Investigating the Role of Topological Methods in Single-Cell RNA-Seq Analysis

Ryan Hayden

Final Project for APM 598

1. Background

1.1. Project Goals

The primary goal of this project is to investigate the aspects of the scRNA-seq analysis pipeline which are particularly relevant to the topological concepts covered in APM 598. This investigation will cover a variety of current best practice methods for performing a scRNA-seq analysis, from the quality control stage to a cell-type annotation, as well as some clinical applications. In particular, we will emphasize the use of topological methods in this analysis pipeline, and we will also consider the pros and cons of such methods when compared to the ”classical” methods. We will then perform an investigation of our own on a variety of gene expression datasets, and we will conclude by analyzing our results and also considering the role of TDA in this new and important field.

1.2. Intro to Genomic Analysis and Single-Cell Technology

Like most aspects of the modern world, the field of biology has undergone a revolution due to the rise of high-throughput technologies and the corresponding big data advances. This increasing access to massive amounts of biological data is helping scientists tackle many problems previously thought to be out of reach, and has thus played a central role in many recent advancements in cancer research and even in the rapid development of the mRNA Covid vaccines. In particular, the proliferation of single cell gene expression data has been central to this revolution.

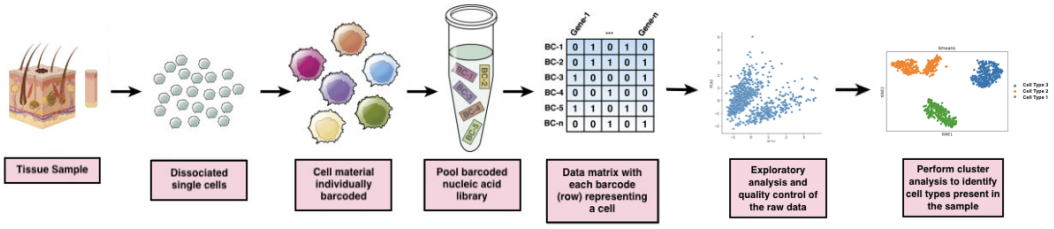


Fig. 1: scRNA seq analysis lets us ”convert” a tissue sample into gene expression matrices upon which we can use dimension reduction and clustering algorithms

Single Cell RNA-Sequence Analysis (scRNA-seq) is a ground breaking new technology for obtaining gene expression data through whole transcriptome profiling that first started being utilized in the early 2010’s. Previous technology limited researchers to methods like bulk RNA-seq analysis, which as its name suggests only provides bulk/aggregate gene counts for a group of cells. On the other hand, scRNA-seq technology, like that of the popular Chromium System from 10x Genomics, allows for a level of single-cell granularity that enables much more detailed and sophisticated analysis than was previously possible. This type of analysis often boils down to a data science or bioinformatics problem as the single cell gene expression data can simply be thought of as a matrix containing gene count values, where the rows represent individual cells and the columns denote different genes.

Given a tissue sample, one of the first things to do when analyzing the associated scRNA-seq data is to determine how many and which types of cells are present in the sample. Mathematically speaking, we want to cluster the gene expression vectors of each of the cells in our dataset into distinct groups of similarly expressed

cells. This is known as a Q-mode cluster analysis of the single cell gene expression data. Along with biologically inspired heuristics, this allows us to determine the number of distinct cell clusters in the data. Then, biological experts can look within each cell cluster for certain ”marker” genes that might indicate which cell type each cluster actually represents.

This process, known as cell type annotation, helps determine cell heterogeneity within a tissue sample, which can then enable rigorous comparisons between control samples and experimental samples. Such an analysis can be of great importance in many biomedical applications. For example, scRNA-seq analysis is central to the field of personalized diagnostics and to biomarker detection in general. However, while this new technology has undoubtedly heralded the beginning of a new age of big data in biology and bioinformatics, it has also brought about a host of computational and statistical problems that need to be addressed if progress should continue.

1.3. Overview of Challenges in scRNA-seq Analysis and the role of Topological Data Analysis

The primary characteristics to note about scRNA-seq data is that this data is obtained via measurements from a biological source, so it is often noisy and very sparse, and perhaps more importantly, the data exists in an extremely high dimensional space, with most datasets containing expression counts of up to millions of individual cells over 20,000+ different genes. In general we must consider all the typical problems of big data analytics.

As such, the first step in the scRNA-seq data analysis pipeline is to perform some quality control and normalization measures, as well as perform basic exploratory analyses. Typically this involves filtering for outlier genes and cells, and running a principal components analysis (PCA) on the data to look for any significant features. At this point, we have moved on from what we consider ”raw” data to the ”cleaned” data that we will actually perform our cluster analysis on. However, since our data is very sparse and exists in such a high-dimensional space, traditional clustering algorithms will not work very well due in part to what is known as the ”curse” of dimensionality. Therefore, the first step is to perform some form of dimensionality reduction.

Specifically, we want to reduce the gene expression data from 20,000+ dimensions down to just 2 dimensions, so that it can be easily visualized for the cluster analysis. One of the most popular methods of dimensionality reduction for scRNA data is a non-linear variant known as t-distributed stochastic neighbor embedding (tSNE), which is often used in tandem with PCA in order to better suppress any noise and also to deal with the sparsity of the data. The central idea underlying these dimensionality reduction methods is that the high-dimensional data actually lies upon some lower-dimensional manifold in the ambient high-dimensional space. This is an idea known as the manifold hypothesis and along with the associated idea of manifold learning, it is central to understanding the key role of topology in genomic data analysis, and high-dimensional analysis in general.

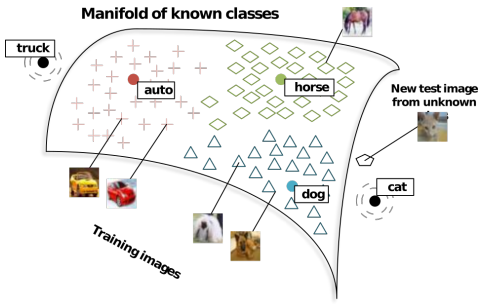


Fig. 2: a popular example of the manifold hypothesis, with image data, which is another form of high-dimensional data, although not nearly as high-dimensional as most gene expression data. However the underlying idea of uncovering a latent lower-dimensional structure from high-dimensional data is the same in scRNA-seq analysis

Essentially, the goal in manifold learning is to capture some of the latent and lower-dimensional topological structure of the data. However, despite relatively strong performance, methods like t-SNE are not very topologically rigorous, and simpler methods like PCA are sub-optimal for dealing with data as complex as gene expression data. As such, one aspect that we will focus on is the emerging use of more sophisticated topological methods, like the Mapper algorithm, for the dimensionality reduction of gene expression data. Once we have successfully reduced the dimension of the data and are able to visualize the plotted data, we will perform a

clustering analyses. Once again this is an area that can be beneficially considered from a topological perspective, as the individual clusters are simply distinct connected components, which can be measured using the tools of algebraic topology and homology. In fact, the Mapper algorithm also produces a form of clustering analyses, so we will compare its results to the results produced by more traditional methods like K-Means clustering. It should also be noted that tSNE and K-Means are not the only popular methods, as there are other useful methods like UMAP dimensionality reduction and Louvain clustering, but we will not focus on their performance in this project, although we will provide the necessary code in the appendix to perform an analysis with such methods.

