Machine Learning and Computational Biology:

1) Application Topic and Introductory Information:

- RNA traditionally known as the intermediary between DNA and protein, and passive conduit for information
- RNA also helps turn genes on and off, aid chemical reactions, slice and dice other RNAs, and build proteins through transcription
- harnessing RNA for use in medicines initiated the field of RNA therapies → first two RNA-based therapies used for hereditary *ATTR amyloidosis*
    - *Disease definition:* progressive and potentially fatal disorder in which abnormal proteins build up in nerves and organs such as heart
- biggest barrier to RNA therapy is delivering RNA to correct place in cells → really key for getting drugs in the liver as many proteins implicated in diseases
- three main categories: (1) those that target nucleic acids (DNA or RNA) (2) targeting proteins (3) encoding proteins
  for targeting nucleic acids, two distinct types of therapies that target nucleic acids: single-stranded antisense oligonucleotides (ASOs) | double-stranded molecules that operate through cellular pathway known as RNA interference (RNAi)
- ASOs are short stretches of modified DNA made up of about 13-25 building blocks, or nucleotides → molecules prevent mRNA from being translated into protein by several mechanisms (blocking start of translation or tagging mRNA for degradation)
    - inotersen is an ASO → main approved drug in 2019
- ASOs can also alter splicing, process that sculpts a precursor messenger RNA into its mature form → two of this type of ASO received FDA approval in 2016: nusinersen targets fatal inherited condition called spinal muscular atrophy | eteplirsen is treatment for Duchenne muscular dystrophy
- the latter is an example of exon skipping drug, using ASO to block only the mutated portion of a gene from being expressed = results in protein that is functional but lacks mutated portion that causes a pathology
- RNAi makes use of double-stranded molecules, making it harder to get into cells than ASOs but advantage is that fewer molecules are needed for therapy to be effective → RNAi involves small interfering RNAs (siRNAs) that are 21-23 nucleotides long or similar to molecules such as microRNAs that degrade mRNA and prevent it from being translated into protein
    - other amyloidosis drug approval called patisiran is siRNA therapy
- RNA therapies that use mRNAs are being used to develop personalized cancer vaccines and those for ZIka viruses
    - researchers also exploring whether these type of treatments can be used as protein-replacement therapies for rare conditions such as haemophilia (blood-clotting disorder)
- many researchers using hybrid strategies of three main RNA forms → Apic Bio using silence and replace which uses RNAi to silence a harmful gene and mRNA component that encodes a normal version of the corresponding mutated protein

RNA-Based Vaccines in Cancer Immunotherapy:

- RNA vaccines traditionally consist of messenger RNA synthesized by in vitro transcription using bacteriophage RNA polymerase and template DNA that encodes antigens of interest for specific disease → mRNA transcripts are translated directly in cytoplasm and then resulting antigens presented to antigen presenting cells for stimulated immune response

    1. alternatively, dendritic cells can be loaded with either tumor associated antigen mRNA or total tumour RNA and delivered to the host to elicit specific immune response
- cancer immunotherapy seeks to stimulate host antitumor immune response, leads to tumor shrinkage and improved clinical outcomes
    1. discovery of agents aimed at enhancing immune responses against tumors exploded → include cytokines, immune checkpoint inhibitors, adoptive T cell therapies, and numerous vaccine strategies
        - several drugs like ipilimumab and nivolumab are impressive for benefits shown in phaseIII
- vaccines can be more difficult to produce and shown more modest clinical responses in patients → moreso well-tolerated and safer therapy that offers potential to avoid drug resistance and obtain durable treatment responses promoter
    1. DNA plasmid administered to host and internalized by host cells
    2. Transcribed in nucleus and translated in cytoplasm by cell
    3. resulting proteins processed into peptides which are presented on the surface of host-antigen-presenting cells (APCs) with major histocompatibility complex molecules (MHC)
    4. peptide-MHC complex recognized by antigen-specific T cells, results in cellular host immune response
    5. RNA vaccines involve messenger RNA synthesized by in vitro transcription using bacteriophage's RNA polymerase and template DNA that encodes antigens of interest
    6. Host cells internalize mRNA transcripts, then translated directly in cytoplasm and then like DNA Vaccines, resulting antigens are presented to APC to stimulate an immune response
- important to recognize mRNA-encoded products are degraded by proteasomes and presented on MHC class I molecules to CDG+T cells and do not reach MHC Class II

