Determining Trajectory of Cells using Optimal Transport

Mayank Ghogale

June 2022

UBC

THE
UNIVERSITY OF
BRITISH
COLUMBIA

The University of British Columbia
Guide: Prof Dr.Geoffrey Schiebinger
Dr.Mattheiu Heitz

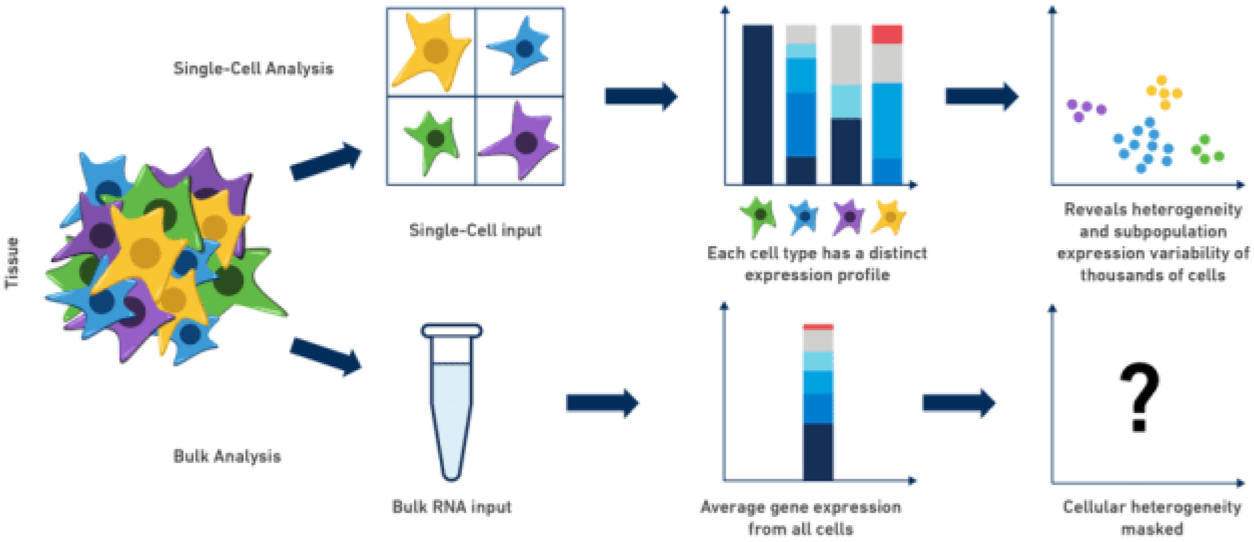
Contents

1 Introduction

The goal of the project was to determine the trajectory of immunology cells also known as the T-Cells which are the second level of defense mechanism in a human body. The data of these cells was provided to us by our collaborating group from University of Pittsburgh, USA. The path to this was via the analysis of Single Cell RNA Sequencing Data of the T-Cells with the help of Optimal Transport. Single Cell RNA Sequencing and analysis of the data obtained from the same along with Optimal Transport are huge topics of study and in this report, I will try to describe them as much as possible.