Overall, our main point of emphasis is that the common difficulties of scRNA-seq data analysis has led to a demand for novel and promising methods from topology which allow us to better probe the ”shape” of extremely complex biological data. With that in mind, we will explore some of these emerging methods and analyze their effectiveness in comparison with some of the main ”classical” methods.

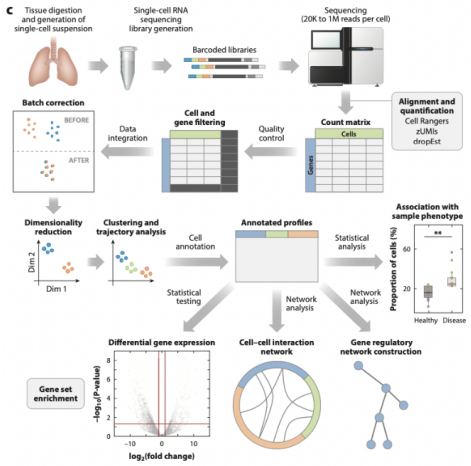


Fig. 3: Overview of the entire scRNA data analysis pipeline shows the central role of dimensionality reduction and cluster analysis, two areas where rigorous topological methods offer a lot of promise and advantage over existing methods

2. Datasets and Software Packages

2.1. Dataset

As previously mentioned, a primary goal in scRNA-seq analysis is to obtain information on the cell types present in a sample, solely from the raw gene expression data. In this investigation we will perform a cell-type cluster analysis on several datasets, including 2 popular benchmarking datasets containing human lung cancer cell lines and also several datasets containing cell types from various tissue samples of the common mouse, Mus musculus. All of these datasets include the neccesary metadata and annotations, so we will be able to check the performance of our methods against the ground truth labels.

Benchmarking datasets:

a) 3 cell line Human Lung Adenocarcinoma dataset, containing cells from the lines HCC827, H1975 and H2228. This dataset contains the gene expression data of 902 cells across 16468 genes

b) 5 cell line Human Lung Adenocarcinoma dataset, containing cells from the lines HCC827, H1975, H2228, H838, and A549. This dataset contains the gene expression data of 3918 cells across 11786 genes

This data is available at https://github.com/LuyiTian/, and is from a paper authored by Luyi Tian, noted in the references section. The data was generated by 10x Chromium and is post quality control but has not yet been gene filtered. This dataset is a great benchmarking set because the data is well annotated and relatively free of errors because it was meticulously generated. We will use this dataset to explain our methodology in the next section.

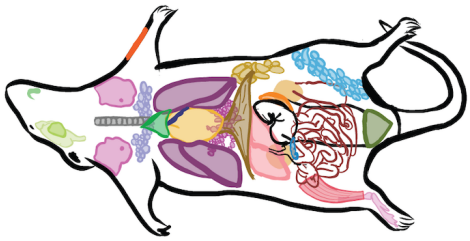


Fig. 4: Mus musculus, also known as the common house mouse, is the organism which we will focus our scRNA analysis on, in large part due to the easily obtainable and well-annotated dataset known as the Tabula Muris

Analysis datasets:

a) Mouse Pancreas dataset, containing cells from 9 different cell types. This dataset contains the gene expression data of 1961 cells across 23433 genes

b) Mouse Spleen dataset, containing cells from 3 different cell types. This dataset contains the gene expression data of 1718 cells across 23433 genes

c) Mouse Colon dataset, containing cells from 5 different cell types. This dataset contains the gene expression data of 4149 cells across 23433 genes

d) Mouse Fat dataset, containing cells from 10 different cell types. This dataset contains the gene expression data of 5862 cells across 23433 genes

e) Mouse Heart dataset, containing cells from 7 different cell types. This dataset contains the gene expression data of 7115 cells across 23433 genes

This data is available at https://tabula-muris.ds.czbiohub.org/ and is part of a larger dataset of over 100,000 cells and 20 organs/tissues, known as the Tabula Muris compendium. The data is from a project funded by the Chan Zuckerberg institute and is also meticulously labeled, but unlike the benchmarking dataset it is in raw form and has not been subjected to quality control measures yet. This makes the dataset well suited for use in a scRNA-seq analysis pipeline experiment since it also has labels to check the accuracy of our methods. We will use these datasets in our investigation and analysis sections, as they will help facilitate comparison between the ”classical” methods and the topological methods like Mapper.

index	0610009C13Rik	0610007C21Rik	0610007L01Rik	0610007N19Rik	0610007P08Rik	0610007P14Rik	0610007P22Rik
J12.MAA000386.3_11_M.1.1	0	61	0	0	0	5	6
C9.MAA000386.3_11_M.1.1	0	595	96	123	135	93	0
D13.MAA000386.3_11_M.1.1	0	221	99	115	0	0	57
F2.MAA000386.3_11_M.1.1	0	2	0	0	0	0	0
G12.MAA000386.3_11_M.1.1	0	56	5	0	0	17	78
I4.MAA000386.3_11_M.1.1	0	0	0	0	0	0	0
J13.MAA000386.3_11_M.1.1	0	113	48	4	0	0	0
L1.MAA000386.3_11_M.1.1	0	0	0	0	0	0	0
L2.MAA000386.3_11_M.1.1	0	0	0	0	0	0	0
M10.MAA000386.3_11_M.1.1	0	77	81	10	0	25	0
N20.MAA000386.3_11_M.1.1	0	0	0	0	0	0	0
P4.MAA000386.3_11_M.1.1	0	10	5	0	0	0	1
M11.MAA000386.3_11_M.1.1	0	122	30	7	0	3	0

Fig. 5: A portion of the raw gene expression data from the mouse fat cell dataset. Rows represent individual cells and columns represent different genes. This particular dataset contains counts for 5862 cells across 23433 genes. In our investigation we analyze this dataset as well as similar ones for the mouse heart, pancreas, spleen, and colon.

We will also note the existence of the R package Splatter, which lets you easily generate simulated scRNA-seq data. More info is available at: https://github.com/Oshlack/splatter

2.2. *Software Packages*

Our investigation will be performed in Python through Google Colab, and in particular, will make use of the package ScanPy, which is a toolkit explicitly built for the analysis of single cell data. We will use tools from ScanPy to perform quality control, dimensionality reduction, and clustering. We will also make use of popular libraries like Scikit-learn and NumPy. Although we are focusing on Python, there is a lot of similar material for the R programming language, like Seurat. For performing topological data analysis, we will use tools from the Scikit-TDA package, as well as the Keppler Mapper python implementation of the Mapper algorithm. In addition, we will provide relevant portions of the code we used in the Appendix, so that our results are reproducible.

3. Methods

In this section we will explain the typical methods that are used in scRNA-seq data analysis, from quality control to a completed cell type annotation. We will then contrast these methods with methods from topological data analysis, and in particular, the Mapper algorithm. In order to illustrate the use of these different methods in determining the cell types present in a tissue sample, we will analyze both the 3 cell line and 5 cell line human lung cancer scRNA datasets.

It should be noted that none of the methods require (or even use) the associated cell-type label information, but we will use it to color the cells by cell-type in order to give an idea of the effectiveness of each method in distinguishing cell types. Moreover, we provide the necessary code in the appendix so that this methodology is reproducible.