Single-cell RNA sequencing Tech:

- mapping genotypes to phenotypes is challenging in biology and medicine → powerful method is currently transcriptome analysis, but transcriptome information in a cell reflects activity of a small subset of genes
    - body's cells each express unique transcriptome, and increasing evidence shows that gene expression is heterogeneous even in similar cell types
- majority of transcriptome analysis is based on assumption that cells from given tissue are homogeneous and these studies likely miss important cell-to-cell variability
    - most biological processes are stochastic, thus we need more precise understanding of transcriptome in individual cells for finding their role in cell functions and understanding how gene expression can promote certain genes
- using cDNA of individual cells, single-cell RNA sequencing was born → this helped provide high-resolution views of single-cell heterogeneity on global scale and allowed for finding differences in gene expression between individual cells has potential to identify rare populations that cannot be detected from analysis of pooled cells
    - eg. finding and characterizing outlier cells within population has potential implications for furthering understanding of drug resistance and relapse in cancer treatment
- with bioinformatics pipelines, can now learn more about highly diverse immune cell populations in healthy and diseased states

    - scRNA-seq is being utilized to delineate cell lineage relationships in early development, myoblast differentiation, and lymphocyte fate determination

Single-cell isolation techniques:

- first step for obtaining scriptome information from individual cell → uses limiting dilution in which pipettes are used to isolate individual cells by dilution
- Flow-activated cell sorting is most common now for isolating highly purified single cells → preferred when target cell expresses very low level of marker
    - in this method, cells are tagged with fluorescent monoclonal antibody, recognizes specific surface markers and enables sorting of distinct populations
- 
    - most common method is g: use bead that holds a barcoded bead that can bind to the RNA from cell lysis
- Computational challenges in scRNA-seq:
    - computational pipelines for handling raw data files is limited → not many tools and is still in infancy
    - first step is pre-processing data → once reads are obtained from well-designed scRNA-seq experiments, quality control performed
    - read alignment is next step and tools available for this procedure are used for bulk RNA and can be used here → when adding transcripts of known quantity and sequence for calibration and QC, low-mapping ratio of endogenous RNA to spike-ins is indication of low -quality library caused by RNA degradation

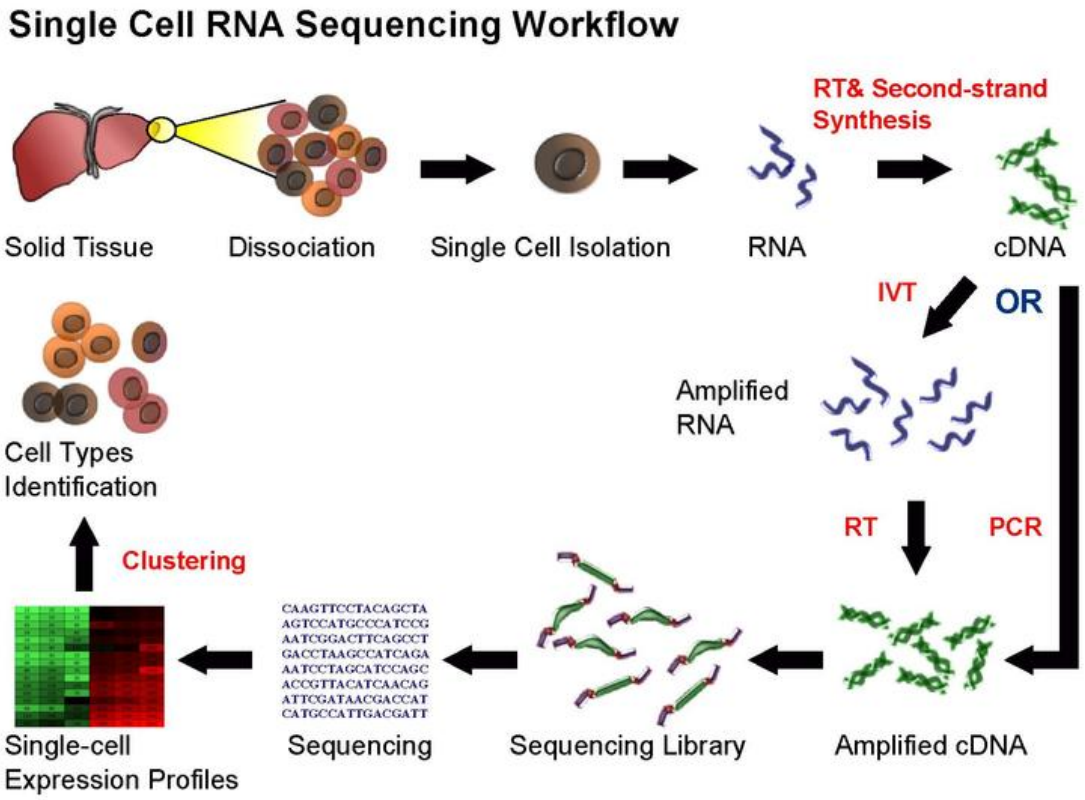
    - following alignment, reads allocated to exonic, intronic, or intergenic features using transcript annotation in format General Transcript
        - only reads that map to exonic loci with high mapping quality considered for generation of gene expression matrix → N x m where N=cells, m=genes
    - there is a presence of zero-inflated counts due to dropout or transient gene expression, which requires normalization to remove cell-specific bias
        - read count of gene in each cell expected to be proportional to gene-specific expression level and cell-specific scaling factors
- potential → scRNA-Seq could revolutionize personalized medicine in cancer (identify individual clones and biomarkers for patient and pick selected drug molecules for targets)