2 Single Cell RNA Sequencing

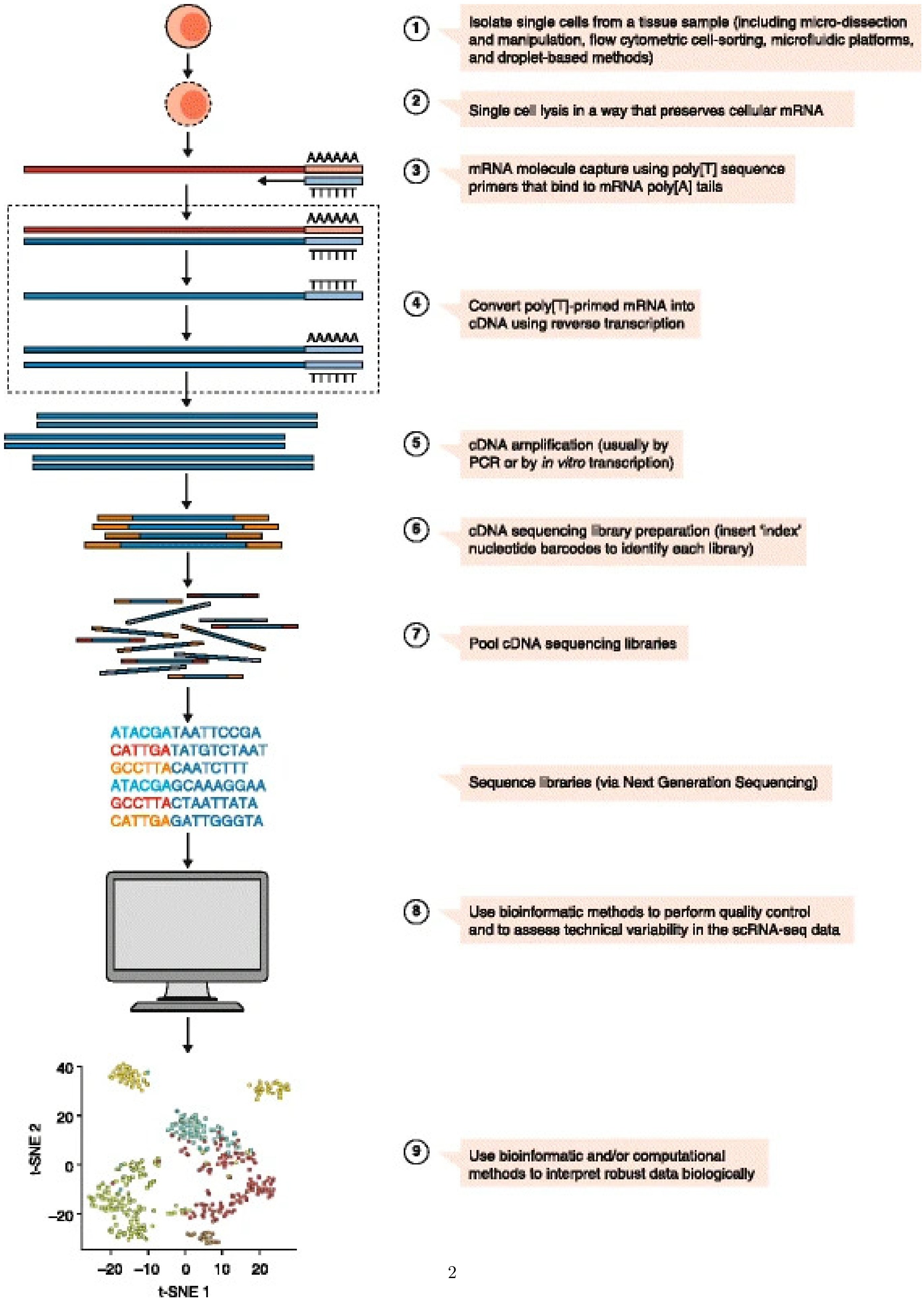
Single Cell RNA Sequencing is a technique where in a single type of cell is analyzed rather than analyzing a bulk of cells. The data obtained by such experiments is more accurate and specific towards the target cells than the one obtained from a bulk of cells. We can understand this by drawing a simple analogy of fruit juice and eating the fruits individually. When we make a blend of different types of fruits, the individual fruits lose their identity which is not the case when we consume them individually.


Single-Cell Analysis
Single-Cell input
Each cell type has a distinct expression profile
Reveals heterogeneity and subpopulation expression variability of thousands of cells
Tissue
Bulk Analysis
Bulk RNA input
Average gene expression from all cells
?
Cellular heterogeneity masked

The image describes the difference between the two techniques and we can see how the single cell sequencing technique can help us get more specific data. The image is taken from the blog on Single-Cell RNA Sequencing by 10x Genomics available **here**.

The RNA Sequencing Data has a lot to offer and if analyzed well can be used to address various topics like the evaluation of the developmental processes, analysis of cancer evolution and characterisation of cells. The extraction of the single cell RNA sequence data from the sample is a study in itself which is summarised in the image below obtained from an interesting article which can be found **here**.

As seen in the image the process has several steps right from isolating the single targeted cells from the tissue sample to the involvement of sequencing libraries. The penultimate and ultimate steps of the image are the ones we are interested in this report which involve first doing a quality check on the data, then preprocessing the data which then can finally be used for making biological interpretations with the help of mathematical models.