3.1. *Quality Control*

3.1.1. Initial Data Exploration

A common first step in scRNA analysis is to examine the raw data, usually by means of a principal components analysis (PCA). Principal components analysis is a popular technique from linear algebra which takes advantage of the data matrix format of the scRNA-seq data. PCA is part of a larger class of dimensionality reduction methods, known as spectral methods. The class of spectral methods includes other popular techniques like metric multidimensional scaling (MDS) and (non-linear) graph-based methods like Isomap and locally linear embedding (LLE). Essentially what spectral methods do is examine the the eigenvectors of a certain matrix. For example, in principal components analysis we examine the eigenvectors of the data covariance matrix through its eigen-decomposition or SVD.

(a) 3 cell line dataset (raw data)

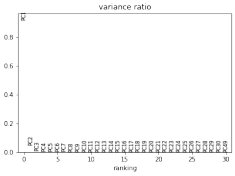
(b) 5 cell line dataset (raw data)

Fig. 6: 2-D PCA plots of the benchmark data reveals linear patterns, which indicates that a few principle components represents a large portion of the variation in the dataset. Moreover, the data failed to separate well. One possible issue is that we could have outliers that we will need to deal with via normalization

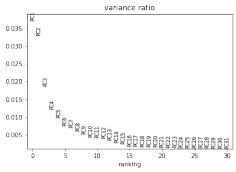
Typically we focus on the top few eigenvectors, as measured by their eigenvalue magnitude, which are also known as the top principal components. The first principal component is thus the direction that maximizes the variance of the data, and the n-th principal component is a direction orthogonal to the first $n-1$ principal components that maximizes the variance of the data. So, simply plotting the data along the first two principal components can give us an idea of any major features in the data. For example, a principal components analysis of the scRNA data covariance matrix might indicate the presence of outlier cells and genes which we would then want to filter out as part of our quality control. This indication would come in the form of outlier principal components responsible for unusually significant portions of the variation in the data.

In Figure 6, we see the result of projecting the 3 cell line and 5 cell line data onto their respective first two principal components. Because the human lung cell data has already been subjected to certain quality control measures, the principal components analysis already looks pretty good and you can see from the color labels that the distinct cell groups are even somewhat separated into clusters. This indicates that the data is in pretty good shape and is pretty close to being ready for downstream analysis. However the human lung cell data has not yet been subjected to gene/cell filtering so we will now move on to that step to further eliminate any outlier data that might corrupt our analysis.

3.1.2. Cell/Gene Filtering and Normalization



(a) Before quality control

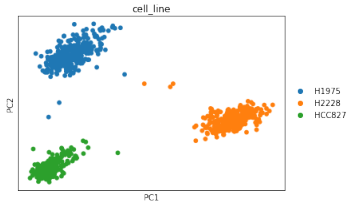


(b) After quality control

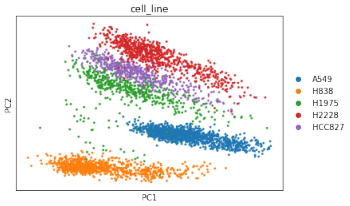
Fig. 7: Our normalization procedure helps to eliminate outliers and produce multiple principle components which contribute significantly to the overall variance of the data. Note that (for visualization purposes) these variance plots are from the Tabula Muris dataset since it shows a more dramatic change post-normalization than the (pre-cleaned) benchmarking data did.

The main purpose of quality control in scRNA-seq analysis is to remove outlier genes and cells that may skew the results of any analysis we perform. Our first step is to compute quality control metrics for both the cells and the genes. Fortunately, there is a ScanPy function which will take care of this for us, so all we have to do is run a few lines of code. Essentially, we want to remove cells which have unusually low amounts of reads, and cells with outlier amounts of detected genes, as both of these may signal data/sequencing corruption issues. Next, we count the number of RNA spike-ins to determine the ratio of ERCC spike-ins and endogenous RNA, in order to identify possibly damaged or dead cells. As for genes, we focus on removing genes which have very low expression levels. Once we have identified our quality control thresholds we can use the cell and gene filtering functions from ScanPy to finish the quality control process. In this investigation we choose to filter all cells with less than 750 genes detected, and we also filter all genes with less than 3 cells capturing at least 5 counts of the gene.

Our final step is to normalize and scale the data while also removing any especially highly expressed genes. The results of such a quality control process on the human lung cancer dataset can be seen in Figure 8, where the PCA plots post quality control show much more defined and distinct clusters, so our process worked and the data is now much more separable (ie: compare Figure 8 with Figure 6). Another way to check the effectiveness of the quality control process is to examine the variance ratio of the principal components. Ideally, we want there to be no outlier PC's responsible for a majority of the variance, and instead we prefer there to be a somewhat Gaussian distribution of the PC's. The beneficial effect that quality control and normalization has on the PC's can be directly seen in Figures 7a and 7b.



(a) 3 cell line dataset post normalization



(b) 5 cell line dataset post normalization

Fig. 8: After normalization and quality control we can see from the above 2-D PCA plots that distinct clusters are starting to form and thus the data is separating a lot better than before

3.2. Dimensionality Reduction and Clustering

Now that we have ran our data through a quality control and normalization process, we are ready to start the main portion of the analysis, namely, dimensionality reduction and clustering. There are fundamental similarities between these two processes, and often they work together hand-in-hand. For example, as we mentioned earlier, in order to avoid the so-called "curse" of dimensionality, we often perform dimensionality reduction before a clustering algorithm, as pairwise distances exhibit odd behaviors in high-dimensions. This is the case in scRNA-seq analysis, wherein we first run the quality controlled data through a dimensionality reduction algorithm before moving onto the cluster analysis. In the previous subsection, we mentioned a simple form of linear dimensionality reduction known as PCA, but in order to deal with our complex single cell data, we turn to a class of methods known as non-linear dimensionality reduction methods, although it should be mentioned these methods are often used in tandem with PCA. There are many such methods for non-linear dimensionality reduction and also many ways to perform a cluster analysis, but in this investigation we will focus on two methods popular in scRNA-seq analysis: t-SNE dimensionality reduction and K-means clustering.

