predictive power of RNA velocity in various tissue environments. Through this work, we aim to continue pushing the boundaries of computational biology in cancer research, providing a tool that

could have significant implications for understanding cellular behavior in complex tissue structures.

4 Research Goals

1) **Identifying Cell Interactions using Expansion Sequencing and RNA Velocity:** We aim to use RNA velocity to detect interactions between cells, such as between immune cells and cancer cells, by linking RNA velocity to the physical locations of cells in the tissue measured using Expansion Sequencing.

2) **Expanding and Optimizing Analysis for Broader Applications**: We would like to enhance and adapt the current code for use across different tissues and larger datasets to find improved statistical significance.

5 Theoretical Background

5.1 Tumor-Immune Cell-Cell Interactions

Tumor-immune cell-cell interactions are a critical aspect of cancer biology and play a pivotal role in the effectiveness of immunotherapy. The immune system is designed to detect and destroy abnormal cells, including cancer cells, but tumors often develop mechanisms to evade immune detection, enabling them to grow and spread. Several types of immune cells, such as T cells, dendritic cells, macrophages, and natural killer (NK) cells, are involved in these interactions. Cytotoxic T cells in particular are capable of directly attacking tumor cells, while dendritic cells present tumor antigens to T cells, initiating an immune response. However, tumors frequently evade this process through strategies like expressing immune checkpoint molecules, such as PD-L1, which bind to T cell receptors and inhibit their activity or creating an immunosuppressive microenvironment by recruiting regulatory T cells or releasing cytokines that suppress immune activity [4]. Interactions between immune and tumor cells through direct physical contact are likely to result in changes at the transcriptomic level. By measuring these molecular shifts that occur when immune cells are near tumor cells, we can gain a deeper understanding of the mechanisms that enable tumors to evade immune detection and manipulate their microenvironment.

5.2 T-Cells

T cells are key players in the adaptive immune system, responsible for coordinating and executing attacks against foreign invaders in the body. These cells are broadly classified into effector T cells, memory T cells, and regulatory T cells (Tregs), each playing distinct roles. Effector T cells, including CD8+ and CD4+ T cells, are critical in mounting an immune response, while Tregs help maintain immune balance by preventing excessive responses. In the tumor microenvironment, T cells importance is second only to macrophages in their prevalence. One of the most important tumor-fighting cells is the CD8+ T cell, also known as the "killer T cell" [5]. Upon recognizing tumor antigens, CD8+ T cells mature into cytotoxic T lymphocytes that release cytotoxic granules like perforin and granzyme, which induce apoptosis in tumor cells by disrupting their membranes and internal proteins. CD4+ T cells, also known as "helper T cells", assist in regulating this antitumor response by producing proinflammatory cytokines that further enhance cytotoxic activity. However, tumor cells have developed mechanisms to evade immune detection by altering T cell functions, including reducing their immunogenicity to avoid being targeted by these killer cells. Despite their potency, the manipulation of T cell responses by tumors is a significant challenge in immunotherapy.

5.3 In Situ Spatial Genomics

In situ spatial genomics enables the study of gene expression within its original spatial context by preserving the spatial information of the tissue sample, unlike single-cell transcriptomics, which loses this spatial organization.

Existing spatial transcriptomics methods face trade-offs between spatial and molecular resolution. Technologies like MERFISH, Spatial Transcriptomics (ST), and antibody-based methods (e.g., multiplexed ion beam imaging) are limited in either resolution, the number of genes they can measure, or in accurately assigning genes to individual cells.

Expansion Sequencing (ExSeq)[6], developed by the lab, offers high spatial and molecular resolution by physically expanding tissues for in situ RNA sequencing with super-resolution. This allows precise detection of gene expression in complex environments like tumor biopsies.

ExSeq overcomes limitations found in other technologies, enabling detailed in situ measurements of cell types and cell states in clinical tumor samples, capturing fine-scale differences between cells.

A challenge in genomic research on tumors is the limited availability of fresh samples. ExSeq was initially incompatible with formalin-fixed, paraffin-embedded (FFPE) tissue samples, hindering its use for retrospective studies on archived tissues. However, a protocol for ExSeq of FFPE biopsies was constructed enabling usage of the method.

5.4 RNA Velocity

RNA velocity, a new analysis method recently introduced, estimates gene production versus degradation rates, essentially predicting the future state ('velocity') of genes based on a single time point[7]. By analyzing the ratio of unspliced to spliced RNA for each gene in individual cells, RNA velocity infers a steady state ratio ('gamma'-γ) for gene expression. Cells with high unspliced RNA ratios compared to the steady state are considered 'induced' or positively evolving, while those with low ratios are 'repressed' or negatively evolving. This information provides insights into the future state of cells, facilitating the distinction between transient and prolonged interactions between T cells and tumor cells within the tumor microenvironment.

The equation used for RNA velocity calculation is $v = u - \gamma s$.

u-unspliced, s-spliced, γ-steady state ratio.

The idea behind calculating genes velocities is to find the future gene expression in time 't'. We use its initial gene expression ($s_{t=0}$) plus his velocity (rate of change) calculated earlier multiplied by 't', $s(t) = s_0 + vt$.

s(t) value is projected into PCA linear space and compared to s_0 location in PCA space.

Choosing PCA (Principal Component Analysis) for projecting RNA velocity values is a good solution because it provides a linear and interpretable way to reduce the dimensionality of the data. PCA captures the most significant variance in the gene expression data, allowing for a clearer visualization of cell trajectories in a simplified space. Unlike non-linear methods like t-SNE or UMAP, PCA retains the spatial relationships between cells, making it easier to track how gene expression changes over time. This linear reduction also aligns with the RNA velocity model, which calculates gene expression changes linearly. Additionally, PCA's simplicity and speed make it well-suited for large datasets like ours.

5.5 Dimensionality Reduction

Dimensionality reduction is a key technique in data analysis and machine learning, used to address the challenges of high-dimensional datasets. As data grows, the number of features or variables can increase, which complicates computation, lowers model performance, and makes data visualization and interpretation more difficult. Dimensionality reduction techniques address these challenges by selecting or transforming features to retain essential information while eliminating redundant or irrelevant ones. This reduction in dimensionality enhances model efficiency, reduces the risk of overfitting, and improves data interpretability.

5.5.1 PCA

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique that transforms a large set of variables into a smaller set while retaining most of the original information. It achieves this by applying an orthogonal transformation that converts correlated variables into uncorrelated ones, known as principal components, which capture the maximum variance in the data. These principal components are linear combinations of the original variables and are ranked in order of decreasing importance. While reducing the number of variables may slightly compromise accuracy, the key to effective dimensionality reduction is finding a balance between accuracy and simplicity[8].

5.5.2 UMAP

UMAP aims to preserve the structure of the data as it reduces the dimensionality, meaning it seeks to maintain the relationships between data points in the high-dimensional space when mapping them to a lower-dimensional space.
UMAP first constructs a graph representing the data's high-dimensional structure. It does this

by estimating the local density of data points around each point to define a neighborhood. The idea is to create a "fuzzy" topological representation of the data, where each data point is connected to its neighbors based on their proximity.
UMAP then attempts to find a low-dimensional representation that preserves this fuzzy topological structure as closely as possible. It does this by optimizing a cost function that balances the preservation of both local and global structures. The algorithm searches for an embedding (a mapping from high-dimensional space to low-dimensional space) where points that are close in the high-dimensional space remain close in the low-dimensional space, while points that are far apart stay distant[9]. .שגיאה! מקור ההפניה לא נמצא

5.6 Hyper Parameters

Hyperparameters are settings or configurations that are chosen before the algorithm runs and can significantly influence its behavior and performance. These parameters are not learned from the data but are set by the user based on experience, experimentation, or domain knowledge.

Hyperparameters can affect the speed, memory usage, and overall effectiveness and accuracy of the algorithm, and choosing the right ones often requires careful consideration and testing.

5.7 Linear Regression

Linear regression is a statistical method used to predict the value of one variable based on the value of another[10]. The variable being predicted is called the dependent variable, while the variable used for prediction is known as the independent variable. The goal of linear regression is to model the relationship between these two variables by fitting a straight line, which minimizes the difference between the predicted and actual values. This line is determined by a linear equation of the form $Y = aX + b$ where Y is the dependent variable, X is the independent variable, b is the intercept and a is the slope (which represents the change in Y for a one-unit change in X).

A key feature of linear regression is its simplicity, offering an easy-to-interpret mathematical formula that generates predictions. The method uses a "least squares" approach to find the best-fit line for a set of data points. Essentially, it estimates the coefficients of the linear equation that best predicts the dependent variable based on one or more independent variables.

Before applying linear regression, it's important to check whether a relationship exists between the variables[11]. This relationship is not necessarily causal but can be an association where changes in the independent variable correspond with changes in the dependent variable. A scatterplot is often used to visualize this relationship, and a correlation coefficient (a number between -1 and 1) helps quantify the strength of this association. If no significant relationship exists, linear regression may not provide meaningful predictions.

Linear regression's simplicity is both a strength and a limitation. It assumes a linear relationship between variables, which may not always hold true in complex real-world scenarios. Despite this, linear regression remains a foundational technique in predictive modelling due to its ease of use and ability to provide quick, interpretable results.

5.8 KNN Algorithm

The K-nearest neighbors (KNN) algorithm is a fundamental machine learning technique used for both classification and regression tasks. KNN operates on the principle of proximity, meaning that it predicts the label or value of a data point based on the labels or values of its closest neighbors in a dataset. The algorithm works by calculating the distance using Euclidean distance between a given point and all other points in the dataset[12]. Then, it selects the 'K' closest points and bases its prediction on the majority class (for classification) or the average value (for regression) of these neighbors. The choice of K plays a crucial role in determining the algorithm's performance, as a small K might make the model sensitive to noise, while a large K may lead to over-smoothing, losing important details.