2) Based on the computational challenges that traditional histology faces and the potential of utilizing computational and data-science methods in better understanding our bodies, I want to build a program that can help better analyze and interpret single-cell RNA sequencing data of a set of tissues, and then analyze different features of these cells such as their gene expression activity (transcriptome), protein interactions, and cell localization. Because I am fairly new to this apart from the basic preprocessing methods (PCA, Normalization, Plotting, etc.), I'll be using several online tools and walk-throughs to better understand the bioinformatics pipeline and then write my own code on a custom dataset.

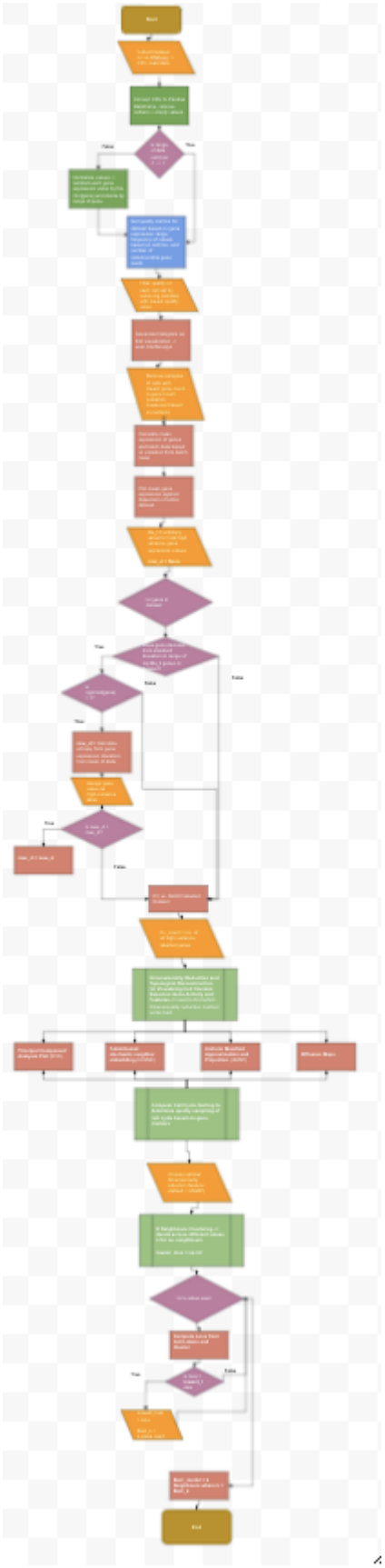
A) The program is designed to visualize and analyze sc-RNAseq data which ultimately looks at gene and cell activity for the individual types of cells in a microenvironment. Ultimately, I want to do this for a cancer dataset which looks at a tumor microenvironment and the different kinds of cells there are (T-cells, CD4 vs CD8+ T Cells vs NKCs in a tissue, FOXP3+/CD3+/SIGLEC 1 T-cell -> the primary goal will be to find patterns and essentially ID different kinds of cells and try to identify whether or not the immune system of the body is actually killing off the tumor.