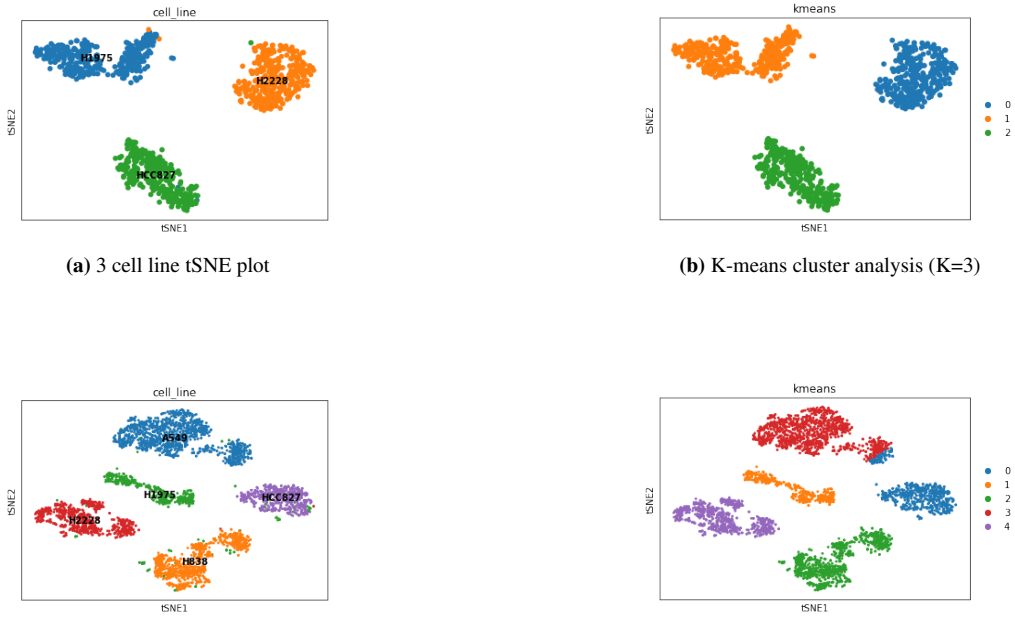
3.2.1. Non-Linear Dimensionality Reduction

The most popular form of non-linear dimensionality reduction in scRNA-seq analysis is t-distributed stochastic neighbor embedding (t-SNE), largely because compared to other methods, it focuses on local structure and tends to break up "tangled" data into seperate clusters. This is an characteristic that is particularly useful in analyzing scRNA seq data, as our primary goal is to separate a "blob" of individual cells into distnct cell-type clusters. To put it simply, t-SNE succeeds where PCA fails because while PCA tends to focus on placing dissimilar data points far apart (ie: it attempts to retain global structure), t-SNE focuses on keeping similar points close together (this is known as a local method). Although the theory behind t-SNE's superior performance is justified in part by the topological assumptions of the manifold hypothesis, t-SNE primarily makes use of probabilistic/statistical modeling and hence the primary calculation involved is the minimization of a KL divergence, as opposed to the spectral decompositions involved in PCA or the more topologically minded calculations involved in the Mapper algorithm. However, we can treat the results of a t-SNE dimensionality reduction similar to the principal components that result from a PCA. This can be seen in Figures 9a and 9c, where we projected the data down to the first 2 t-SNE components and colored each point by cell-type to illustrate the effectiveness of the method in preserving local structure and keeping similar cells near each other. The results are impressively accurate in regards to the ground truth, but this is partly due to the particularly nice nature of the benchmarking data. We will see in our analysis section that often the t-SNE results are not quite as impressive, and in fact, there are many criticisms of t-SNE's widespread use in genomics, which we will elaborate on at the end of the analysis section.

3.2.2. Clustering Methods

In our setting of scRNA-seq analyis, we are interested in performing a Q-mode cluster analysis, which groups together similar rows of the data matrix, which in our case represent individual cells. Given our t-SNE dimensionality reduced data, a Q-mode analysis on the data will give us an estimation of the number of cell-types present. In order to identify exactly which cell-types these are we would need to look within each cell-cluster for distinguising genetic bio-markers. However, all of the datasets we are analyzing have annotations/labels so we will be able to check the accuracy of our clustering analysis without having to uncover specific biomarkers.

The clustering method that we will focus on is K-means, which is popular in scRNA-seq analysis because it is easily scaled to large datasets and it is guaranteed to converge. K-means is part of a larger class of clustering methods known as centroid based clustering, in which the goal is to minimize intra-cluster distances from the cluster centroid. This makes it particularly well suited for our situation of attempting to find distinct cell clusters, and it can be seen in Figures 9b and 9d that besides a few small regions, the K-means algorithm recovered the ground truth clusters. However, the K-means method is also flawed in that it is sensitive to outliers and perhaps most detrimentally, it requires that the number k of clusters be chosen beforehand. There are other methods like DBSCAN clustering, hierarchical clustering, and the graph-based Louvain clustering which are also popular in scRNA-seq analysis, and we include code to run these methods in the appendix, but they also suffer from similar flaws. As such, we will consider a new class of clustering methods which draw heavily from topology.



(a) 3 cell line tSNE plot

(b) K-means cluster analysis (K=3)

(c) 5 cell line tSNE plot

(d) K-means cluster analysis (K=5)

Fig. 9: Dimensionality reduction of the normalized data, followed by a cluster analysis, reveals the presence of distinct clusters matching the ground truth. This benchmarking dataset has particularly nice properties and in practice most plots won't be as nice.

3.3. Topological Methods

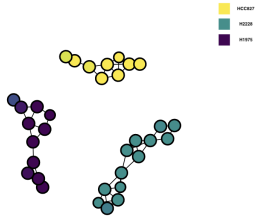
As we mentioned earlier, the manifold hypothesis provides a setting where one might expect topologically grounded tools to perform better than other methods. In this subsection, we will focus on the Mapper algorithm which was developed by some of the founders of the field of topological data analysis and has seen recent adoption in the scRNA community.

3.3.1. The Mapper Algorithm

It should first be noted that the Mapper algorithm is not simply a dimensionality reduction algorithm or clustering method, but rather some form of hybrid of both. In fact, the Mapper algorithm makes use of dimensionality reduction and clustering as part of its process. In this investigation we employed a version of the Mapper algorithm which uses t-SNE dimensionality reduction and DBSCAN clustering. The part where the Mapper algorithm really sets itself apart is that also produces an approximation of a Reeb Graph of the data, which is a rigorous topological description of the underlying shape of the data, given by level sets of some function defined on the data (ie: the filter function). This rigorous topological representation of the data is constructed from simplicial complexes built on the data, which we can consider as a point cloud in high-dimensional space. This provides us with a compact and global combinatorial representation of the data which lends itself very well to downstream analysis including machine learning pipelines. In addition, it is designed to capture the underlying topology, like the number of connected components and other latent structures, without the algorithmic/computational costs of other TDA methods like persistent homology. Similar to the classical methods used in scRNA-seq analysis, the Mapper algorithm is particularly useful in the scRNA domain as it preserves local features, like relations between clusters, as part of its fundamental emphasis on the topological concept of open neighborhoods.

Moreover, the Mapper algorithm is actually an improvement upon the classical methods for a variety of reasons. First, it does not require setting the number of clusters a priori, as the K-means algorithm does, and it allows for a multi-scale resolution which enables comparison of different tuning parameters. In addition, it is insensitive to small changes in the data, so it is robust to biological and sequencing noise. One particular advantage that Mapper has over a method like t-SNE is that Mapper actually clusters the points in the ambient high-dimensional space as opposed to the projected 2-D space, so it avoids some of the distortions common in t-SNE plots. The relative effectiveness of the Mapper algorithm compared to traditional methods can be seen in Figures 10 and 11,

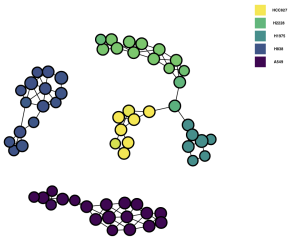
where we were able to recover the distinct cell-type clusters in each dataset, as well as possibly uncover further info regarding the inter-cluster relationships (for example, 3 out of the 5 clusters in Figure 11 are connected, and perhaps there is a biological reason for this being the case). Overall, as an exploratory tool, Mapper provides a comparable output to current best practice methods and even offers the promise of further benefits.