One of the main advantages of KNN is that it is a non-parametric method, meaning it doesn't make assumptions about the underlying data distribution. This makes it versatile for a wide range of applications, especially when relationships between data points are nonlinear or complex. However, KNN can be computationally expensive, especially for large datasets, since the algorithm requires distance calculations for every point in the dataset.

5.9 Precision, Recall and F1-Score

Precision measures the accuracy of positive predictions made by the model. Specifically, it is the ratio of correctly predicted positive instances (True Positives) to the total instances that were predicted as positive (True Positives + False Positives).

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

High precision indicates that when the model predicts a positive outcome, it is usually correct. Precision is especially important in scenarios where the cost of false positives is high, such as in medical diagnoses or spam detection[13].

Recall, also known as Sensitivity or True Positive Rate, measures the model's ability to correctly identify all positive instances. It is the ratio of correctly predicted positive instances (True Positives) to all actual positive instances (True Positives + False Negatives).

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negative\ (FN)}$$

High recall means the model can identify most of the actual positive cases. This is critical in situations where missing positive cases has serious consequences, such as in disease screening[9].

F-score (F_1 score) is the harmonic mean of Precision and Recall. It provides a single metric that balances both concerns, particularly when you need to find an equilibrium between precision and recall.
The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if precision and recall are zero[9].

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

5.10 Permutation Test

A permutation test is a non-parametric method used to assess the statistical significance of an observed result by comparing it to what might occur by chance. In its basic form, a permutation test involves repeatedly rearranging (or "shuffling") the labels or data points in the dataset while keeping certain aspects constant and recalculating the test statistic for each permutation. The goal is to generate a distribution of results that represent what would happen if there were no real effect, providing a reference point against which the observed result can be compared.

The advantage of permutation tests is that they don't assume any underlying distribution of the data, making them particularly useful in complex or non-standard data scenarios. In our

analysis, we used a permutation test to assess whether the observed correlation between CD8A T-cells and tumor cell proximity was statistically significant. By shuffling cluster labels while preserving overall proximity and cluster sizes, we were able to create a null distribution and calculate how often random permutations resulted in a correlation as strong as the one observed.

6 Methods

6.1 Preprocessing

First, we created the data frames necessary for the RNA velocity calculation. We received spatial genomic data for five tissues from Shahar Alon's lab. The first task was to create two data frames for each tissue: one containing the gene expression levels for each cell in the nucleus and the other containing this information for the cytoplasm. The final data frames included only the cells that had gene expression in at least one of these states. After that, we combined all the data frames from all the tissues into two large data frames that contained only the genes expressed in all the tissues. Finally, we obtained two data frames with 4,530 cells and 271 genes.

We aimed to create a spatial indicator from these tables to assist in predicting future gene expression. Our focus was on the table containing cytoplasmic mRNA (spliced mRNA), which is crucial for RNA velocity computation.

Using the Seurat package in R, we conducted spatial data analysis to identify and cluster cells based on their gene expression profiles. Our approach involved two dimensionality reduction techniques: UMAP (Uniform Manifold Approximation and Projection) and PCA (Principal Component Analysis). UMAP was employed to visualize and group cells into distinct clusters, using genes known to act as markers for specific cell types.

Seurat helps manage technical noise in single-cell RNA-seq data by clustering cells based on their PCA scores, where each PC (Principal Component) represents a 'metafeature' that captures correlated gene expression patterns. To determine the appropriate number of PCs for the analysis, we applied Seurat's JackStraw procedure, which generates a null distribution by permuting a subset of the data. By comparing the p-value distributions of each PC to a uniform distribution, we identified seven significant PCs that were enriched with low p-values, capturing meaningful biological variation.

To fine-tune the analysis and match the results from Dr. Alon's lab, which analyzed both nuclear and cytoplasmic mRNA (while our focus was solely on cytoplasmic mRNA), we wrote a custom R script. The code was designed to explore different dimensionality and resolution combinations, with the goal of identifying the optimal parameters for clustering.

After running the code, we found that reducing the dimensions to four PCs and setting the resolution to 0.7 yielded the best clustering results. This configuration produced seven distinct clusters, each corresponding to cell types with clearly distinct gene expression profiles. The final clusters were consistent with known cell types identified in Dr. Alon's previous analysis, confirming the accuracy of our cytoplasmic-only approach.

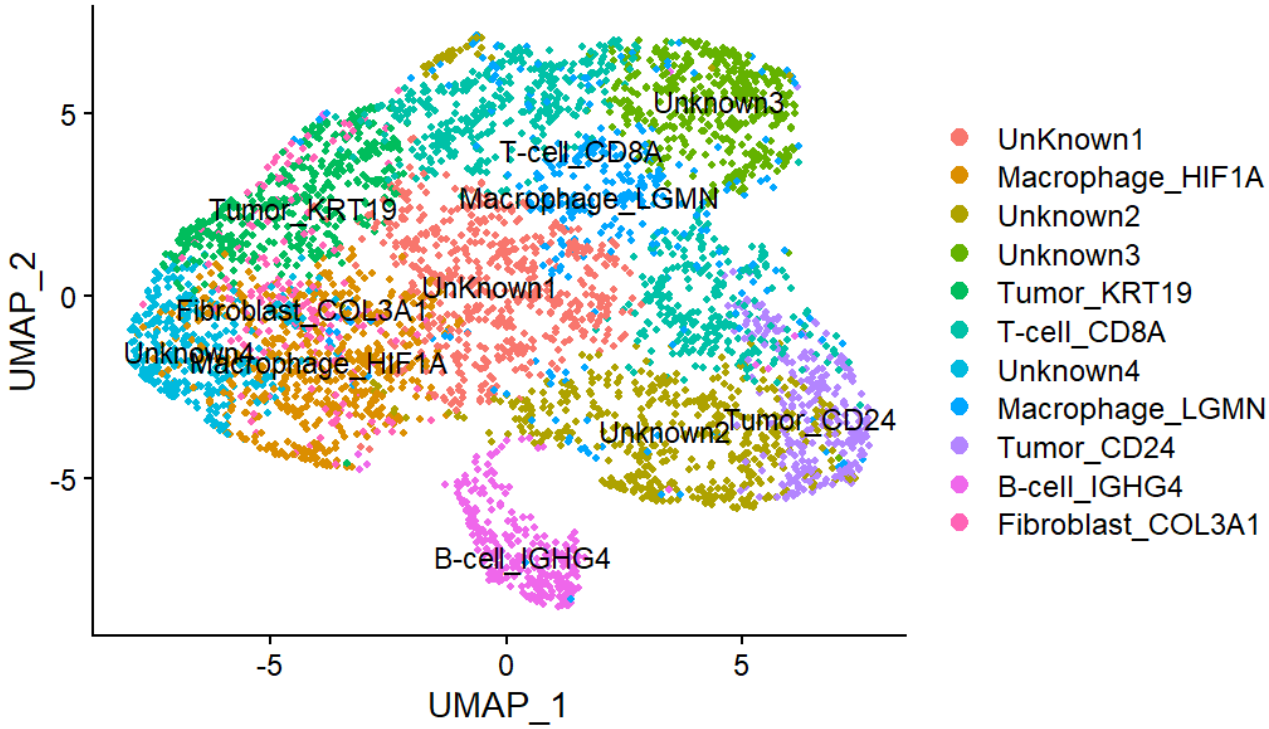


Fig 1.1: UMAP Projection of Cell Clusters with Annotated Cell Types

We also needed to represent the cells in space using PCA because of its linear properties. This approach addresses the challenge of projecting future cellular states in an embedding space, which can be complex and difficult to interpret[14]. For comparing initial gene expression to gene expression at a specific time t, a linear space greatly simplifies the analysis compared to other spatial methods used in previous experiments.

We use the table we created for cytoplasmic mRNA to generate the PCA space and initialize points in three principal components (PCs) for each cell, where $S_0(pc1, pc2, pc3)$ represents the initial condition for the final gene expression equation: $S(t) = S_0 + vt$.

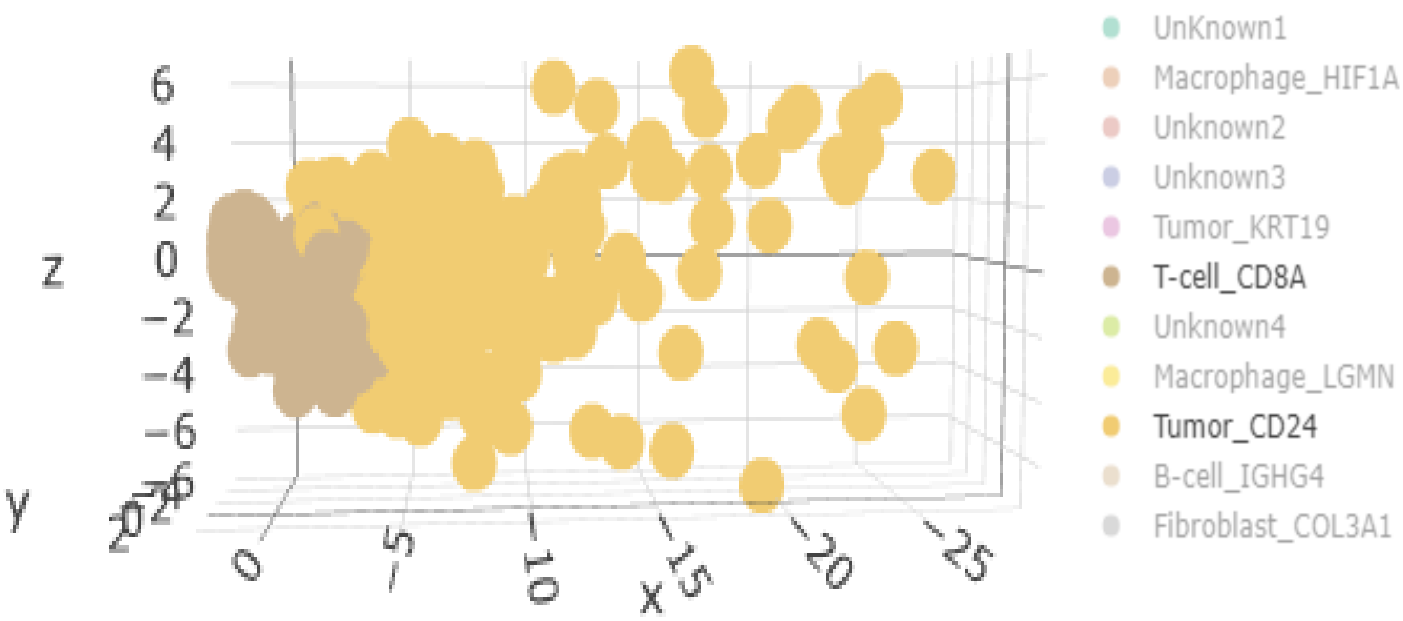


Fig 1.2: This 3D PCA plot shows the spatial distribution of cell populations, with a focus on T-cell CD8A (brown) and Tumor CD24 (yellow). The analysis captures key variance across three principal components (x, y, z axes), helping to differentiate between cell types based on their gene expression profiles. This visualization provides insight into the clustering and separation of immune and tumor cells in the dataset.