1. Isolate single cells from a tissue sample (including micro-dissection and manipulation, flow cytometric cell-sorting, microfluidic platforms, and droplet-based methods)

2. Single cell lysis in a way that preserves cellular mRNA

3. mRNA molecule capture using poly[T] sequence primers that bind to mRNA poly[A] tails

4. Convert poly[T]-primed mRNA into cDNA using reverse transcription

5. cDNA amplification (usually by PCR or by *in vitro* transcription)

6. cDNA sequencing library preparation (insert 'index' nucleotide barcodes to identify each library)

7. Pool cDNA sequencing libraries

Sequence libraries (via Next Generation Sequencing)

8. Use bioinformatic methods to perform quality control and to assess technical variability in the scRNA-seq data

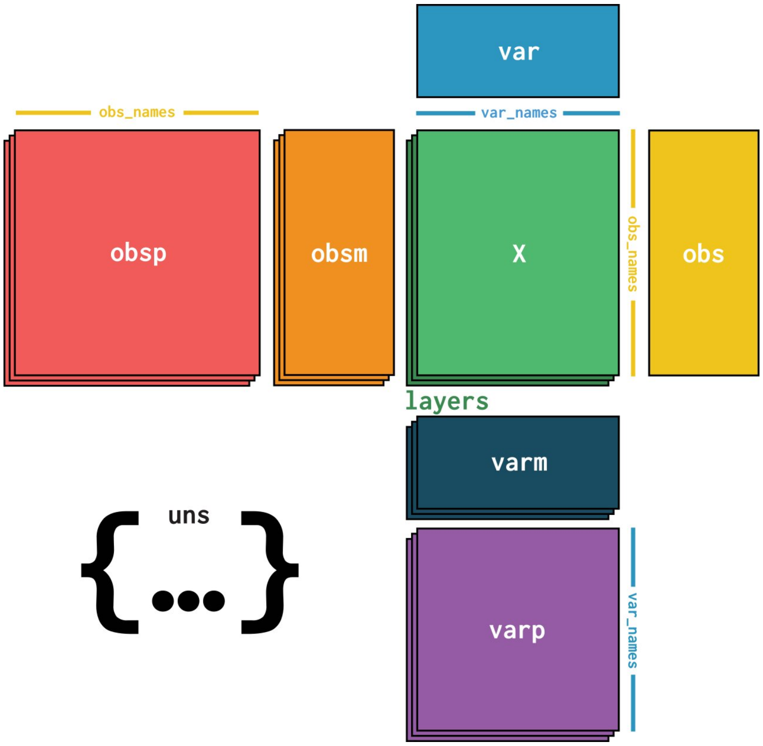
9. Use bioinformatic and/or computational methods to interpret robust data biologically

3 Preprocessing of the Single Cell RNA Sequencing Data

The preprocessing marks the most important step in the journey to analyze the data obtained from the single cell RNA sequencing. It consists of several steps which are followed sequentially and they are most commonly done by using **Seurat** package in R while the Python the equivalent package is **ScanPy**. The preprocessing as mentioned is done in several steps which are described below.
Before doing any of the steps, the data first has to be read from a h5ad format file which is described in the following section.

3.1 Annotated Data

The Annotated Data or commonly abbreviated as anndata is divided into several parts as shown in the figure (figure obtained from the documentation site of **Anndata**).



It is a compressed form of representing the data and mentioned below are the details of what each section of annotated data consists of.

i. **Sparse Matrix(X)** :- This is a sparse matrix with number of rows as number of observations(cells) and number of features as the number of columns(genes). Each value in the sparse matrix represents whether a particular gene is expressed in the cell or not.

ii. **Observation(obs)** :- This is a dataframe with number of rows equal to the number of observations(cells) consisting of various columns which describe the cells. Some of the common columns which make up the obs data-frame are the name of the cell, information about the sample from which it is obtained, information regarding the time at which observation was taken and other such metadata. The information about which cluster the cell belongs to, once calculated is also added to this dataframe.

iii. **Variable(var)** :- This is a dataframe with one column and number of rows equal to the number of genes or features. Usually this contains the names of the genes.

iv. **Unstructured(uns)** :- This dataframe contains information about all the unstructured data like information about dimensionality reduction and various other such aspects.

v. **Variable/Observation- level matrices(varm/obsm)** :- These are similar to **obs** and **var** which are explained above, the only difference being these contain the information in multi-dimension while **obs** and **var** have them in a single dimension.

vi. **Layers(layers)** :- Layers are used to store different forms of the data, for instance the one before and after normalization.
The Annotated Data is stored in a special file format of h5ad and hence these files have the extension of .h5ad. These files are generally huge in size occupying a lot of space in the memory but there are ways to make these files more efficient so as to they occupy less memory in the disk.