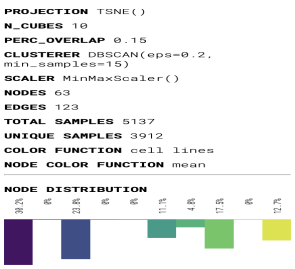
(a) 3 cell line Mapper graph

(b) 3 cell line Mapper summary

Fig. 10: Mapper analysis of the 3 cell line dataset using tSNE reduction and DBSCAN clustering clearly indicates the presence of 3 distinct clusters, matching the ground truth. Note that the node color represents the average cell-line label of the cells represented within each node.



(a) 5 cell line Mapper graph



(b) 5 cell line Mapper summary

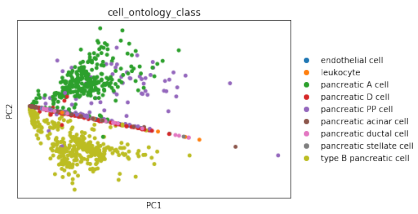
Fig. 11: Mapper analysis on the 5 cell line dataset indicates the presence of 5 distinct clusters, although it also potentially contains information regarding the inter-relatedness of these clusters

4. Investigation

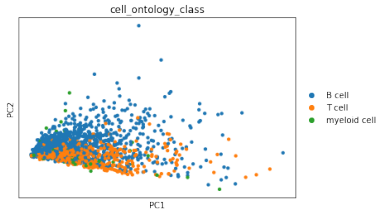
Now that we have provided the neccesary background for scRNA-seq analysis and explained the common methods, we will perform an investigation of our own. The goal of this investigation is to construct accurate cell-type annotations from raw scRNA-seq data. As previously mentioned we will analyze several datasets from the Tabulus Muris compendium, namely the mouse pancreas, spleen, colon, fat, and heart datasets. This will give us a good approximation of the scale and complexity of an actual investigation in a lab, although as before, we will have the true cell-type labels to check the efficacy of our methods. As part of the analysis we will discuss the performance of the stated topological methods as compared to classical methods in genomic analysis. We will then conclude by mentioning some major results in TDA-guided scRNA-seq analysis.

4.1. Initial Data Exploration and Quality Control

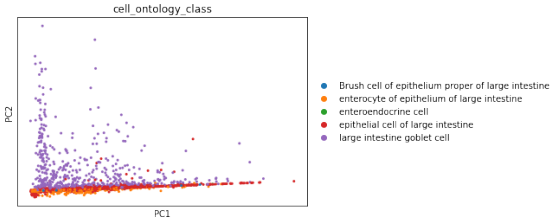
The 5 mouse cell data sets contain gene expression counts for over 20,000 individual cells across over 23,000 different genes. Since this data is completely raw, we must start by running a full quality control analysis on the data. The specific methods were outlined in the last section, and the effects of the QC and normalization processes can be seen in the differences between Figure 12 and Figure 13. One interesting thing to note is that all the datasets exhibited dominating levels of one particular gene, called Rn45s, so this was adressed as part of the quality control measures.



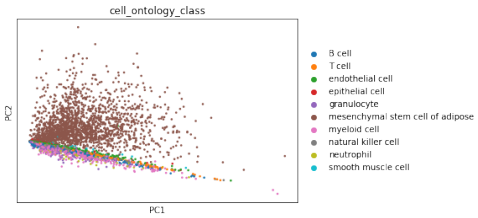
(a) Pancreas



(b) Spleen

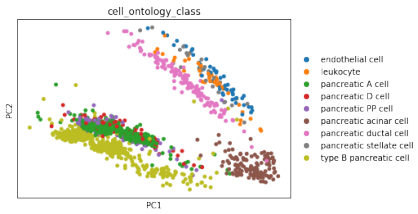


(c) Colon

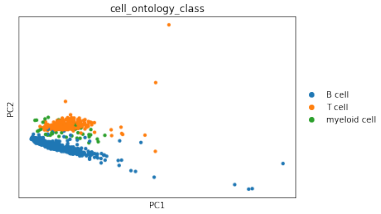


(d) Fat

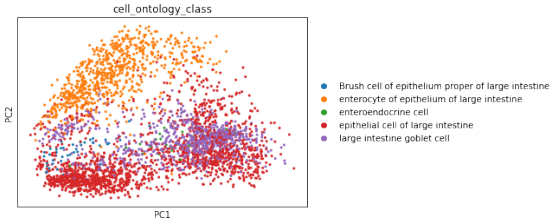
Fig. 12: Pre-normalized PCA plots of several portions of the Tabula Muris dataset



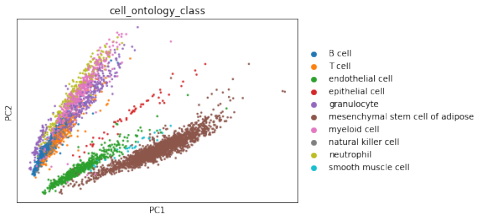
(a) Pancreas



(b) Spleen



(c) Colon

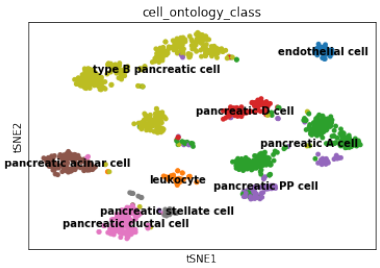


(d) Fat

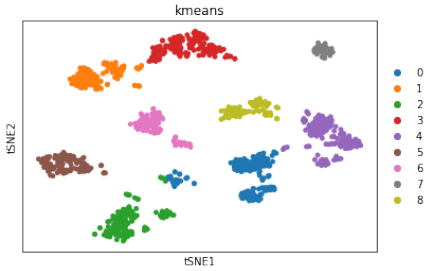
Fig. 13: Post-normalized PCA plots of several portions of the Tabula Muris dataset

4.2. Dimensionality Reduction and Cluster Analysis

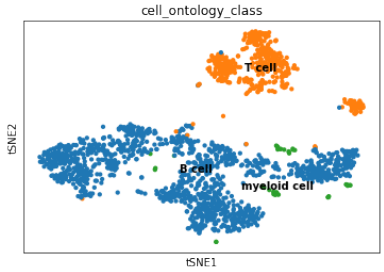
Now that the data has been subjected to strict quality control measures we can apply dimensionality reduction followed by a clustering analysis. The results of this analysis are shown in Figure 14, and although the cluster analysis was not as accurate as it was on the benchmarking dataset (it sometimes confused/mixed up some of the smaller cell-type clusters) it still performed remarkably well. Given that it involves such relatively simple methods, one can see why a t-SNE + K-means analysis is such a popular method in scRNA-seq analysis.