For the RNA velocity calculations, we created a data frame that contains the cell number, its type based on the cluster calculation, and its initial position in the principal components (PCs) space.

6.2 Hyperparameters tuning

After realizing that the available RNA velocity packages were not flexible enough and didn't fit the data, a new Python code was developed in Shahar Alon's lab. We are using this custom code for our project, applied to a larger dataset. For this code, it was necessary to configure several important hyperparameters.

6.2.1 Impact of Quantile Size and Number of Neighbors (K) on Gamma Calculation

In estimating the gene-specific equilibrium coefficient γ, we rely on regression using the extreme expression quantiles rather than the entire dataset. This method ensures a more accurate and reliable estimation, especially when most of the cells being observed are not in steady-state conditions. By concentrating on the most extreme values, we can focus on the critical points that best reflect the gene's transcriptional dynamics, without interference from the more variable intermediate points that might distort the results. This approach enhances the linear regression's ability to accurately capture the equilibrium coefficient. To fine-tune the gamma calculation, we conducted a sensitivity analysis by testing different quantile ranges, specifically between 5% and 10%, to identify the range that offers the most robust and accurate estimation.

Alongside selecting the quantile value, another important parameter that influences the gamma calculation is the number of neighbors to include. Each cell has its own position in PCA space, but relying on the expression of a single cell risks introducing inaccuracies, aberrations, and random noise. To address this, the position of a cell in PCA space is calculated as the average of the cell's expression values and those of its neighbors.

For each point, we calculated the Euclidean distances to all other points. To determine the optimal number of neighbors (K), we computed gamma values across a range of K values. We aim to select the smallest K at which the gamma values converge, meaning increasing K further does not significantly alter the gamma values, while still avoiding excessive averaging that could dilute the precision of the data.

6.2.2 Filtering Genes for Reliable Gamma Values

After determining the optimal K value and the required quantiles, we were able to calculate the gamma value for each gene and generate plots of gene expression, where the weighted points of the cells and the linear gamma line were displayed.

We aimed to identify an elliptical shape in the plot, which indicates a gene's expression dynamics. First, we created plots for each gene and manually filtered the genes based on their visual patterns, selecting those that exhibited the desired elliptical shape while discarding those with random dispersion that could potentially interfere with the overall calculation.

The genes that did not passed the visual inspection filter had also exhibiting very low expression levels. To refine our analysis, we initially filtered out genes lacking the expected elliptical shape. From the remaining genes, we established a ground truth group. We then employed F-score analysis to determine the expression level threshold necessary for gene inclusion in further studies.

6.2.3 Identifying the Time Parameter (t) for Predicting Future Expression States

To determine the optimal t parameter for the calculation of future gene expression, given by $S(t) = S_0 + vt$, we considered several key factors. The primary criterion was that the sum of the s(t) vector should remain equal to the sum of the initial s(0) vector, which we had normalized to 1. This ensures that only the direction of the vector changes, while its magnitude remains constant.

Upon examining this condition, we observed that for values of t up to 4, the change in the sum of s(t) remained within 1-10%. As a result, we decided to explore t values in the range of 0.2 to 4. The second crucial factor in determining the optimal t was robustness. The optimal t value should be chosen such that the resulting vectors for t values greater than this threshold exhibit approximately the same orientation.

6.3 Hypothesis Testing: Association Between T Cell Proximity to Tumor Cells and Their Future States

In our analysis, CD8A was the only T-cell subtype identified in the dataset, making it particularly relevant for understanding its influence on tumor cells. After optimizing all hyperparameters, we calculated the future states of CD8A cells based on their RNA velocity and projected these states into PCA space. We then clustered the predicted future states into distinct groups and compared these clusters to the groups derived from their proximity to tumor cells to assess any potential correlation.

To evaluate the statistical significance of the correlation, we conducted a permutation analysis. In this analysis, we preserved the overall number of CD8A cells close to tumor cells but randomly shuffled the cluster labels while keeping the cluster sizes constant. For each permutation, we recorded the number of instances where the proximity percentage exceeded the observed value for each cluster.

In the permutation analysis, we preserved the overall number of CD8A cells close to tumor cells but randomly shuffled the cluster labels while keeping the cluster sizes constant. For each permutation, we recorded how many had a proximity percentage higher than the observed value for each cluster. Notably, for Cluster 2, which had the highest observed proximity percentage, we obtained a p-value lower than 0.05, indicating statistical significance.

This result suggests a meaningful interaction between the T-cells and tumor cells, based on both their future gene expression and their spatial proximity.

7 Results

To truly understand the interactions between cells, spatial genomic maps alone are insufficient. These maps provide valuable information about cell locations and their gene expression at a single moment in time, but they fall short of capturing dynamic cellular processes. To analyze potential interactions between cells, we need to examine their gene expression over a broader time frame, beyond just a snapshot.

This is where RNA velocity comes in as a powerful solution. By utilizing a simple equation, RNA velocity allows us to predict a cell's future gene expression based on the balance between spliced and unspliced mRNA. This temporal perspective helps overcome the limitations of studying cells at a single moment.

Our hypothesis is that by combining the spatial proximity of cells with their gene expression over time, we may gain deeper insights into whether interactions between cells are taking place. Such discoveries could have far-reaching implications for time-resolved phenomena like embryogenesis, tissue regeneration[15], and even personalized treatment decisions.

More specifically, for our data derived from a biopsy of a cancer patient, we aim to distinguish between immune cells that are merely in proximity to cancer cells and those that are actively interacting with them. Identifying these interactions could be crucial in determining whether immunotherapy is a suitable treatment option for the patient, potentially improving the precision of therapeutic decisions.

7.1 RNA velocity implementation

From our initial dataset, obtained from five tissue samples of a cancer patient biopsy, we derived two data frames: one containing the genomic data of 271 genes inside the nucleus, and the other containing the data from outside the nucleus (in the cytoplasm), across a total of 4,530 cells. we clustered the cells to seven clusters by cell types. And created PCA space for three PCs for each cell for the S_0 component, which is the initial gene expression for each cell represent in the PCA space, for the equation $S(t) = S_0 + vt$.

7.1.1 Hyperparameter Results

7.1.1.1 quantiles and K

As mentioned, to calculate RNA velocity using the Python code developed in Dr. Alon's lab, we first needed to determine specific hyperparameters and adjust the code for our large dataset. We began by examining the quantiles to use for gamma calculation. Sensitivity tests were performed using quantile values between 5% and 10%, and we compared the percentage change in the gamma results between these values

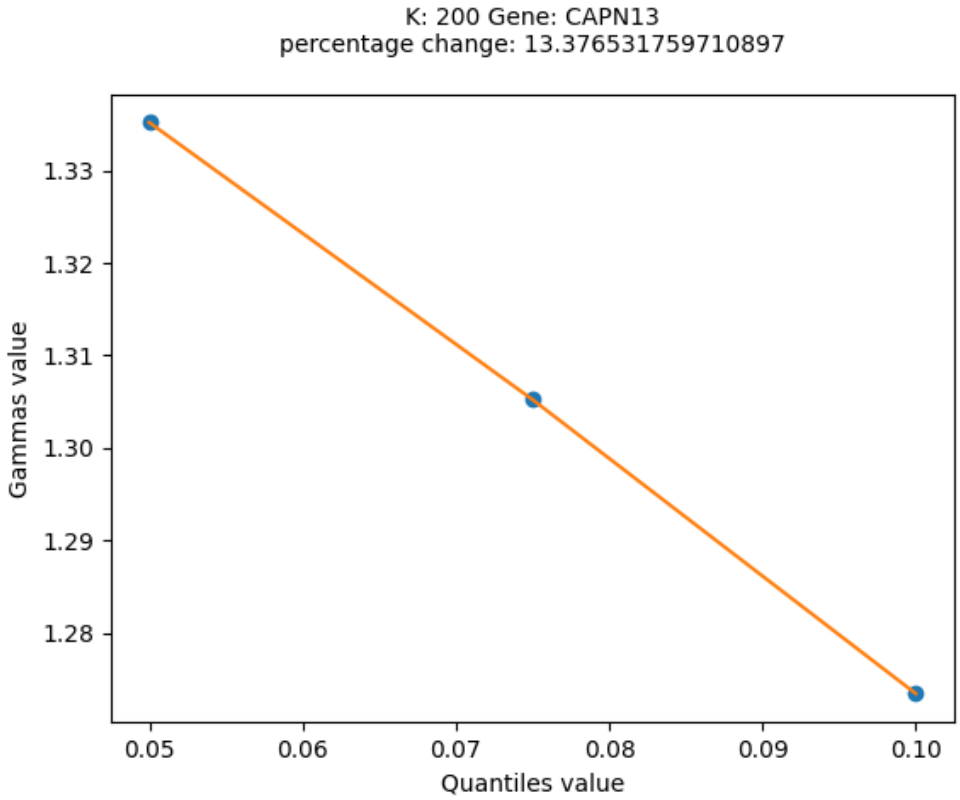


Fig 1: *Percentage Change in Gamma Values Across 5%-10% Quantiles.*

After calculating gamma for these quantile values, we observed no significant differences in the gamma results (up to 20%), so we decided to proceed with a 10% quantile value.