Input: sc-RNAseq database, ~7500 tissue samples, several thousand gene expression data.

Output: clustered and labelled cell types in tissue based on gene expression markers.



B) Flowchart Design -> simplified and in-depth pipeline and procedure of steps. A successful implementation can accurately cluster the different cell types in the tumor microenvironment at a high degree of accuracy while also showcasing how certain cells and their respective behaviour can lead to other cell types and their respective interactions. Link is here



C) Algorithm Steps ->

1. Parse CSV of Dataset -> dataset includes gene expression levels, gene names based on global index, and other cellular features obtained from sc-RNAseq, has the type of cell as a label which can be used to optimize clustering task
2. Data preprocessing -> scale values from -1 to 1 using log scale by using mean normalization, remove any examples in dataset with missing values as those cannot be inferred correctly
3. Data visualization -> plot gene expression levels and cell counts in scatter plot and histogram respectively to identify high-variance cells based on their nomination from standard deviation
4. Dimension reduction (Algorithm 1) example -> PCA:
   This algorithm reduces the total number of redundant features and columns in the dataset and tries to embed them in a simplified manner. In our case, you have 7500 different columns and try to condense the data down to 2 features (or axis) to easily plot the topology and cell lineage of the cells. The following repeats for every column in set (N) (think of this as trying to turn a 7500-D plot to a 2D or 3D plot by removing redundant features and simplifying related features into a new feature that is an embedding of said relationships and accurately captures its interactions / relationships with other sets of features)
   a. Take features n-N/2 and (N/2)+1 of the different genes for all samples
   b. Calculate average of both axis by calculating average of both feature columns
   c. Shift data where average becomes center or origin (relative to 0, 0) or mean
   d. Fit line of best fit/slope set by minimizing the squared sum of the perpendicular distance for each datapoint to the line OR maximize squared sum of distances of perpendicular projects on best line of fit from the origin/mean
   e. Best fit computes Principle Component assigned to slope relative to origin (scalar value) to embed type of relationship + variance (slope is ratio of spread of the data)
      i. PC1 is fundamentally computing a linear combination of n-N/2 and N/2+1-N and its respective principal components
   f. Scale values with unit vector of 1 as slope to implement Single-Value-Decomposition (distance from origin to point is unit vector scaled known as eigenvector of PC1) -> use eigenvalues to determine relationship of PC1
   g. Sqrt of Eigenvalues = PC1 single value
   h. PC2 is perpendicular to PC1 (alternative or orthogonal representation of PC1 relationship with second axis, repeat process to compute scaled eigenvalues and vectors with loading scores)
   i. Determine most important PC by calculating Eigenvalues which represents the variation of PCs
      i. Repeat process for all features in PC plot, collect eigenvalues for all PCs generated and account for variation (not done in while loop, done at the same time, while loop comes at step h to compute principal components for all given dimensions on 2-D planes (perpendicular to all possible sets of perpendicular fits)

j. Plot all variances in scree Plot, pick d-highest (d represents dimension of PCA you want to reduce to, does not have to be 2 or 3 since this is all done using matrix multiplication in numpy)
k. Use respective data points on both PCs on d-D axis through SVD for both PCs (because all PCs are perpendicular representations of PC1, respective points can be found by their respective value)