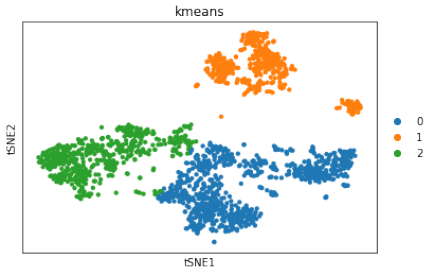
(a) Pancreas tSNE plot



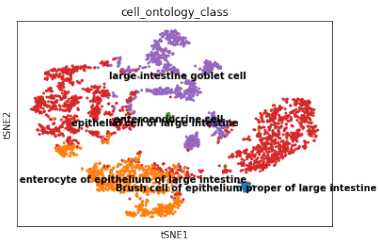
(b) Pancreas K-means (K=9)



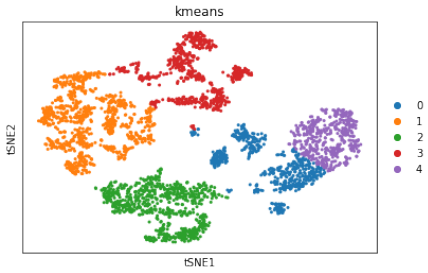
(c) Spleen tSNE plot



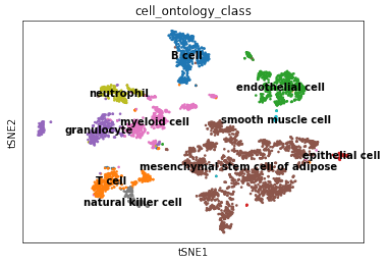
(d) Spleen K-means (K=3)



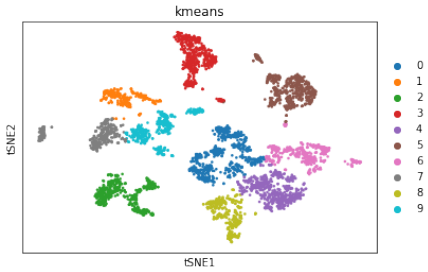
(e) Colon tSNE plot



(f) Colon K-means (K=5)



(g) Fat tSNE plot

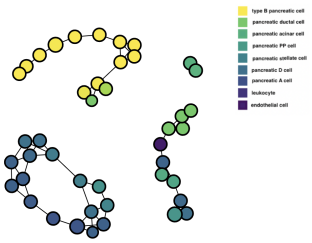


(h) Fat K-means (K=10)

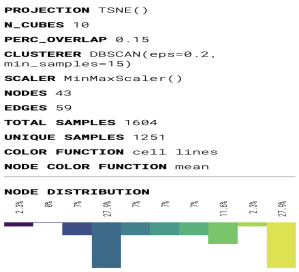
Fig. 14: Dimensionality reduction and cluster analysis of the 4 Tabula Muris datasets reveals the strengths of tSNE-based analysis

4.3. *Mapper Analysis and TDA*

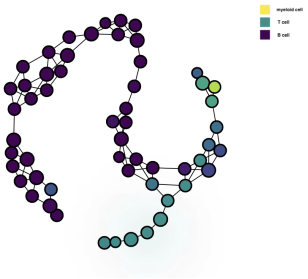
Now to compare with the classical methodology, we run the Mapper algorithm on the same datasets, using t-SNE dimensionality reduction and DBSCAN clustering. An initial inspection reveals promising and competitive results when compared to the classical methods in the previous subsection. In particular, most major cell clusters were recovered, and unlike the classical methods, we did not need to specify the number of clusters a priori. We will further analyze the comparative performance of this TDA method in the next section.



(a) Pancreas Mapper graph

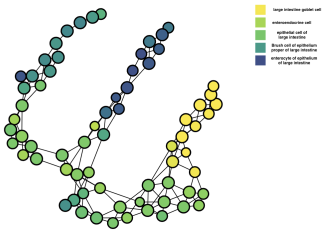


(b) Pancreas Mapper summary

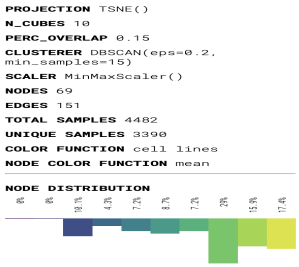


(c) Spleen Mapper graph

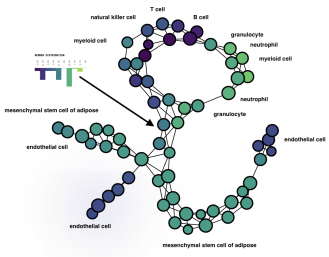
(d) Spleen Mapper summary



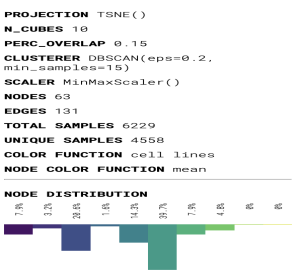
(e) Colon Mapper graph



(f) Colon Mapper summary



(g) Fat Mapper graph with region labels



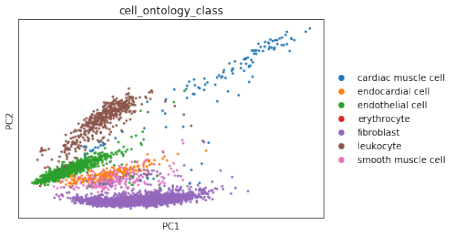
(h) Fat Mapper summary

Fig. 15: The Mapper algorithm was able to succesfuly capture many of the features shown in the t-SNE/K-means analysis including uncovering major components/clusters, as well as potentially revealing some new structures and features

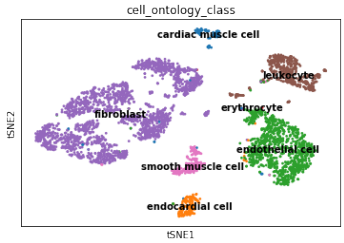
5. Analysis and Conclusion

5.1. Effectiveness of Topological vs Classical Methods

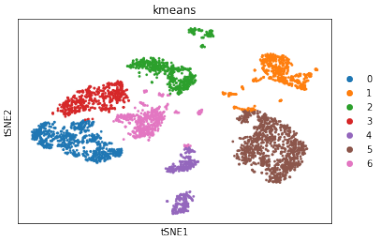
In order to fully illustrate the differences in the classical and topological methods we've covered, we have included a direct comparison of the methods in Figure 16, with the mouse heart dataset. The post normalized PCA indicated the presence of somewhere between 3-5 distinct clusters and a further t-SNE + K-means analysis yielded the image in Figure 16c. Despite having the knowledge of true number of clusters, the classical analysis was not able to correctly identify the true cell types, which are shown in the t-SNE plot in Figure 16b. The classical methods were not even able to pick up the fibroblast cell cluster, which is the largest cluster by number of cells. Meanwhile, the Mapper plot indicates the presence of at least 4-6 distinct cell types (as indicated by the Mapper summary) and also picks up on the 3 biggest clusters, namely the endothelial cells, leukocyte cells, and the fibroblast cells. Moreover, it keeps these 3 cell-types in largely separate clusters, which the classical analysis missed, as the K-means clustering broke the fibroblast cluster into 4 separate clusters. Overall, the Mapper analysis not only has competitive results, but it surpasses the classical method in some ways and also offers the ability to explore the graph and rescale for different parameters. Our analysis seems to indicate that at least as an exploratory tool, the Mapper algorithm is particularly useful and goes beyond just rivaling existing methods to actually providing certain unique insights.