In parallel to determining the quantiles, we also needed to optimize the value of K, which represents the number of neighbors used to determine each cell's PCA space. To do this, we performed sensitivity tests, calculating gamma across several values of K for each gene to identify the most suitable parameter.

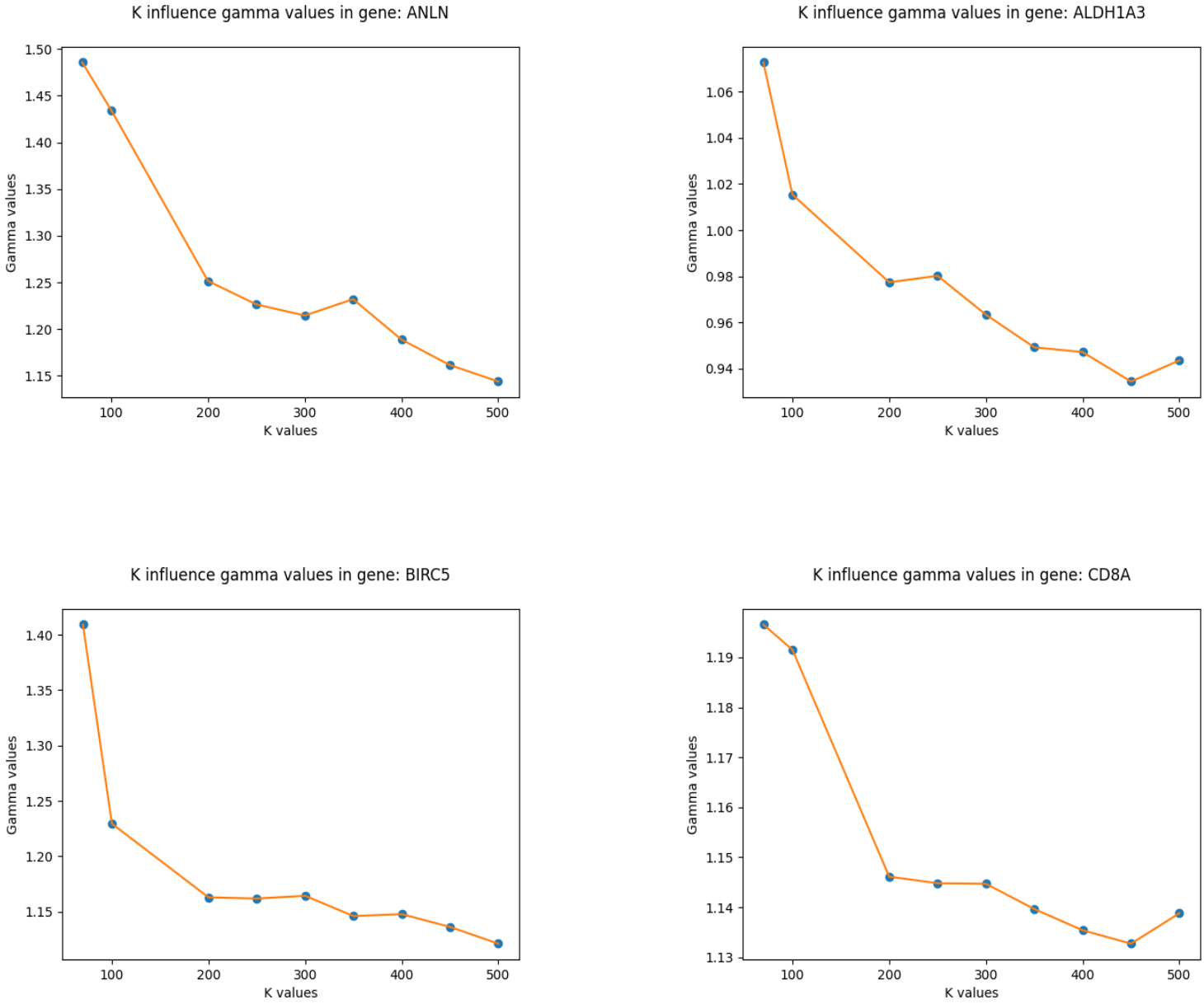


Fig 2: *Influence of K on Gamma Values*

The conclusion from the results indicated that K = 200 is the optimal number of neighbors we need.

To sum up, gamma is calculated by taking the 10% of points with the lowest expression values and the 10% of points with the highest expression values, where each point represents the average expression value of the cell and its K nearest neighbors.

7.1.1.2 Genes

As mentioned in La Manno et al., [16]to assess whether a gene is actively expressed, we look for an almond-shaped distribution when plotting gene expression. This shape is determined based on the chosen quantiles and the value of K, indicating a well-defined relationship between spliced and unspliced mRNA levels.

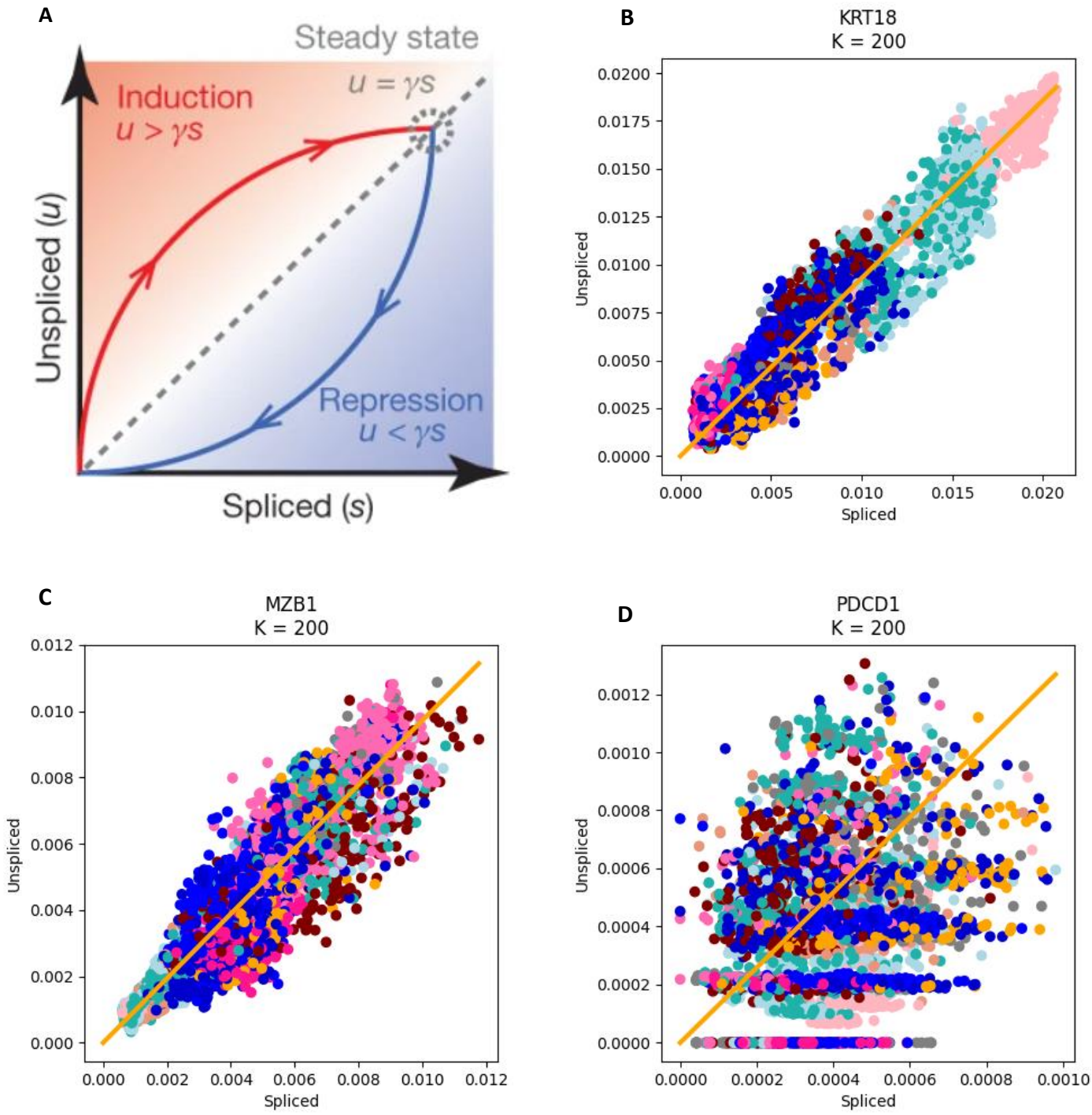


Fig 3: *Gene Filtration by Shape.* ***A:*** *Taken from La Manno et al., illustrating the balance between unspliced (u) and spliced (s) mRNAs as an indicator of cellular state progression. The graph shows two phases: induction (u > γs), where unspliced mRNA increases, and repression (u < γs), where it decreases, eventually reaching a steady state (u = γs).*
B-D: *Scatter plots representing the relationship between unspliced and spliced mRNAs for different genes.* ***B*** *and* ***C*** *(KRT18, PDCD1) show genes with the expected elongated shape, suggesting a well-defined splicing dynamic.* ***D*** *(MZB1) depicts a gene that does not follow the typical elongated shape, indicating deviation from expected splicing behavior.*

As seen in Fig 3, some genes displayed the expected almond-shaped distribution, while others did not match the desired shape. Another example of a gene's shape can be found in Figs. 9-10. The genes that deviated from this shape also showed very low expression levels. Out of 271 genes, 165 passed the visual inspection filter, which we defined as our ground truth. Next, we aimed to exclude genes whose expression levels were too low to be significant and could potentially be considered noise. To determine an appropriate cutoff for the maximum expression level, we used F-score calculations. The results indicated that the optimal expression cutoff is 0.005 in the nucleus and 0.004 in the cytoplasm.