5. Clustering (Algorithm 2 and 3) examples -> K-mean clustering or KNN depending on supervised or unsupervised:
   a. If labels provided in data:
      i. Create dictionary of model_saves based on random K-values (range of 2-12)
      ii. Iterate for each potential k-value in range (2, 12)
      iii. Iterate over a random portion (70%) of dataset, label as non-labelled while they have a ground-truth
      iv. Assign label to non-labelled datapoint based on the mode label of its k-closest neighbours
      v. Find Optimal K-Neighbour Model based on MSE of predicted label from closest neighbours and the ground truth, sum of MSE is loss of k-model
      vi. Lowest loss for k-model is final k-model
   b. If labels not provided in data:
      i. Iterate for each potential k-value in range(8, 20)
      ii. Randomly assign values to each **centroid** (centroids act as mean of cluster, and num. centroids = k)
      iii. Best_sum_accuracy is assigned as infinite
      iv. While the Best_sum_accuracy is higher than calculated Dunn index, repeat the following steps
      v. Iterate over each datapoint and calculate distance of each datapoint from each centroid, assign the data point to closest centroid
      vi. Calculate mean of current assigned data point based on their centroid assignment, this mean is now assigned as new centroid
      vii. Calculate Dunn index for new positioning of centroid, assign it to Best_sum_accuracy if it is less than
      viii. Repeat process until you align centroids

D) PseudoCode for Implementation

```
BEGIN
import necessary libraries
set pandas_df -> csv contents
pandas_df clear null and 0
set mean_expressions -> empty_dictionary with keys of sample num
if data not normalized:
      FOR gene expression in set
            FOR gene in gene expression
                  set blank in mean_expressions <- mean(gene expression)
                  (gene-min(gene_expression))/(max(gene_expression)-(min(gen
                  e_expression)))
set dict_counts <- cell_label count in pandas_df for every cell_label
```

```
set mit_counts <- mitochondrial gene label count in pandas_df
calculate variance of mit_counts, check outliers
pandas_df -> remove sample with lowest dict_count
pandas_df -> remove sample with highest mit_count
histogram plot -> range of gene expression frequency by mean gene
                  expression per sample
scatter plot -> datapoint by mean_expressions to frequency in histogram
set high_variance_num -> approximate number of outliers in scatter plot
set max_d -> None
add column to pandas_df: high_or_not variance
FOR gene in pandas_df:
      Only if variance from standard deviation is high and is n-th highest
      based on high_variance_num:
            Only if sigmoid(gene) = 1:
                  set new_d <- entropy of gene
                  set pandas_df high_or_not <- "high"
            Otherwise:
                  skip
                  Only if new_d > max_d:
                        set max_d <- new_d
      Otherwise:
            set pandas_df high_or_not <- "not"
set high_variance_count <- count label "high" in pandas_df column
                           [high_or_not]
#assign all calculations of plots to _plot_dict_
set plot_dict <- empty_dict
calculate PCA plot for pandas_df
calculate Td-SNE plot for pandas_df
calculate UMAP plot for pandas_df
calculate Diffusion plot for pandas_df

FOR every plot in plot_dict:
      calculate cell cycle sorting
      add plot_dict value column cell cycle sorting score
default = highest score in plot_dict

set no_label_df <- default but remove all cell labels
set label_df <- default

define compute_clusters:
      set no_label_cluster <- calculate K-means clusters with no_label_df
      set label_cluster <- calculate K-means clusters with label_df

set best_models <- empty_dict with 2 placeholders for no_label and label
                   cluster model
set min_MSE <- infinite

FOR iterations up to 200:
      run compute_clusters
      set mse <- calculate MSE loss between label_cluster and
                 no_label_cluster
      Only if mse < min_MSE:
            set min_MSE <- mse
```

set best_models <- no_label_cluster, label_cluster

visualize clusters for both no_label_best and label_best
END

E) Project Management

As part of all my projects, I like to keep a to-do list organized here with a respective **Notion** repository with all updates, code changes, and notes + I use version control to make sure I can go back to previous revisions and open-source it effectively.

Rough [TIME]line:
23rd: Look at checklists and make success criteria + complete proposal, do research on topic
24-25th: Do Smaller Exercises for Tests + Revise Plan Base on New Learnings
- **Datasets could potentially change**
- **Clustering approach can be fitted to dataset**

- Work on slideshow for presentation as project proposal is already made + research completed by now
- Get some market research done on this problem and assess who can I sell it to -> customer and individual labs

25th-27th: Work on implementation -> look at documentation and review biological concepts
- While building out implementation and small features, use modular programming and Agile-lite processes to document code and then write documentation for features accordingly

28th: Fix bugs and implementation details, compile documentation and practice presentation

For a more detailed timeline, refer to the Notion's included GANTT chart.