3.2 Preprocessing and Clustering of the Data

The preprocessing of the data is carried out in multiple steps each of which is described below. The preprocessing starts with **Basic Filtering** and ends with **Finding the Marker Genes** and each step is described below in detail. The data is first loaded from the h5ad using the ***read_h5ad*** function from ScanPy.

- Once the data is loaded, the first step is to do the **Basic Filtering**. In this step, the genes which are expressed in number of cells which is below the threshold and number of cells which express number of genes which are below the threshold, are removed using the ***filter_genes*** and ***filter_cells*** functions of ScanPy.

- After the above step is performed, the cells having too many mitochondrial genes or having too high total counts are removed using the ***calculate_qc_metrics*** function from ScanPy.

- Following the above step, data is normalized and then transformed(usually **log-transform** is used but in this project, **SCTransform** was used.)

- Once the data is transformed, the highly variable genes are identified using the inbuilt ScanPy function and the annotated data is modified to store only these genes. Following this step, **Principal Component Analysis** is applied on the data to reduce the dimensions of the data which also reveals the main axes of variation and also denoises the data.

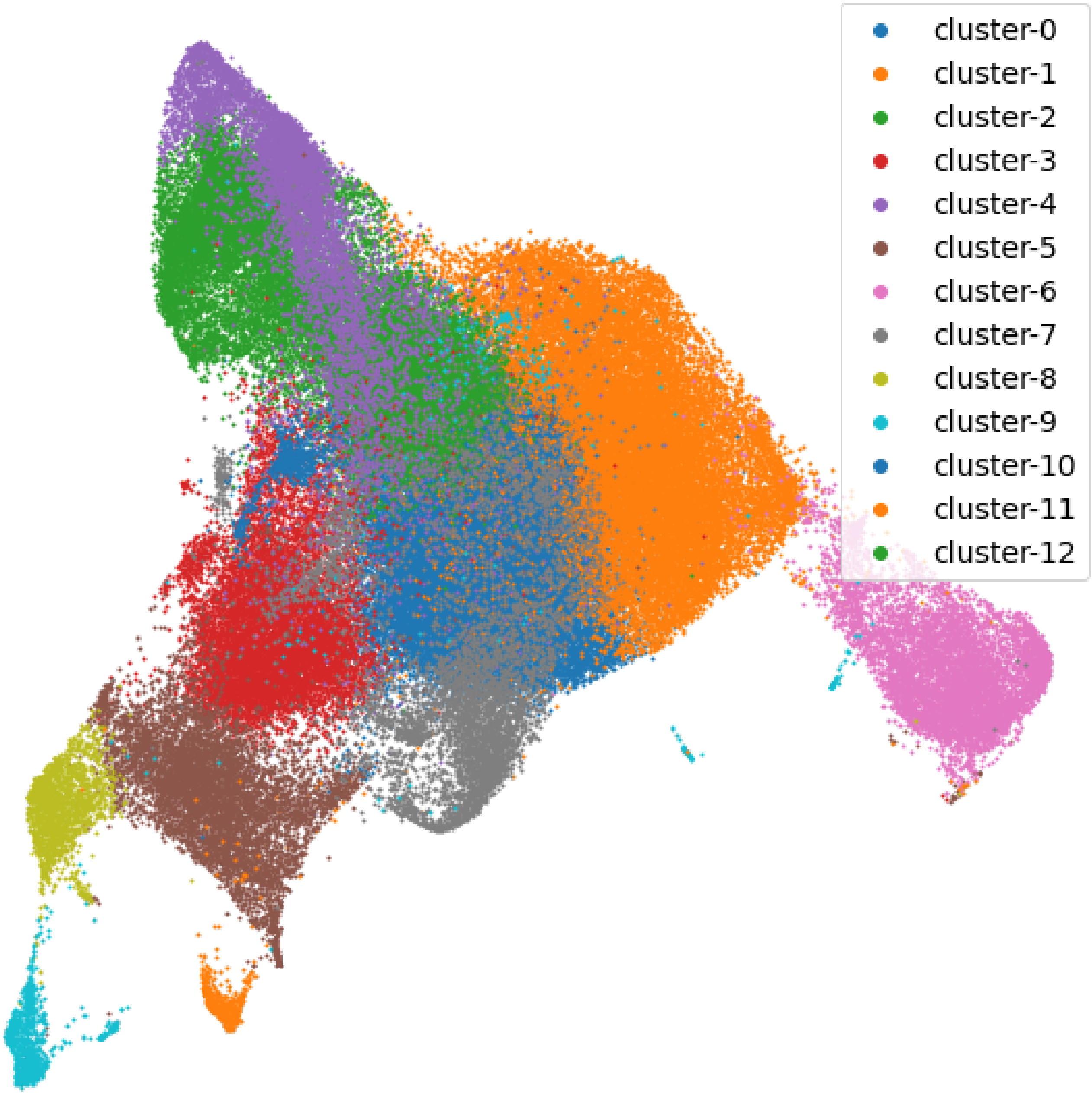
- The last three steps namely **Computing the Neighborhood Graph, Embedding the Neighborhood Graph and Clustering the Neighborhood Graph** are taken to divide the cells into clusters where in similar cells constitute the clusters. At the end of these three steps, we get the cells divided into clusters depending on their properties and these clusters can be well visualized with the help of **UMAPs or FLEs**.