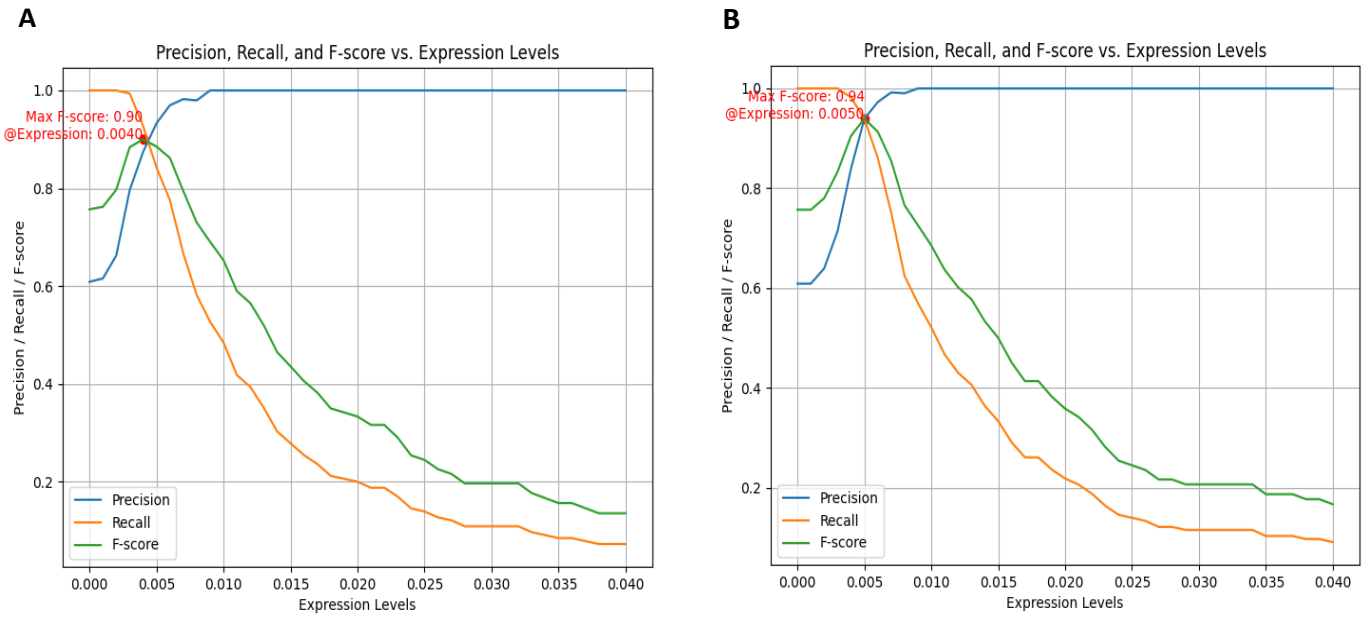


Fig 4: *Determining Expression Thresholds Based on F-score Analysis.* ***A:*** *Plot showing precision, recall, and F-score as functions of expression levels for spliced mRNAs. The optimal threshold, indicated by the maximum F-score (0.99), is found at an expression level of 0.0040.* ***B:*** *Similar analysis for unspliced mRNAs, where the highest F-score (0.94) occurs at an expression level of 0.0050.*

7.1.1.3 Time Step, t.

We computed the predicted future gene expression states for T- cell, based on their velocities, using various t values. After evaluating the results, we selected t=1.2, as we observed no significant changes beyond t=1.4, and the other results remained largely consistent. In addition, we ensured that the sum of S(1.2) does not exceed S(0), maintaining the vector's magnitude while allowing only its direction to change.

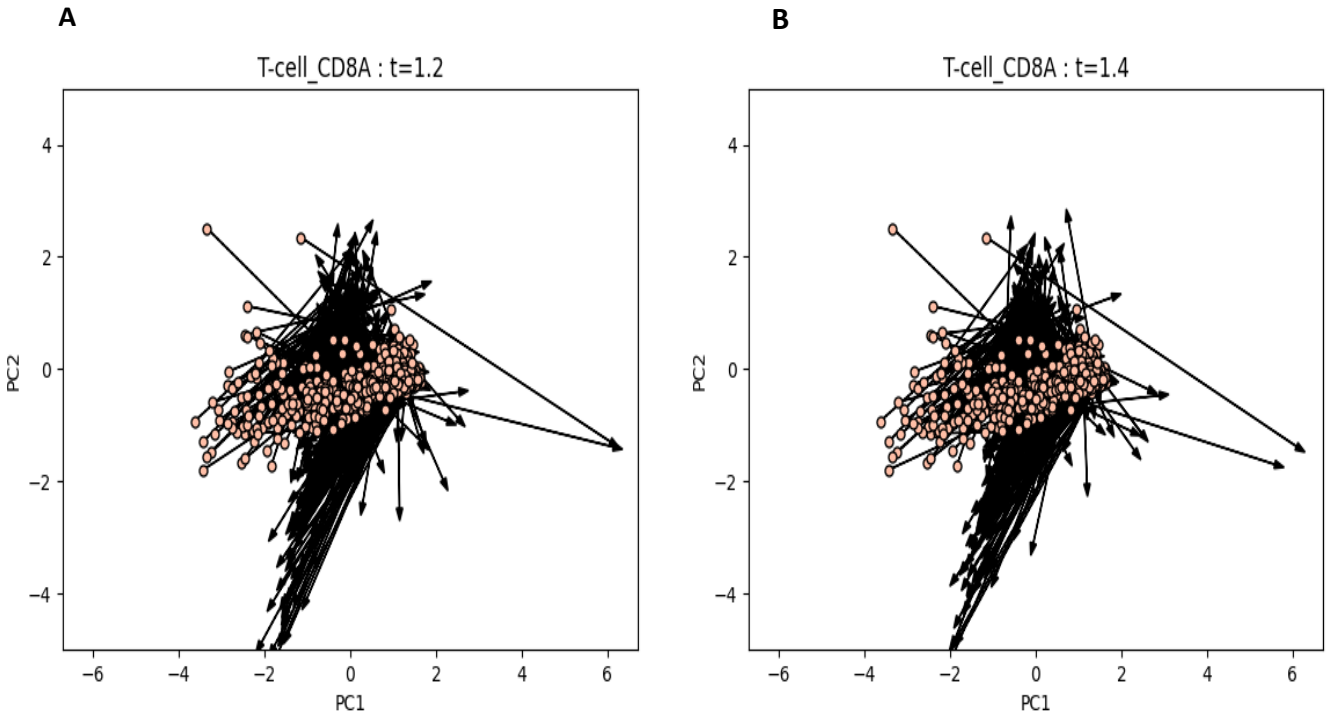


Fig 5: *Predicted Future Gene Expression of CD8A T-cells.* ***A:*** *Future gene expression states of CD8A T-cells projected into PCA space at t = 1.2. The arrows represent the RNA velocity vectors, indicating the predicted direction and magnitude of future gene expression changes.* ***B:*** *Similar projection at t = 1.4, showing how the predicted future gene expression states continue to evolve over time.*

7.2 Proximity to Tumor Cells and Permutation Test

Our data was derived from a cancer patient biopsy using Dr. Alon's ExSeq method. From this dataset, we extracted the proximity of each cell to tumor cells. Our analysis revealed that 19% of the CD8A cells were in close proximity to tumor cells.