[SCOPE]ing out Features:
The ultimate goal is to help researchers and patients alike better understand their tumors on a biological and holistic level with a clear snapshot that can take the first step at identifying and organizing data in a visualization for further interpretation.

This is done through clustering, and so the first priority will always be differentiating between all different kinds of cells in the tumor with a high degree of accuracy such that it reflects the progression and level of the tumor.

The target audience is customers and ultimately researchers, and both require a similar interface that can easily help them plug in data and get a set of visualizations and key metrics of their tumor microenvironment (cell type frequency, scale of severity, etc).

With all of these considerations, the final scope of the project will include the following:

- [ ] A user interface in command line that can allow users to input a link to a publicly hosted CSV
- [ ] Providing Cell Viability and Gene Expression Patterns
- [ ] Deriving Variance of Data, Number of Outliers, and Number of Cell Types Found
- [ ] Plot that Visualizes Topology and Clusters of these Cell Types with Clear Labels
- [ ] Additional Readings for Patients to be more informed

Things that will definitely not be included (even if I get time, I should spend it on improving the quality of what I have):

- [ ] Providing clustering and insights of the types of cellular activity and lineages beyond labels provided in dataset
- [ ] Going deeper into gene expression markers and picking out key outliers which are responsible for specific clusterings and patterns based on prior knowledge

This is fairly barebones and minimizes key **[RISKS]** primarily because the scope is minimal with the hardest features probably being just the clustering algorithms (which themselves are automated). There is little cost to maintain and communications is not involved, yet the cost to deploy and scale on cloud architecture + customer service to not only help but get feedback on what features need expanding cna come at a later stage. However, **[QUALITY]** management is reduced because of the scope, as while we are offering key features useful to patients to analyze their cell composition, this is done through a command line and not necessarily a GUI because of the time constraints. Issue management ties in directly with risk management primarily because we have just 5-6 days to work on this project and failures will significantly hinder the progress, thus we've kept the complexity of the algorithms in use very minimal and limited the number of cells we want to have (dependent on simplicity of the dataset).

There are also several online exchanges and websites which go in depth to debugging and problem solving, so this can offer significant help when we do run into errors and reduces the risk.

While the quality for the final product will be high, there is a chance that there would be sacrifices in the presentation, primarily because of the technical depth and focus on the documentation and product. However, this is less of a risk since the time of the presentation is just 5 minutes, meaning that a quick overview of a proposal will prove sufficient.

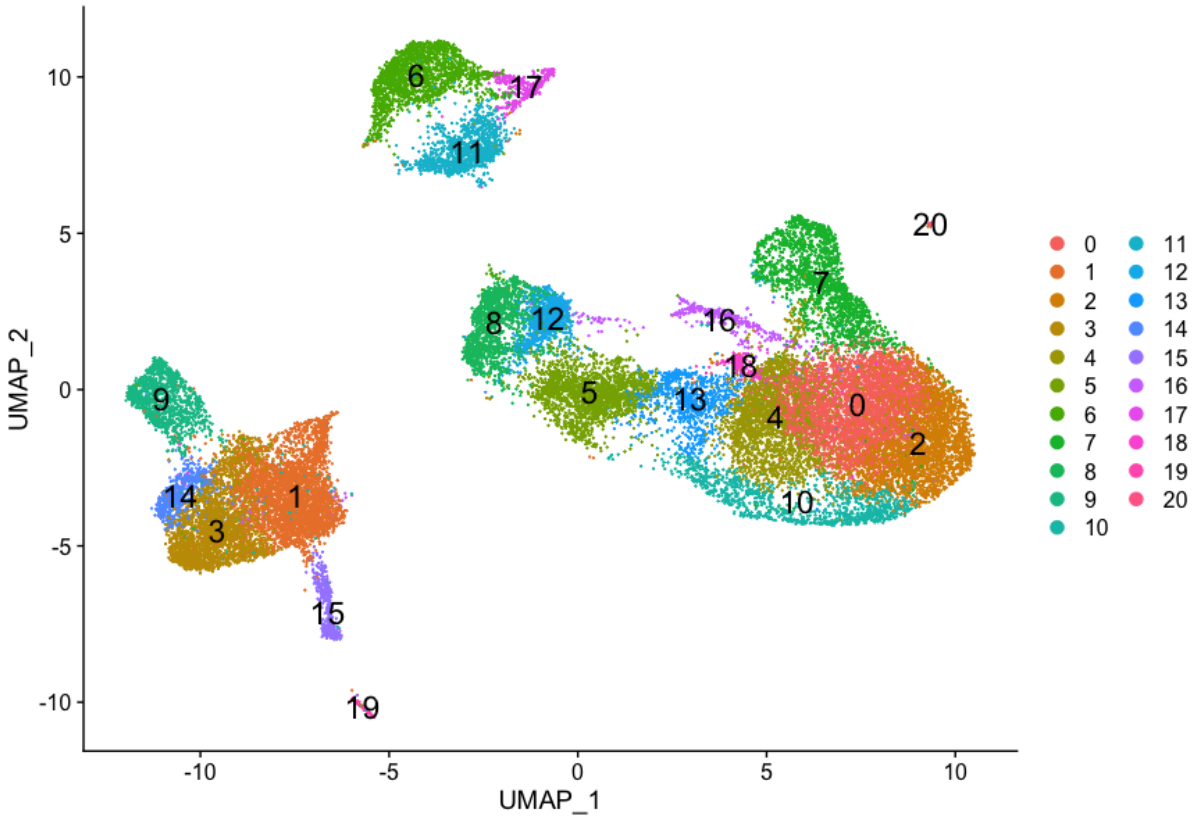
For issue management, I will end up using my notion and the resources I have above to prevent sidetracking across my project. A fitness for use can be demonstrated based on whether the visualizations are effective, clean, and simple to interpret along with a clear documentation.