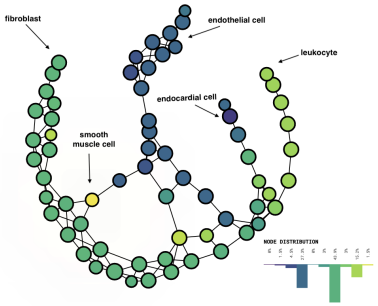
(a) Heart PCA



(b) Heart t-SNE



(c) Heart K-Means

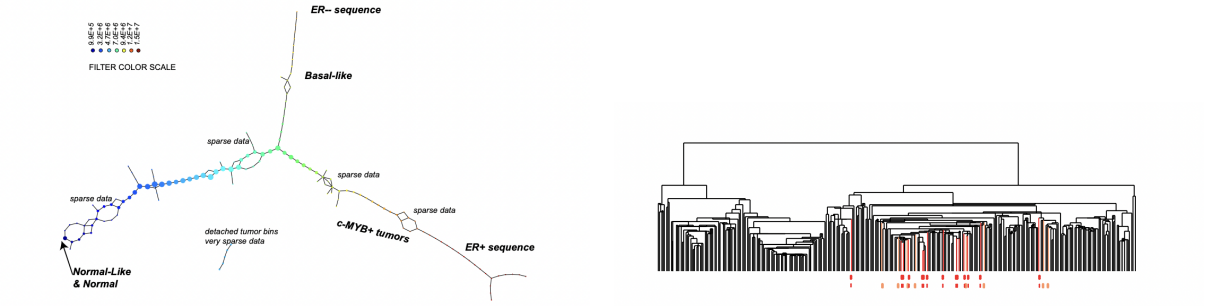


(d) Heart Mapper

Fig. 16: A comparison of the classical methodology and the topological methodology reveals the relative strengths of TDA.

However, it should be noted that the coloring on the Mapper graph is from the ground truth labels, so in practice we would not be able to rely on the colors to identify distinct cell clusters. Instead, due to Mappers approximation of a Reeb graph of the scRNA data, we can expect to see tendrils more-so than distinct clusters/components. As such, the Mapper graph in Figure 16d displays at least 4 such tendrils, which do in fact match up with the ground truth labels of 4 distinct cell-type clusters. And as mentioned before, the classical methods were not able to correctly recover the existence of these 4 cell-type clusters. Moreover, the connection between such tendrils can also indicate further information about the cell heterogeniety of the sample. In the case of Figure 16d, even without the label colors, we would have reason to believe the intersection regions between tendrils could contain relevant info. For example, the yellow node, identified as a cluster of smooth muscle cells, could be investigated using biomarker techniques so that it could be identified without the use of labels/annotations.

The advantages and promises of Mapper are made even more obvious when considering more complicated scenarios than just identifying distinct cell-type clusters in a tissue sample. For example, one of the most famous papers to utilize these TDA techniques is a 2011 paper by M. Nicolau (cited in the references), which successfully identified a previously unknown subgroup of Estrogen Receptor-positive breast cancers, known as the $c-MYB^{+}$ subgroup, which has improved clinical outcomes compared to other subgroups. The authors of the paper performed a version of Mapper analysis known as a Progression Analysis of Disease (PAD), which is very similar to the analysis that we performed, except they changed their filter function to one guided by what is known as Disease-Specific Genomic Analysis (DSGA). The basic results of this analysis, and their fundamental similarity to the analysis we performed, can be seen in Figure 17. In general, Mapper displays strengths in recovering genetic and clonal history, unlike other methods, so it offers exceptional promise in areas of research like personalized diagnostics and even evolutionary genomics.



(a) PAD analysis using Mapper

(b) Comparison with traditional clustering methods

Fig. 17: Progession Analysis of Disease using Mapper identified a previously unknown subgroup of breast cancer with a unique mutational profile and better than average survival rates. The image on the right reveals that traditional clustering methods missed this unique subgroup (marked in red), instead mixing it within multiple other clusters. This type of analysis could be key to personalized medicine

5.2. Conclusion

In this project, we set out to investigate the use of topological methods in scRNA-seq analysis. Specifically, we wanted to compare and contrast the use of methods like the Mapper algorithm with the so-called ”classical” methods of the field. In addition to providing a detailed methodology, we also performed an analysis on several tissue samples from the common mouse, wherein we found that a Mapper analysis is certainly competitive with current state-of the-art methods. In addition, we discussed the ways that a Mapper analysis even out-performs other existing methods, such as its unique ability to capture genetic history and relationships. The effectiveness of Mapper in scRNA-seq analysis is particularly relevant as there has been growing criticism of t-SNE based analysis from within the biology and statistical genetics community. Most notably, L. Pachter recently released a paper entitled, ”The Specious Art of Single-Cell Genomics” complete with an associated software package called ”Picasso” which claims to let you embed scRNA-seq data into any 2D shape of your choosing while maintaining a ”more faithful” representation of the data than a t-SNE plot does (see Figure 18).

Needless to say, there is a growing demand for more rigorous methods in scRNA-seq analysis, and topological tools like the Mapper algorithm certainly offer a lot of promise in this regard. However, in order to be truly effective, TDA methods need to be better developed so that they fit into existing analysis pipelines. These rigorous methods are not of much use if the people employing them do not understand them and are not able to integrate their results with existing downstream pipelines. Overall, the field of TDA is still relatively new and its reliance on topology will certainly be a barrier for entry for many biologists, but its rigorous mathematical underpinnings will serve it well in an area as complex as scRNA-seq analysis and as such, it is only a matter of time before tools like Mapper start gaining more widespread adoption.

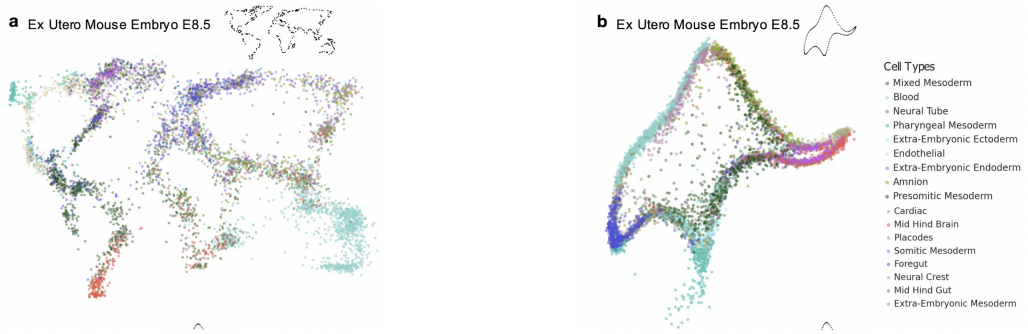


Fig. 18: There are many criticisms of tSNE and UMAP, particularly regarding the arbitrary nature of their results. One paper even mocks this arbitrary nature by claiming that you can embed the scRNA into any 2D shape while maintaining a "more faithful" representation of the raw data than tSNE or UMAP. Criticisms like these are part of the reason there is growing interest in more rigorous methods, like Mapper, which are built from the strong foundations of topology

References

1. S. Bell, "Analysis of Single Cell RNA-Seq Data Workshop" , (Chan-Zuckerberg Initiative, 2019).
2. M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," in *Proceedings of the National Academy of Sciences* (Academic, 2011).
3. L. Pachter, "The Specious Art of Single-Cell Genomics" , (Biorxiv Preprint, 2021).
4. R. Rabadan, and A. Blumberg, "Topological Data Analysis for Genomics and Evolution" , (Cambridge University Press, 2020).
5. Tabula Muris Consortium, "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris" , (Nature, 2018).
6. L. Tian, "scRNA-seq mixology: towards better benchmarking of single cell RNA-seq analysis methods" , (Nature Methods, 2019).
7. T. Wilson, "Clustering and Topological Data Analysis of Single-Cell RNA Sequencing Data" , (University of Surrey, 2021).