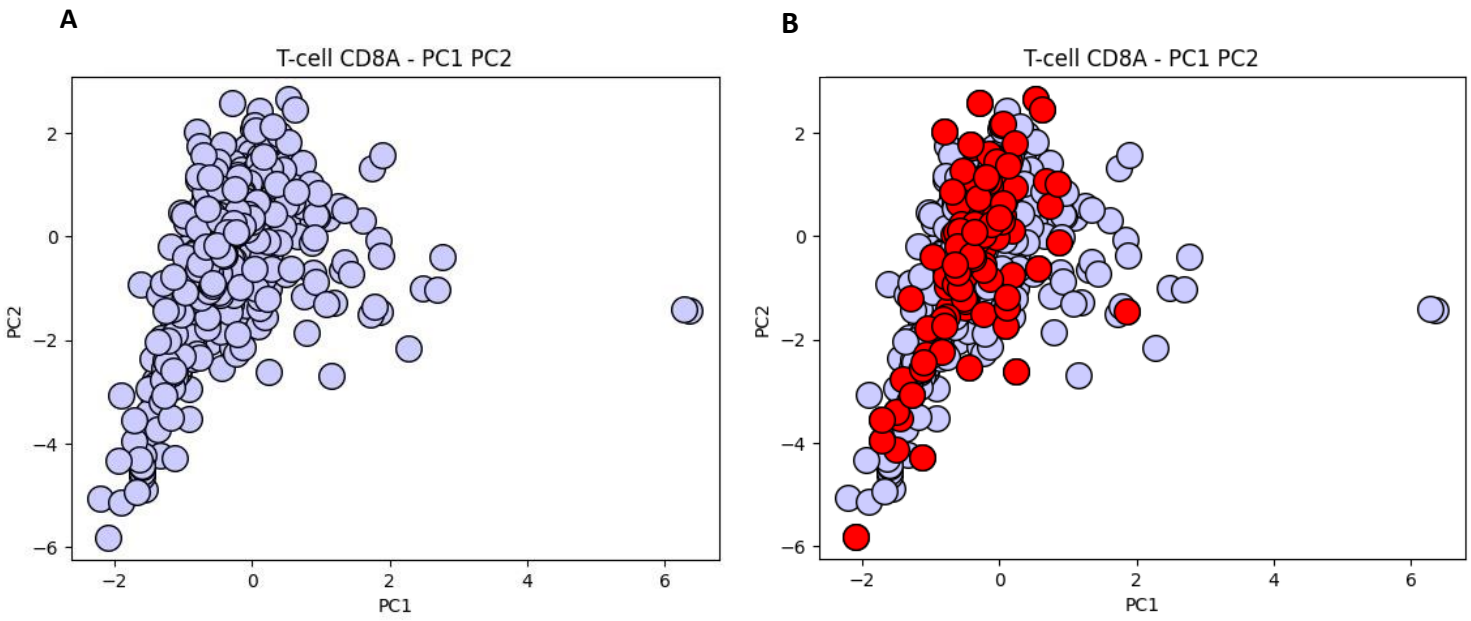


Fig 6: *The image shows two scatter plots of T-cell CD8A populations in PCA space (PC1 vs. PC2).* ***A****: the cells are shown in light blue, representing the overall distribution of CD8A cells.* ***B****: the cells in red represent the 19% of CD8A cells that are in close proximity to tumor cells.*

To achieve a better statistical understanding and identify potential patterns, we divided the population into three distinct clusters. Notably, Cluster 2 exhibited a significantly higher proximity rate, with 26% of CD8A cells located near tumor cells—1.3 times the overall average.

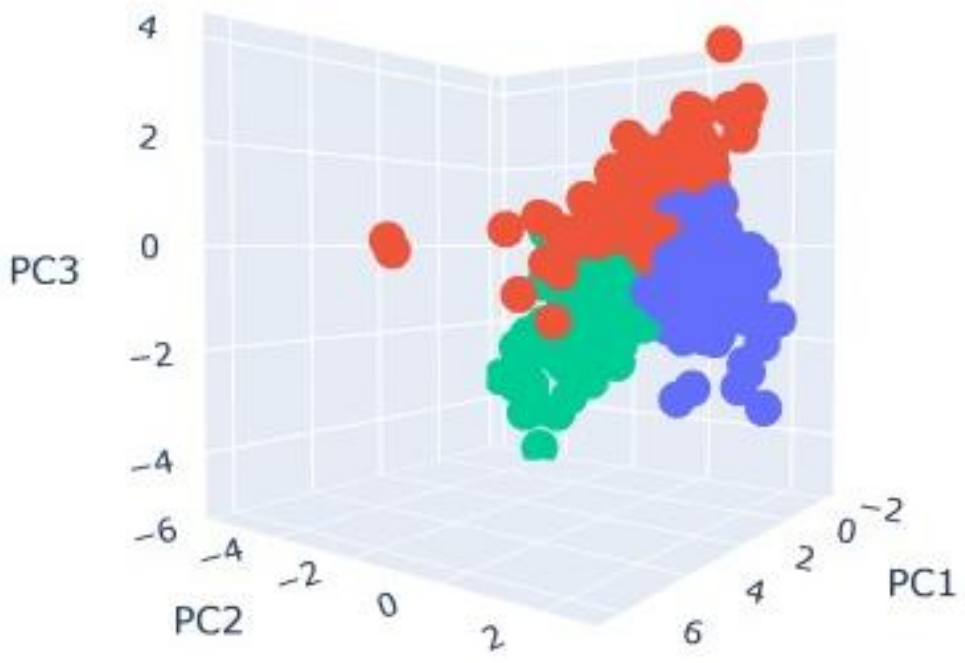


Fig 7: *3D scatter plot shows T-cell CD8A populations clustered based on their gene expression profiles, with colors representing distinct cell groups across three principal components (PC1, PC2, PC3).*

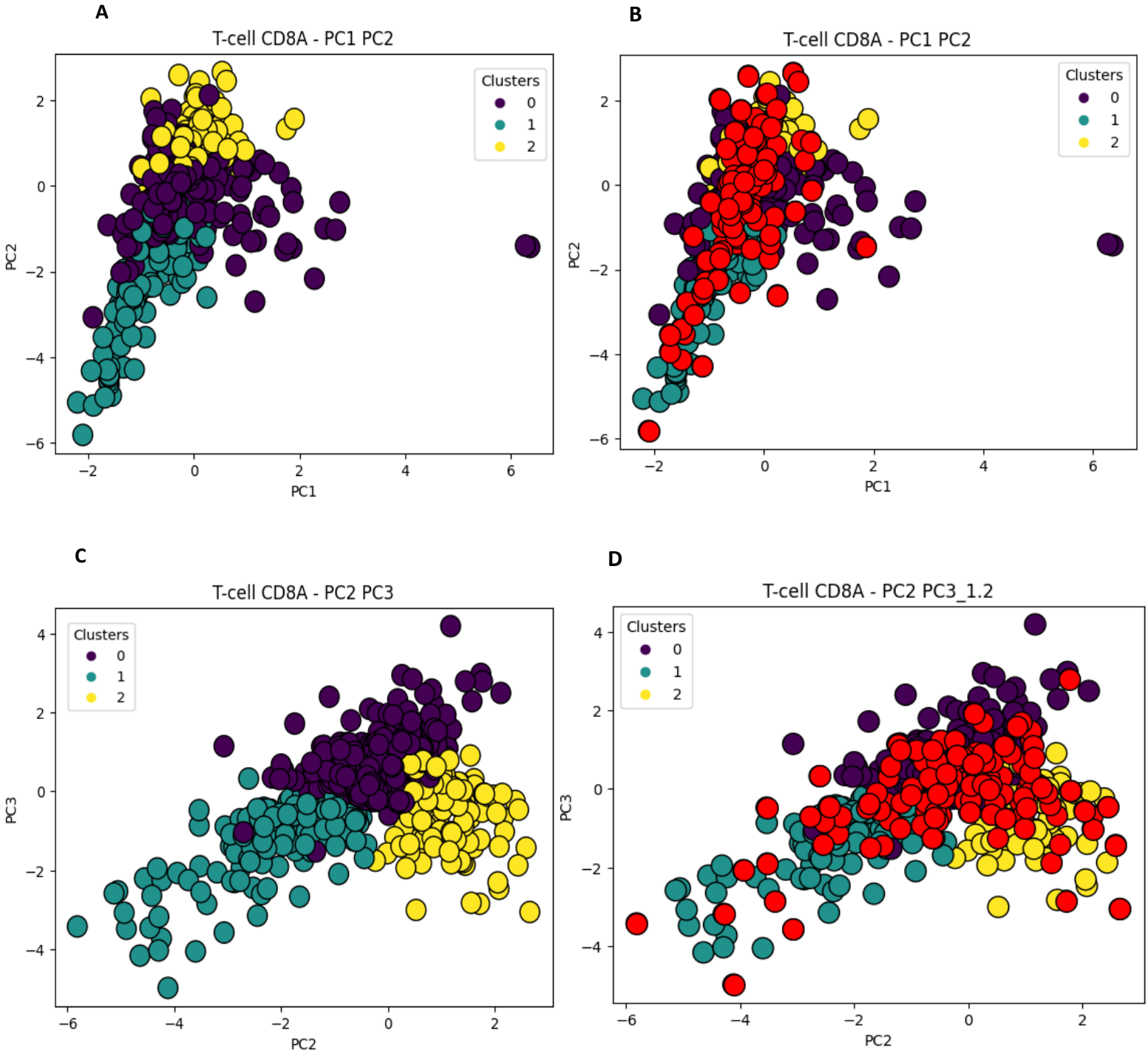


Fig 8: *Clustering of CD8A T-cells and Proximity to Tumor Cells.* ***A:*** *The PCA plot shows the division of CD8A T-cell into three distinct clusters, represented by different colors.* ***B:*** *The same PCA plot where res dots represent CD8A cells in close proximity to tumor cells, indicating a potential interaction or influence based on spatial proximity.* ***C:*** *The same as A, but showing the clusters along the PC2 and PC3 axes.* ***D:*** *The same as B, but showing the clusters along the PC2 and PC3 axes.*