- In Computing the Neighborhood step, the neighborhood graph of cells is calculated using the PCA representation of the data matrix.

- In Embedding the Neighborhood step, the graph of neighborhood is embedded in two dimensions with the help of either **UMAP** or **t-SNE** the formal being better as it preserves the trajectories.

- In the final step of Clustering the Neighborhood, the cells are clustered based on the data obtained from the step of Computing the Neighborhood.

The UMAP below shows what the cells look like after the entire preprocessing and clustering steps are complete. As seen in the image, the data has cells which can be divided into a total of thirteen clusters. Now, we can check for the biological importance of each cluster and decide what analysis is to be carried out on which cluster.

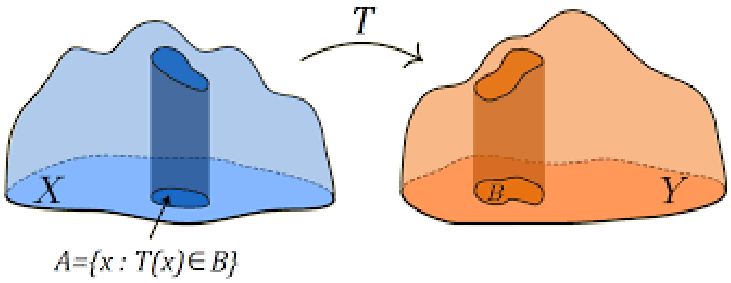

cluster-0
cluster-1
cluster-2
cluster-3
cluster-4
cluster-5
cluster-6
cluster-7
cluster-8
cluster-9
cluster-10
cluster-11
cluster-12

4 Inference from the Waddington OT

The preprocessed data was then applied to the Waddington OT model to get the inferences. This was done in accordance to the tutorials mentioned on **Waddington OT**. Before we can get into details of how the **Waddington OT** works we need to understand some basics of Optimal Transport which is described in the upcoming section.

4.1 Optimal Transport

Optimal Transport, as explained **here** is the general problem of moving one mass distribution to another as efficiently as possible. Examples are using a pile of dirt to fill a hole of the same volume while minimizing the distance moved or associating two sets X and Y having the same number of data points(N) in them with each other such that the cost incurred for the match; which is c(x,y), is minimized. The image below describes Optimal Transport very nicely and is taken from the notes of Prof Matthew Thorpe, Centre for Mathematical Sciences, University of Cambridge.



Optimal Transport is a huge field of study in mathematics and its applications range from Optimization to Image Processing and Trajectory Inferencing from the Single-Cell RNA Sequencing Data. I had referred to tutorials of Optimal Transport from **Numerical Tours** to understand the concepts better.