A. Code

```
# imports and installs
!pip install matplotlib==3.1.3 # current version produces error w/ scanpy
!pip install sklearn # tools for general data analysis
!pip install scanpy # tools for scRNA-seq analysis
!pip install kmapper # tools for Mapper TDA analysis
!pip3 install KMeans # clustering
!pip install louvain # clustering

from google.colab import files, output, drive
import sklearn
import matplotlib.pyplot as plt
import kmapper as km
from kmapper import jupyter
import seaborn as sns
import scanpy as sc
import pandas as pd
import io
import numpy as np
import math
from sklearn import metrics
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans, DBSCAN
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler

!pip install matplotlib==3.1.3 # reinstall to force old package version
```

Fig. 19: Set up the necessary installs and imports

```
# set up local file access
from google.colab import files
uploaded = files.upload()

# load gene expression data
data = pd.read_csv(io.StringIO(uploaded['Fat-counts.csv'].decode('utf-8')), index_col=0)

# check data, transpose if neccesary
data=np.transpose(data)
data

# set up local file access
from google.colab import files
uploaded = files.upload()

# load gene expression metadata
metadata = pd.read_csv(io.StringIO(uploaded['annotations_FACS.csv'].decode('utf-8')), index_col=0)

# check metadata, select relevant columns if neccesary
metadata=metadata.loc[metadata['tissue'].isin(['Fat'])]
metadata

# run this if data and metadata dont match up
cellclass=metadata.cell_ontology_class
cellclass=cellclass.to_frame()
mergedf=data.merge(cellclass, left_index=True, right_index=True)
metadf=mergedf.cell_ontology_class
metadf=metadf.to_frame()
bladf=mergedf.drop(columns=['cell_ontology_class'])

# now raw data and metadata have matching sizes
data=bladf
metadata=metadf

# confirm data and metadata match up
data.shape
metadata.shape

# create annotated data matrix (ie: anndata) to use with scanpy
adata_raw = sc.AnnData(X = data, obs = metadata)
```

Fig. 20: Load your data

```
# quality control
adata_qc=adata_raw # keep copy of the raw data
is_spike_in = {}
for gene_name in adata_qc.var_names:
    if 'ERCC' in gene_name:
        is_spike_in[gene_name] = True # record that we found a spike-in
    else:
        is_spike_in[gene_name] = False # record that this was not a spike-in
adata_qc.var['ERCC'] = pd.Series(is_spike_in) # label the spike ins
qc = sc.pp.calculate_qc_metrics(adata_qc, qc_vars = ['ERCC']) # scanpy function
cell_qc_dataframe = qc[0] # cell quality control
gene_qc_dataframe = qc[1] # gene quality control

# cell filtering and gene filtering
low_ERCC_mask = (cell_qc_dataframe['pct_counts_ERCC'] < 10)
adata_qc = adata_qc[low_ERCC_mask]
sc.pp.filter_cells(adata_qc, min_genes = 750) # filter cells
sc.pp.filter_genes(adata_qc, min_cells = 2) # filter genes
sc.pp.filter_genes(adata_qc, min_counts = 10)

# run PCA as exploratory measure to check the data out
sc.pp.pca(adata_qc)
sc.pl.pca_overview(adata_qc, color='cell_ontology_class') # plot

# normalize the data
adata_norm=adata_qc # keep copy of qc data
sc.pp.normalize_per_cell(adata_norm, counts_per_cell_after=1e6)
sc.pp.normalize_total(adata_norm, target_sum=1e6, exclude_highly_expressed=True)

# (OPTIONAL) Remove highly expressed genes distorting the data
not_Rn45s = adata_norm.var.index != 'Rn45s'
adata_no_Rn45s = adata_norm[:, not_Rn45s] # keep copy of normed data
# need to check which genes to remove

# scale the data
adata_scale=adata_no_Rn45s
# adata_scale=adata_norm
sc.pp.log1p(adata_scale)
sc.pp.scale(adata_scale)

# re-run PCA, should seperate data better this time
sc.pp.pca(adata_scale)
sc.pl.pca_overview(adata_scale, color='cell_ontology_class') # plot

adata=adata_scale # adata is now quality controlled, normalized, and scaled
```

Fig. 21: Perform quality control and normalization on the raw gene expression data

```
# tSNE dimensionality reduction
sc.tl.tsne(adata, perplexity=30, learning_rate=1000, random_state=0)
sc.pl.tsne(adata, color='cell_ontology_class', legend_loc='on data') # plot

# UMAP dimensionality reduction
sc.pp.neighbors(adata)
sc.tl.umap(adata, min_dist=0.5, spread=1.0, random_state=0, n_components=2)
sc.pl.umap(adata, color='cell_ontology_class', legend_loc='on data') # plot

# K-means clustering with tSNE data (can swap for UMAP)
tsne_coordinates = adata.obsm['X_tsne'] # retrieve coordinates
kmeans = KMeans(n_clusters=10, random_state=0).fit(tsne_coordinates)
# make sure to set number of clusters for k-means
adata.obs['kmeans'] = kmeans.labels_
adata.obs['kmeans'] = adata.obs['kmeans'].astype(str)
sc.pl.tsne(adata, color='kmeans') # plot

# Louvain graph based clustering with tSNE (can swap for UMAP)
sc.tl.louvain(adata, resolution=0.2)
sc.pl.tsne(adata, color='louvain') # plot
```

Fig. 22: Perform dimensionality reduction and cluster analysis

```
# convert data matrix in adata to dataframe
df = pd.DataFrame(adata.X)
df.set_index(adata.obs.index)

# Mapper analysis
mapper = km.KeplerMapper(verbose=0)
projected_data = mapper.fit_transform(df, projection=sklearn.manifold.TSNE(), distance_matrix="euclidean")

# build simplicial complex
graph = mapper.map(projected_data, clusterer=sklearn.cluster.DBSCAN(eps=0.2, min_samples=15),  cover=km.Cover(n_cubes=10, perc_overlap=0.15))

# keep track of metadata
metadata=adata.obs['cell_ontology_class']
meta=metadata.to_frame()
metadata.cell_line = pd.Categorical(meta.cell_ontology_class)
celltype=metadata.cell_line.codes

#visualize
graph_file = "Mouse Fat scRNAseq_graph.html"
graph_title = "Mouse Fat scRNA-seq Simplicial Complex"
html = mapper.visualize(graph, color_values=celltype, color_function_name="cell lines", path_html=graph_file, title=graph_title)
files.download(graph_file) # plot

meta.index = np.arange(0, len(meta))
meta # can use this to check Mapper color labels
```

Fig. 23: Run Mapper analysis and visualization