8 Supplementary Figures

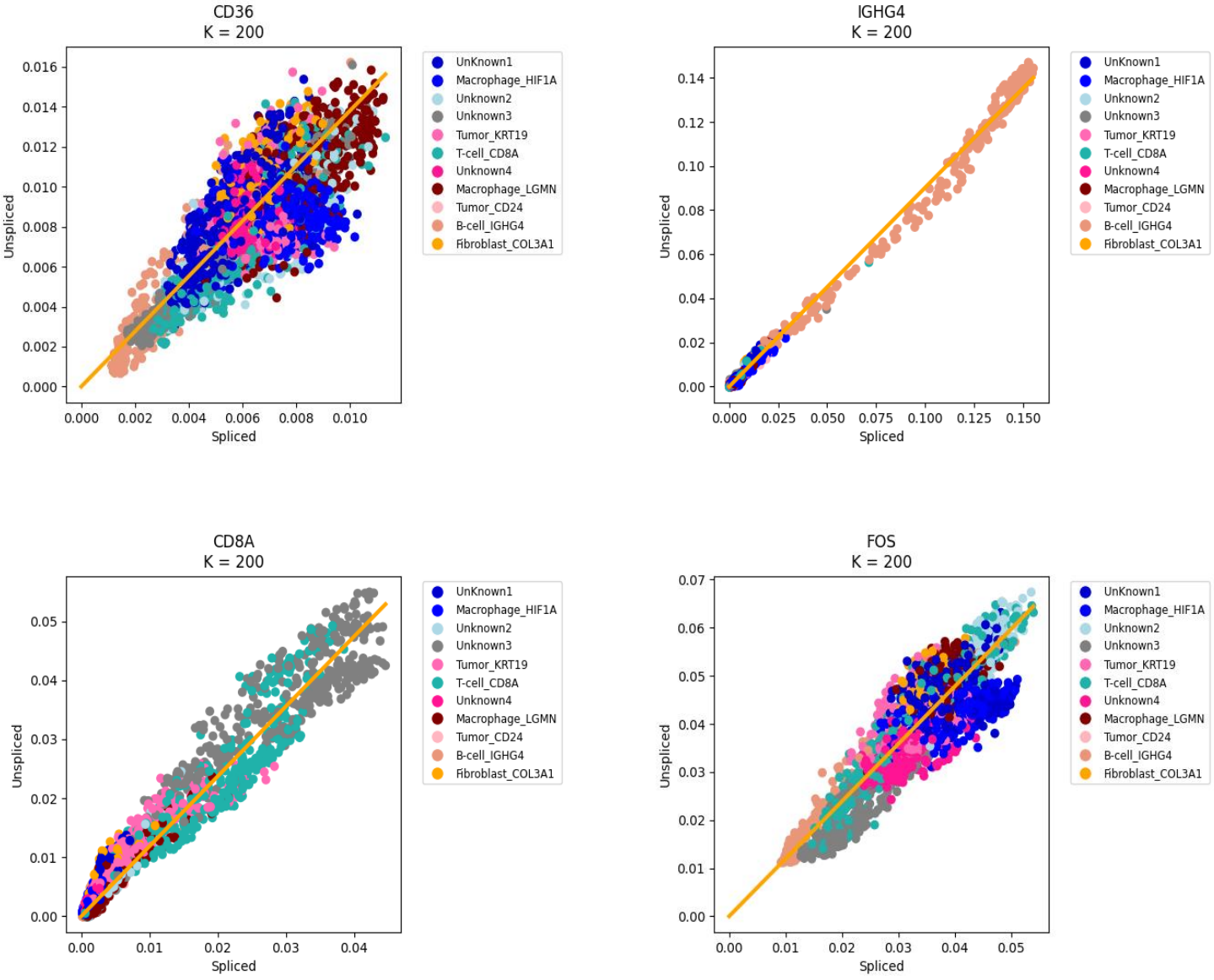


Fig 9: *Genes displaying a well-defined elliptical shape.*

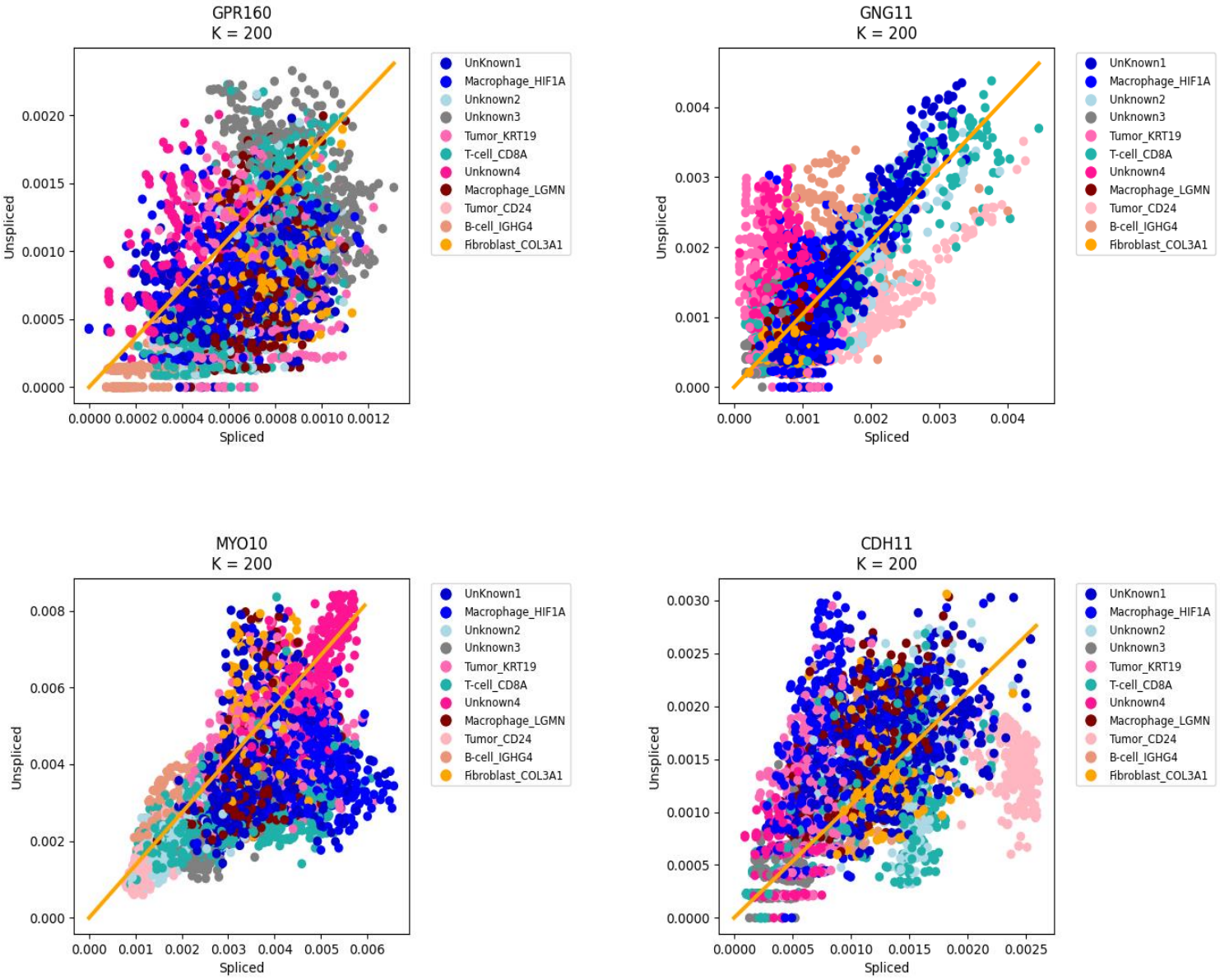
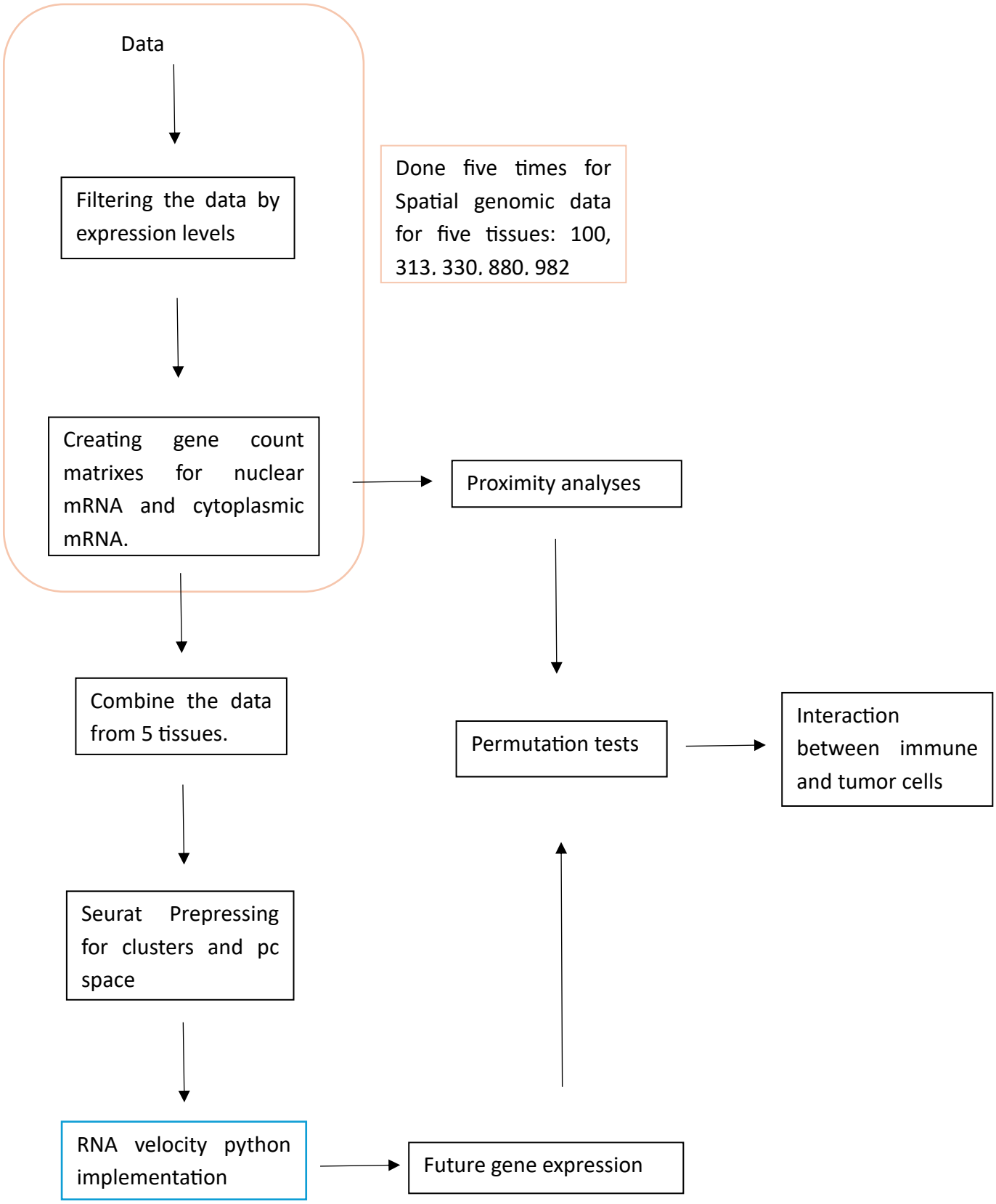
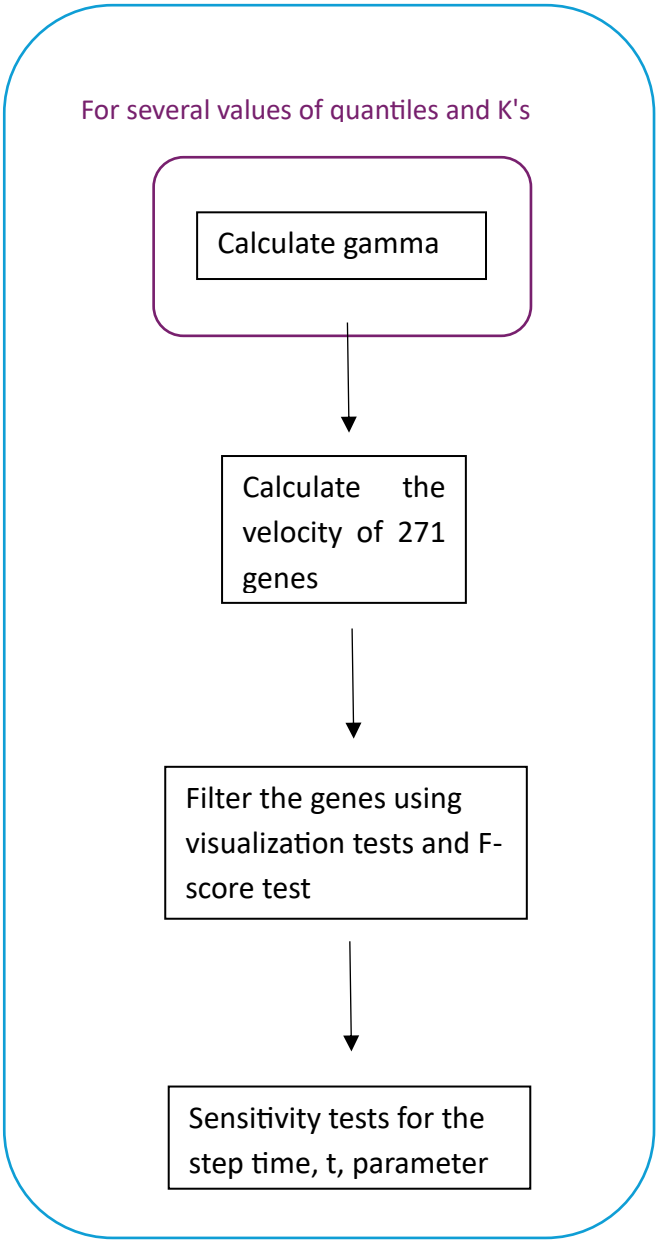