4.2 Inference of the growth rate of cells

The first step in calculating the trajectories was to determine the growth rates of the cells from the given proliferation and apoptosis scores which are in turn obtained from the gene scores. First the birth and death rates are determined from the proliferation and apoptosis scores of the gene scores using the beta and delta functions respectively. Growth Rate is the given as the exponential of the difference between the birth and death rate.

4.3 Predicting cell growth rates over time

The observation part of the Anndata(Anndata.obs) along with the number of growth iterations and hyper parameters are passed to the Waddington-OT model. The growth iterations signifies how many growth rates will the Waddington OT model predict and the time taken for computation is so directly proportional to the number of iterations. We also pass the timestamps between which the inference is expected and for this project it was the days parameter of the Anndata.obs dataframe. The two hyper parameters along with $\epsilon\ are\ \lambda 1,\ \lambda 2$.

4.4 Output of the Waddington OT Model

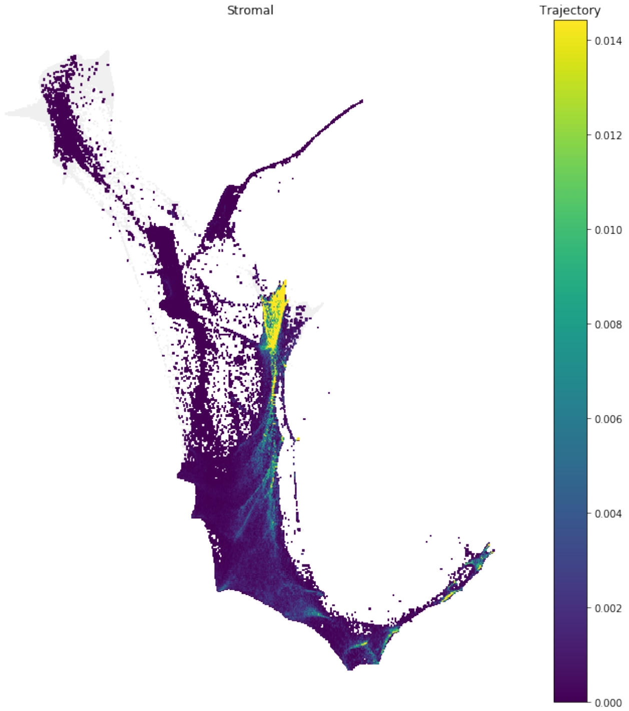
The output given by the Waddington OT is known as a Transport Map which is also a h5ad file with variable file having the names of genes and the observation part consisting of the growth rates of cells. The obtained file occupies a lot of space and this step is computationally expensive. The time required for computation of the file increases exponentially as the value of the number of growth iterations increases.

4.5 Computation of the Trajectories

Once the model computes the required number of growth rates, then we use the obtained growth rates to inference the trajectories of the cell. To do so, we first calculate the trajectories by first calculating the population list and then passing it to the function computing trajectories from the list of populations at different days.
The list of population is made by appending populations of different days which in turn are calculated using cell set. Cell set is nothing but a dictionary consisting of key as the name of the cell and value as the cluster number to which the cell belongs. To calculate the population at different days from cell sets, we use function offered by the Waddington OT package. Once we have the list having populations, the trajectories are computed using the inbuilt function of the Waddington OT package. After this, the trajectories are visualized and inferred from, which is described in the following section.

4.6 Visualisation and Inference of the Trajectories

The obtained output from the function computing trajectories is again a h5ad file which is first read into a variable and all such outputs are combined to be stored in a list. The coordinates of the UMAP projection are also available which are read in from the metadata file into a variable. On this coordinates, the values of the sparse matrix of the computed trajectories are plotted with the help of colour bars and those help in the inference of the actual trajectory of the cells.



The image shows how the final trajectory looks like in a FLE Plot which is another way to visualize the data from Single Cell RNA Sequencing. The image is taken from tutorial of **Waddington OT**.

5 Conclusion

Using the Waddington OT model, the trajectories of T-Cells were determined between time points which were different days in this case. These trajectories were then inferred and verified using knowledge of biology in collaboration with the Pittsburgh group.