If changes need to be made, I will deprioritize things based on how impactful they are to the final product quality (code) as this is the main purpose (if I need to compensate for time, I can reduce the visual aesthetic of the slides to make cleaner code and documentation). My primary concern is not having the necessary configuration for libraries or hardware requirements

(intel i7 -> most preferably, I can always create a .yml file online to install and manage my packages).

Mockup of final goal and vision of the project (most barebones final product):

Input: Please enter the URL of your csv file hosted online.
Output:



Prioritization will become key to mitigate risk and so I will sacrifice everything to meet the product up until I get to the following barebones output, and then only working on slides and documentation after).

I am optimizing for personal growth and development while showcasing my data science knowledge and implementation ability.

At the same time, I will demonstrate quality metrics by routinely setting clear goals for myself and making sure that the final product is useful and easy to see -> that can be measured by talking to several people and pre-testing the app before shipping it off. I will also use checklists to manage quality and routinely update them based on risk and time accordingly.

3) If I have time
Some of the primary things I would like to do is go through my "DO NOT" scope things I have in the product to further demonstrate my abilities. Most likely, I would like to focus on using a library called **streamlit.io** and use

the library to create a GUI and user interface that can effectively help users customize and directly interact with everything going on in the model. This can include tweaking hyperparameters such as the K value for clustering, picking the number of Principal components, embedding their own interpretations of the data and cell composition while including newer resources and links, and so much more. IT can become the ultimately user-friendly data science panel for studying and learning more about tumors and their cellular composition. I would also like to use this time to then create mockups of the design and re-evaluate my proposal.

My second point would be to dive deeper into the biological side of things and start analyzing specific clusters and give them names in an **unsupervised fashion**. If I did not have the label names of the cells, I would want to further investigate some key features of each cluster such as cellular activity, specific gene expression profiles, which is essentially gene analysis and metastable dynamic states which I can then use to identify cell-types for newer data sets. This would be very critical in actually learning about new things of the patient, such as how effective a therapy was and whether it can be seen here or to visualize the immune system impacting the tumors in the body.

Another idea is to collect metrics on this such as cell lineages and cell viability metrics to give more detailed information about the clusters and activity. Perhaps it can help to also provide more background information for readers and turn this into a cancer information hub that outlines useful stats on gene expression and regulatory elements involved in cancer progression or reduction.

The fourth and final idea would be to look at trajectory inference where I can look at differential gene expressions amongst different clusters identified related to cell cycles, replication, genetic info transfer, and more which can help to identify progression of cells in the body through time. This adds a completely new layer to the data and can thus allow for better analysis of tumor progression in earlier stages.