Fig 10: *Genes that deviate from the expected elliptical pattern.*

9 Block diagram



RNA velocity python implementation- hyperparameters



10 Summary and Conclusions

The primary goal of this project was to explore whether interactions between immune and tumor cells can be inferred from their proximity and the gene expression dynamics of immune cells over time. To investigate this, we used spatial genomic data from five tissues taken from a biopsy of a breast cancer patient, obtained via Dr. Alon's ExSeq method. This data provided the spatial locations of cells within tissues along with their initial gene expression levels. Combining this with RNA velocity, an approach that estimates future gene expression based on the ratio of spliced to unspliced mRNA, allowed us to make informed predictions about the interaction between immune and tumor cells.

We started by organizing the raw data, focusing on cells with gene counts detected in both the nucleus and the cytoplasm. Next, we filtered the data to include only those cells and genes that appeared across all five tissues, resulting in two matrices: one for nuclear (spliced) mRNA and the other for cytoplasmic (unspliced) mRNA. From this, we retained a total of 4,530 cells and 271 genes for further analysis.

Using the Seurat package in R, we clustered the cells based on gene markers, using PCA to reduce the dimensions for initial gene expression data. This PCA space was then employed for the subsequent RNA velocity calculations. To ensure accuracy, we fine-tuned key hyperparameters for the RNA velocity model, focusing on the quartile values and the number of nearest neighbors K for the gamma calculation. Sensitivity testing indicated that using the 10% quantiles and 200 neighbors yielded the most reliable gamma estimates.

Once gamma values were calculated and each gene's expression trajectory plotted, we filtered the results to focus on biologically relevant genes. The first round of filtering was visual, with genes displaying an elliptic shape being classified as part of the "Ground Truth" group, while others were excluded. A second round of filtering was based on expression levels: low-expressing genes were removed to avoid noise, as they did not significantly contribute to the analysis. To determine appropriate expression cutoffs, we used precision, recall, and F-score testing.

Next, we determined the optimal time step t for calculating future gene expression levels. According to La Manno's article, the size of S(t) should remain unchanged from S(0), as this ensures that the predicted change reflects only the direction of gene expression, not its magnitude, which would improve the prediction accuracy.

With all hyperparameters optimized, we calculated RNA velocity and focused on CD8A T-cells, the main immune cell type present in the dataset. We then analyzed the proximity of these

immune cells to tumor cells, revealing that 19% of the general CD8A population was located near tumor cells. To ensure these results were statistically significant, we clustered the T-cells into three groups based on their expression levels. Among these, Cluster 2 showed a notable proximity of 26% to tumor cells. Using permutation tests, we validated the statistical significance of Cluster 2, with a p-value of less than 0.05.

In conclusion, based on their proximity to tumor cells and their predicted future gene expression, T-cells in Cluster 2 exhibit evidence of interaction with tumor cells. This suggests that spatial positioning combined with gene expression dynamics can provide important insights into immune-tumor cell interactions.

11 Future Directions

Our study has provided valuable insights into immune-tumor interactions using data from a breast cancer patient, laying the foundation for further investigation. However, to fully explore the potential of this research, several improvements and extensions can be made. These suggestions aim to enhance the depth and applicability of the findings, allowing for a broader understanding of gene expression and immune responses in different types of cancer. By expanding the dataset, incorporating additional immune cells, and leveraging machine learning for gene filtering, future research can address the limitations of the current study and provide more comprehensive insights into immune-tumor dynamics. The following recommendations highlight key areas where further exploration could significantly enhance the impact of the research.

1. **Expanding the Dataset:** To improve the robustness and generalizability of the research, we suggest significantly expanding the dataset. The current data was derived from a breast cancer patient, and there are known differences in the tissues between breast cancer and other types of cancers. By increasing the number of biopsies collected from a broader range of cancer types and patient profiles, researchers can capture a more comprehensive view of immune-tumor interactions. This broader dataset will allow for a more accurate and reliable understanding of gene expression changes across different cancer environments, enhancing the predictive power of the computational model.

2. **Incorporating Additional Immune Cells:** Future research will also aim to incorporate more immune cell types. While our current study primarily focuses on T-cells, particularly CD8A cytotoxic T-cells, other immune cells such as natural killer (NK) cells, regulatory T cells (Tregs), and macrophages could be critical in understanding tumor behavior. Since these immune cells may interact differently with tumor cells across various cancers, incorporating data from these additional immune cells could provide a more nuanced understanding of immune-tumor dynamics in different tissue types.

3. **Improving Gene Filtering with Machine Learning:** One of the critical steps in our current workflow involves manual visual inspection of gene expression patterns to filter out noise and irrelevant genes. This process, though effective, is time-consuming and subjective. To streamline and enhance this step, we propose incorporating machine learning algorithms into the gene filtering process. By training models to recognize patterns indicative of meaningful gene expression dynamics, the filtering process

could be automated and enhanced for greater precision. This approach would be particularly beneficial in identifying subtle differences in gene expression across various tissue types, such as between breast cancer and other cancers. Automated filtering could reduce human bias, improve the accuracy of gene selection, and accelerate the overall workflow.

In summary, these are just a few of the potential directions that could help strengthen and broaden the scope of our research. But these suggestions are not exhaustive. There are undoubtedly many other approaches and advancements that could be explored to further refine and enhance the research. As new technologies and methods continue to evolve, future researchers may uncover additional strategies that will contribute to a deeper understanding of immune-tumor interactions and improve the overall impact of our findings.

12 References

[1] Johns Hopkins inHealth, Immunotherapy: Precision Medicine in Action

[2] Sharma, P. et al. Primary, Adaptive, and Acquired Resistance to Cancer Immunotherapy. Cell 168(4): 707–723 (2017).

[3] Kharchenko Lab and Linnarsson lab, Velocyto.

[4] Chen, D. S. & Mellman, I. Elements of cancer immunity and the cancer-immune set point. Nature 541, 321-330 (2017).

[5] Shayon, M. et al. Cancer prognosis and immune system. Microbial Crosstalk with Immune System, (2022)

[6] Alon, S. et al. Expansion sequencing: Spatially precise in situ transcriptomics in intact biological systems. Science 371, (2021).

[7] La Manno, G. et al. RNA velocity of single cells. Nature 560, 494–498 (2018).

[8] Geeks for Geeks. Principal Component Analysis (PCA).

[9] Geeks for Geeks. UMAP: Uniform Manifold Approximation and Projection.

[10] IBM, What is Linear Regression

[11] Stat Yale Edu, Linear Regression (1998)

[12] Geeks for Geeks. K-Nearest Neighbor(KNN) Algorithm. 2024

[13] Dagang Wei, Essential Math for Machine Learning: Confusion Matrix, Accuracy, Precision, Recall, F1-Score. Medium (2024)

[14] La Manno, G. et al. RNA velocity of single cells.

[15] Ibis.

[16] Ibis.